



# CLMaaS CERT+ API Guide

---

Version: 2022.1.0

# Copyright AppViewX, Inc.

**Copyright © 2022 AppViewX, Inc. All Rights Reserved.**

This document may not be copied, disclosed, transferred, or modified without the prior written consent of AppViewX, Inc. While all content is believed to be correct at the time of publication, it is provided as general-purpose information. The content is subject to change without notice and is provided “as is” and with no expressed or implied warranties whatsoever, including, but not limited to, a warranty for accuracy made by AppViewX. The software described in this document is provided under written license only, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. Unauthorized use of software or its documentation can result in civil damages and criminal prosecution.

## **Trademarks**

The trademarks, logos, and service marks displayed in this manual are the property of AppViewX or other third parties. Users are not permitted to use these marks without the prior written consent of AppViewX or such third party which may own the mark.

## **External Reference Links**

This product includes software developed by the CentOS Project ([www.centos.org](http://www.centos.org)).

This product includes software developed by Red Hat, Inc. ([www.redhat.com](http://www.redhat.com)).

This product includes software developed by VMware, Inc. ([www.vmware.com](http://www.vmware.com)).

All other trademarks mentioned in this document are the property of their respective owners.

## **Contact Information**

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: [info@appviewx.com](mailto:info@appviewx.com)

Web: [www.appviewx.com](http://www.appviewx.com)

# Contents

Preface.....	15
Revision History.....	15
About this Guide .....	15
Audience.....	15
Text Conventions.....	15
<b>Chapter 1. Overview.....</b>	<b>16</b>
<b>Chapter 2. Prerequisites.....</b>	<b>17</b>
Overview-Prerequisites.....	17
API Access with User Account.....	17
Retrieve Session ID using Login API.....	17
Use Session ID in the Header for Further API Calls.....	19
API Access with Service Account.....	21
Retrieve Access Token using get-service-token API.....	21
Use Access Token in the Header for Further API Calls.....	23
<b>Chapter 3. Overview.....</b>	<b>25</b>
<b>Chapter 4. Request Structure.....</b>	<b>26</b>
<b>Chapter 5. Response Structure.....</b>	<b>27</b>
<b>Chapter 6. Status Codes.....</b>	<b>28</b>
<b>Chapter 7. Use Case.....</b>	<b>29</b>
<b>Chapter 8. Sample Response:.....</b>	<b>30</b>
<b>Chapter 9. Overview.....</b>	<b>31</b>
<b>Chapter 10. Before you begin.....</b>	<b>32</b>
<b>Chapter 11. Request Structure.....</b>	<b>33</b>
<b>Chapter 12. Response Structure.....</b>	<b>39</b>
<b>Chapter 13. Status codes.....</b>	<b>40</b>
<b>Chapter 14. CA Specific Information.....</b>	<b>41</b>
Sectigo.....	41

Sectigo Request Objects.....	41
Sample Request/Response.....	45
DigiCert.....	47
DigiCert Request Objects.....	47
Sample Request/Response.....	51
InCommon.....	53
InCommon Request Objects.....	53
Sample Request/Response.....	56
AppViewX.....	58
AppViewX Request Objects.....	58
Sample Request/Response.....	62
EJBCA.....	64
EJBCA Request Objects.....	64
Sample Request/Response.....	68
GlobalSign.....	70
GlobalSign Request Objects.....	70
Sample Request/Response.....	75
Trustwave.....	76
Trustwave Request Objects.....	76
Sample Request/Response.....	80
Entrust.....	82
Entrust Request Objects.....	82
Sample Request/Response.....	86
Symantec.....	88
Symantec Request Objects.....	88
Sample Request/Response.....	92
Let's Encrypt.....	94
Lets Encrypt Request Objects.....	94
Sample Request/Response.....	98

GoDaddy.....	100
GoDaddy Request Objects.....	100
Sample Request/Response.....	104
Google.....	106
Google Request Objects.....	106
Sample Request/Response.....	110
Amazon.....	112
Amazon Request Objects.....	112
Sample Request/Response.....	115
Microsoft Enterprise.....	117
Microsoft Enterprise Request Objects.....	117
Sample Request/Response.....	121
Microsoft Standalone.....	123
Microsoft Standalone Request Objects.....	123
Sample Request/Response.....	128
OpenTrust Request Objects.....	130
Sample Request/Response.....	137
Sample Request:.....	138
Sample Response:.....	139
Entrust MPKI.....	139
Entrust MPKI Request Objects.....	139
Sample Request/Response.....	143
<b>Chapter 15. Common Payload Objects.....</b>	<b>146</b>
<b>Chapter 16. Overview.....</b>	<b>147</b>
<b>Chapter 17. Request Structure.....</b>	<b>148</b>
<b>Chapter 18. SyncResponseStructure.....</b>	<b>155</b>
<b>Chapter 19. Sample Request/Response.....</b>	<b>156</b>
Sample Request:.....	156
Sample Response:.....	157

<b>Chapter 20. Overview.....</b>	<b>159</b>
<b>Chapter 21. Before you begin.....</b>	<b>160</b>
<b>Chapter 22. Request Structure.....</b>	<b>161</b>
<b>Chapter 23. Response Structure.....</b>	<b>163</b>
<b>Chapter 24. Status codes.....</b>	<b>164</b>
<b>Chapter 25. Sample Request/Response.....</b>	<b>166</b>
Sample Request:.....	166
Sample Response:.....	166
<b>Chapter 26. Overview.....</b>	<b>167</b>
<b>Chapter 27. Request Structure.....</b>	<b>168</b>
<b>Chapter 28. Sync Response Structure.....</b>	<b>171</b>
<b>Chapter 29. Sample Request/Response.....</b>	<b>172</b>
Sample Request:.....	172
Sample Response:.....	172
<b>Chapter 30. Overview.....</b>	<b>173</b>
<b>Chapter 31. Before you begin.....</b>	<b>174</b>
<b>Chapter 32. Request Structure.....</b>	<b>175</b>
<b>Chapter 33. Response Structure.....</b>	<b>177</b>
<b>Chapter 34. Status codes.....</b>	<b>178</b>
<b>Chapter 35. Sample Request/Response.....</b>	<b>180</b>
Sample Request:.....	180
Sample Response:.....	180
<b>Chapter 36. Overview.....</b>	<b>181</b>
<b>Chapter 37. Request Structure.....</b>	<b>182</b>
<b>Chapter 38. Sync Revoke Response Structure.....</b>	<b>185</b>
<b>Chapter 39. Sample Request/Response.....</b>	<b>186</b>
Sample Request:.....	186
Sample Response:.....	186
<b>Chapter 40. Overview.....</b>	<b>187</b>

<b>Chapter 41. Before you begin.....</b>	<b>188</b>
<b>Chapter 42. Request Structure.....</b>	<b>189</b>
<b>Chapter 43. Response Structure.....</b>	<b>191</b>
<b>Chapter 44. Status codes.....</b>	<b>192</b>
<b>Chapter 45. Sample Request/Response.....</b>	<b>194</b>
Sample Request:.....	194
Sample Response:.....	194
<b>Chapter 46. Overview.....</b>	<b>195</b>
<b>Chapter 47. Request Structure.....</b>	<b>196</b>
<b>Chapter 48. Sync Response Structure.....</b>	<b>199</b>
<b>Chapter 49. Sample Request/Response.....</b>	<b>200</b>
Sample Request:.....	200
Sample Response:.....	200
<b>Chapter 50. Overview.....</b>	<b>202</b>
<b>Chapter 51. Before you begin.....</b>	<b>203</b>
<b>Chapter 52. Request Structure.....</b>	<b>204</b>
<b>Chapter 53. Response Structure.....</b>	<b>206</b>
<b>Chapter 54. Status codes.....</b>	<b>207</b>
<b>Chapter 55. Sample Request/Response.....</b>	<b>209</b>
Sample Request:.....	209
Sample Response:.....	209
<b>Chapter 56. Overview.....</b>	<b>210</b>
<b>Chapter 57. Request Structure.....</b>	<b>211</b>
<b>Chapter 58. Sync regenerate Response Structure.....</b>	<b>214</b>
<b>Chapter 59. Sample Request/Response.....</b>	<b>215</b>
Sample Request:.....	215
Sample Response:.....	215
<b>Chapter 60. Overview.....</b>	<b>216</b>
<b>Chapter 61. Before you begin.....</b>	<b>217</b>

<b>Chapter 62. Request Structure.....</b>	<b>218</b>
<b>Chapter 63. Response Structure.....</b>	<b>220</b>
<b>Chapter 64. Status codes.....</b>	<b>221</b>
<b>Chapter 65. Sample Request/Response.....</b>	<b>223</b>
Sample Request:.....	223
Sample Response:.....	223
<b>Chapter 66. Overview.....</b>	<b>224</b>
<b>Chapter 67. Before you begin.....</b>	<b>225</b>
<b>Chapter 68. Request Structure.....</b>	<b>226</b>
<b>Chapter 69. Response Structure.....</b>	<b>228</b>
<b>Chapter 70. Status codes.....</b>	<b>229</b>
<b>Chapter 71. Sample Request/Response.....</b>	<b>231</b>
Sample Request:.....	231
Sample Response:.....	231
<b>Chapter 72. Overview.....</b>	<b>232</b>
<b>Chapter 73. Request Structure.....</b>	<b>233</b>
<b>Chapter 74. Response Structure.....</b>	<b>239</b>
<b>Chapter 75. Status codes.....</b>	<b>241</b>
<b>Chapter 76. Sample request/response.....</b>	<b>242</b>
Sample Request:.....	242
Sample Response:.....	242
<b>Chapter 77. Overview.....</b>	<b>246</b>
<b>Chapter 78. Request Structure.....</b>	<b>247</b>
<b>Chapter 79. Response Structure.....</b>	<b>249</b>
<b>Chapter 80. Status codes.....</b>	<b>250</b>
<b>Chapter 81. Sample Request/Response.....</b>	<b>251</b>
Sample Request:.....	251
Sample Response:.....	251
<b>Chapter 82. Overview.....</b>	<b>254</b>

<b>Chapter 83. Request Structure.....</b>	<b>255</b>
<b>Chapter 84. Response Structure.....</b>	<b>257</b>
<b>Chapter 85. Status codes.....</b>	<b>258</b>
<b>Chapter 86. Sample Request/Response.....</b>	<b>259</b>
Sample Request:.....	259
Sample Response:.....	259
<b>Chapter 87. Overview.....</b>	<b>261</b>
<b>Chapter 88. Request Structure.....</b>	<b>262</b>
<b>Chapter 89. Response Structure.....</b>	<b>263</b>
<b>Chapter 90. Status codes.....</b>	<b>264</b>
<b>Chapter 91. Sample Request/Response.....</b>	<b>265</b>
Sample Request:.....	265
Sample Response:.....	265
<b>Chapter 92. Overview.....</b>	<b>267</b>
Request Structure.....	267
Response Structure.....	268
Status codes.....	268
Sample Request/Response.....	269
Sample Request:.....	269
Sample Response:.....	269
<b>Chapter 93. Overview.....</b>	<b>270</b>
<b>Chapter 94. Request Structure.....</b>	<b>271</b>
<b>Chapter 95. Response Structure.....</b>	<b>272</b>
<b>Chapter 96. Status codes.....</b>	<b>273</b>
<b>Chapter 97. Sample Response.....</b>	<b>274</b>
<b>Chapter 98. Overview.....</b>	<b>275</b>
<b>Chapter 99. Request Structure.....</b>	<b>276</b>
<b>Chapter 100. Response Structure.....</b>	<b>279</b>
<b>Chapter 101. Status codes.....</b>	<b>280</b>

<b>Chapter 102. Sample Request/Response.....</b>	<b>282</b>
Sample Request:.....	282
Sample Response:.....	282
<b>Chapter 103. Overview.....</b>	<b>283</b>
<b>Chapter 104. Common Response Objects.....</b>	<b>284</b>
<b>Chapter 105. Overview.....</b>	<b>285</b>
<b>Chapter 106. Before you Begin.....</b>	<b>286</b>
<b>Chapter 107. Request Structure.....</b>	<b>287</b>
<b>Chapter 108. Response Structure.....</b>	<b>292</b>
<b>Chapter 109. Status codes.....</b>	<b>293</b>
<b>Chapter 110. Sample Request/Response.....</b>	<b>295</b>
Sample Request:.....	295
Sample Response:.....	295
<b>Chapter 111. Overview.....</b>	<b>297</b>
<b>Chapter 112. Before you begin.....</b>	<b>298</b>
<b>Chapter 113. Request Structure.....</b>	<b>299</b>
<b>Chapter 114. Response Structure.....</b>	<b>300</b>
<b>Chapter 115. Status codes.....</b>	<b>301</b>
<b>Chapter 116. Sample Request/Response.....</b>	<b>303</b>
Sample Request:.....	303
Sample Response:.....	303
<b>Chapter 117. Overview.....</b>	<b>304</b>
<b>Chapter 118. Before you begin.....</b>	<b>305</b>
<b>Chapter 119. Request Structure.....</b>	<b>306</b>
<b>Chapter 120. Response Structure.....</b>	<b>308</b>
<b>Chapter 121. Status Codes.....</b>	<b>309</b>
<b>Chapter 122. Sample Request/Response.....</b>	<b>310</b>
Sample Request:.....	310
Sample Response:.....	310

<b>Chapter 123. Overview.....</b>	<b>312</b>
<b>Chapter 124. Before you begin.....</b>	<b>313</b>
<b>Chapter 125. Request Structure.....</b>	<b>314</b>
<b>Chapter 126. Response Structure.....</b>	<b>332</b>
<b>Chapter 127. Status Codes.....</b>	<b>333</b>
<b>Chapter 128. Sample Request/Response.....</b>	<b>335</b>
Sample Request:.....	335
Sample Response:.....	336
Sample Request/Response For Apache Server.....	336
Sample Request:.....	336
Sample Response.....	337
Sample Request/Response for IPlanet Server.....	337
Sample Request.....	337
Sample Response.....	338
Sample Request/Response for Tomcat Server.....	338
Sample Request.....	338
Sample Response.....	339
Sample Request/Response for Weblogic Server.....	340
Sample Request.....	340
Sample Response.....	340
Sample Request/Response for Websphere Server.....	341
Sample Request.....	341
Sample Response.....	341
Sample Request.....	342
Sample Response.....	342
<b>Chapter 129. Overview.....</b>	<b>344</b>
<b>Chapter 130. Before you begin.....</b>	<b>345</b>
<b>Chapter 131. Request Structure.....</b>	<b>346</b>
<b>Chapter 132. Response Structure.....</b>	<b>347</b>

<b>Chapter 133. Status Codes.....</b>	<b>348</b>
<b>Chapter 134. Sample Request/Response.....</b>	<b>350</b>
Sample Request:.....	350
Sample Response:.....	350
<b>Chapter 135. Overview.....</b>	<b>351</b>
<b>Chapter 136. Approve a Request.....</b>	<b>352</b>
Request Structure.....	352
Response Structure.....	353
Status codes.....	354
Sample Request/Response.....	354
Sample Request:.....	354
Sample Response:.....	355
<b>Chapter 137. Implement Request.....</b>	<b>356</b>
Request Structure.....	356
Response Structure.....	357
Status codes.....	358
Sample Request/Response.....	358
Sample Request:.....	358
Sample Response:.....	359
<b>Chapter 138. Overview.....</b>	<b>360</b>
<b>Chapter 139. Request Structure.....</b>	<b>361</b>
<b>Chapter 140. Response Structure.....</b>	<b>362</b>
<b>Chapter 141. Status Codes.....</b>	<b>364</b>
<b>Chapter 142. Overview.....</b>	<b>365</b>
<b>Chapter 143. Request Structure.....</b>	<b>366</b>
<b>Chapter 144. Response Structure.....</b>	<b>367</b>
<b>Chapter 145. Status Codes.....</b>	<b>368</b>
<b>Chapter 146. Push and Bind Certificate to a ADC Profile.....</b>	<b>369</b>
Before You Begin.....	369

Fetch available profiles API.....	369
Request Structure.....	372
Response for Push and Bind ASYNC API.....	389
<b>Chapter 147. Push and Bind Certificate to ADC Devices.....</b>	<b>391</b>
Before you begin.....	391
Fetch available profiles API.....	391
Request for Push and Bind SYNC API.....	394
Response for Push and Bind SYNC API.....	408
<b>Chapter 148. Rollback to ADC profile.....</b>	<b>416</b>
Before you begin.....	416
Request Structure.....	416
Response Structure.....	417
Status Codes.....	418
Sample Request/Response.....	419
Sample Request:.....	420
Sample Response:.....	420
<b>Chapter 149. Overview.....</b>	<b>421</b>
<b>Chapter 150. Before you begin.....</b>	<b>422</b>
<b>Chapter 151. Request Structure.....</b>	<b>423</b>
<b>Chapter 152. Response Structure.....</b>	<b>425</b>
<b>Chapter 153. Status codes.....</b>	<b>426</b>
<b>Chapter 154. Sample Request/Response.....</b>	<b>428</b>
Sample Request:.....	428
Sample Response:.....	428
<b>Chapter 155. Overview.....</b>	<b>429</b>
<b>Chapter 156. Before you begin.....</b>	<b>430</b>
<b>Chapter 157. Request Structure.....</b>	<b>431</b>
<b>Chapter 158. Response Structure.....</b>	<b>433</b>
<b>Chapter 159. Status Codes.....</b>	<b>434</b>

<b>Chapter 160. Sample Request/Response.....</b>	<b>436</b>
Sample Request:.....	436
Sample Response:.....	436
<b>Chapter 161. Glossary.....</b>	<b>437</b>

# Preface

## Revision History

Revision	Description	Date
2.0	Updated release of document for release 2022.1.0 FP2 Beta	November 2022
1.0	Initial release of document for release 2022.1.0	June 2022

## About this Guide

This guide documents the detailed description of the APIs used to enable the CLM functionality for the SaaS-based deployment of CERT+.

## Audience

This guide is intended for AppViewX's customers looking to modify configurations via the API.

## Text Conventions

The following text conventions are used in this document:

Convention	Description
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>codeblock</code>	Indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

# Chapter 1: Overview

# Chapter 2: Prerequisites

- [Overview-Prerequisites](#)
- [API Access with User Account](#)
- [API Access with Service Account](#)

## Overview-Prerequisites

For accessing APIs, AppViewX supports the following two types of accounts:

- **User Account:** These are accounts used by real users.
- **Service Account:** Service accounts support the oAuth standard for a secured and standard way of accessing APIs. Therefore, AppViewX recommends that these accounts should be used by system services such as web servers, automation tools, and so on for accessing APIs.

## API Access with User Account

For accessing the AppViewX APIs with a user account, you will be required to get a session ID by providing your username and password to the **login** API. This session ID will then be used for accessing the other APIs.



**Note:** While you can use your username and password directly for accessing the APIs, instead of the session ID, it is not a recommended practice.

- [Retrieve Session ID using Login API](#)
- [Use Session ID in the Header for Further API Calls](#)

## Retrieve Session ID using Login API

- **URL:** `http://<Hostname>:31443/avxapi/login?gwsouce=external`
- **Method:** POST
- **Response format:** JSON
- **Requires authentication:** Yes
- **Request timeout period:** 15 minutes
- [Parameters](#)
- [Possible Response Message and Code](#)

- [Sample Request](#)
- [Sample Response](#)

## Parameters

Param Type	Name	Description	Field Type
Header	username**	Login username	String
Header	password**	Password for the username	String
Header	Contnt-Type**	Specifies the nature of the data in the payload (The value of the param should be "application/json")	String
Query	gwsource**	Source from which the request is triggered (for example, web, external)	String

## Possible Response Message and Code

HTTP Code	Description	Response Message
400	Bad Request	Login failed. Invalid credentials.
200	OK	Login successful.
401	Unauthorized	Authentication failed. The reason - Invalid credentials

## Sample Request

- **URL:** http://<Hostname>:31443/avxapi/login?gwsource=external
- **Headers:**

```
username: admin
password: AppViewX@123
Content-Type: application/json
grant_type=client_credentials
```

- **Request payload:**

```
{}
```

## Sample Response

```
{
  "response":
  "status": "SUCCESS",
  "appStatusCode": null,
  "statusDescription": null,
  "sessionId": "ce7f1a14-2bf9-4e4a-89a8-bc780a255813", "availableLoginAttemptCount": 10
},
"message": null,
"appStatusCode": null,
"tags": null,
"headers": null
}
```

## Use Session ID in the Header for Further API Calls

- **Example:** Add a role using API with `sessionId`
- **URL:** `http://<Hostname>:31443/avxapi/login?gwsource=external`
- **Method:** POST
- [Parameters](#)
- [Possible Response Message and Code](#)
- [Sample Request](#)
- [Sample Response](#)

## Parameters

Param Type	Name	Description	Field Type
Header	sessionId**	Use sessionId retrieved from login API	string
Query	gwkey	Tenant Key. Need only incase of multitenant installation	string

Param Type	Name	Description	Field Type
Query	gwsource**	Source from which the request is triggered (for example, web, external)	string
Payload	name **	Name of the role to be added	string
Payload	description	Description for the role to be added	string

## Possible Response Message and Code

HTTP Code	Description	Response Message
400	Bad Request	Role name cannot be left empty or blank.
201	Created	Role added successfully
209	Conflict	Role Name Already Exists

## Sample Request

- **URL:** http://<Hostname>:31443/avxapi/role?gwsource=external
- **Headers:**

```
sessionId: ce7f1a14-2bf9-4e4a-89a8-bc780a255813
```

- **Request payload:**

```
{
  "payload": {
    "name": "role_1",
    "description": "Adding a new role"
  }
}
```

## Sample Response

```
{
  "response": "Role added successfully",
  "message": "Role added successfully",
  "appStatusCode": null,
}
```

```

"tags": null,
"headers": null
}

```

## API Access with Service Account

For accessing APIs using a service account, you will be required to get an access token by providing the client ID and the client secret to the `get-service-token` API.

- [Retrieve Access Token using get-service-token API](#)
- [Use Access Token in the Header for Further API Calls](#)

### Retrieve Access Token using get-service-token API

- **URL:** `http://<Hostname>:31443/avxapi/acctmgmt-get-service-token?gwsource=external`
- **Method:** POST
- **Response format:** JSON
- **Requires authentication:** Yes
- **Request timeout period:** 15 minutes
- [Parameters](#)
- [Possible Response Message and Code](#)
- [Sample Request](#)
- [Sample Response](#)

### Parameters

Param Type	Name	Description	Field Type
Header	client_id**	Client ID of the Service Account	String
Header	client_secret**	Client Secret of the Service Account	String
Header	Contnt-Type**	Specifies the nature of the data in the payload (The value of the param should be "application/json")	String



```

NGJiOTA4MDQ4liwiaXNzljoiYXZ4liwiZXhwIjoxNjUwMzY5MzY3LjNcmFudCB0eXBlljoiY2xpZW50X2NyZWRIbnRpYWxzIn0.HZnkuUEjXleqJWqpqiNWFHqIDi7
GYf4cWx6VwbjGD_0",
"message": null,
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## Use Access Token in the Header for Further API Calls

- **Example:** Add a role using API with access token
- **URL:** `http://<Hostname>:31443/avxapi/role?gwsouce=external`
- **Method:** POST
- [Parameters](#)
- [Possible Response Message and Code](#)
- [Sample Request](#)
- [Sample Response](#)

## Parameters

Param Type	Name	Description	Field Type
Header	Token**	Use token retrieved from login API	string
Query	gwkey	Tenant Key. Need only incase of multitenant installation	string
Query	gwsouce**	Source from which the request is triggered (for example, web, external)	string
Payload	name **	Name of the role to be added	string
Payload	description	Description for the role to be added	string

## Possible Response Message and Code

HTTP Code	Description	Response Message
400	Bad Request	Role name cannot be left empty or blank.

HTTP Code	Description	Response Message
201	Created	Role added successfully
209	Conflict	Role Name Already Exists

## Sample Request

- **URL:** http://<Hostname>:31443/avxapi/role?gwsorce=external

- **Headers:**

Token:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJwbGF0Zm9ybSIsImF1ZCI6ImF1dG8iLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

jBj

```
NGJiOTA4MDQ4liwiaXNzljoiYXZ4liwiZXhwIjoxNjUwMzY5MzY3LCJmcmFudCB0eXBlljoiY2xpZW50X2NyZWRIbnRpYWxzIn0.HZnkuUEjXleqJWqppiNWFHql
```

DI7 GYf4cWx6VwbjGD\_0

- **Request payload:**

```
{
  "payload": {
    "name": "role_1",
    "description": "Adding a new role"
  }
}
```

## Sample Response

```
{
  "response": "Role added successfully",
  "message": "Role added successfully",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

# Chapter 3: Overview

Authenticate into the AppViewX application with the username and password.

# Chapter 4: Request Structure

URL: /login

Type: POST

Param Type	Name	Description	Field Type	Constraints
Header	username	AppViewX login username	String	NA
Header	password	AppViewX login password	String	NA
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
Query	gwkey	Tenant Key. Needed only in case of multi-tenant installations	NA	NA
Query	gwsouce	Source from which the request is triggered (for example: web, external)	String	NA

## Chapter 5: Response Structure

200 OK, returns a string of type application/JSON with the following body params:

Name	Description	Field Type
response	Contains the response for the session activation with the authentication/ session ID.	Key Value Pair
message	Success message along with the objectIds or failure description in case of error.	String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	Key value pair

## Chapter 6: Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Login successful	NA
400 Bad request		Username or password cannot be null or empty	Check and ensure if a non-null/non-empty value is given in the header field for username or password
401 Unauthorized		Login failed. Invalid credentials.	Check and ensure if the username and password provided are valid and correct.

## Chapter 7: Use Case

Login to the application with the username `appviewxuser` and password `appviewxpassword`. The session id received as response in the example below is `ce7f1a14-2bf9-4e4a-89a8-bc780a255813`

## Chapter 8: Sample Response:

```
{
  "response":
  {
    "status": "SUCCESS",
    "appStatusCode": null,
    "statusDescription": null,
    "sessionId": "ce7f1a14-2bf9-4e4a-89a8-bc780a255813",
    "availableLoginAttemptCount": null
  },
  "message": null,
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

## Chapter 9: Overview

Create a new certificate in async mode. In async mode, certificate generation is requested and request-id is provided in the response. The API will submit a request to create a certificate in async mode. Request payload varies depending on the Certificate Authority (CA) with which enrollment is requested. Once the create certificate API has been triggered successfully, the response will contain the resourceid, which can be used to trigger the [search](#) API to fetch the created certificate. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 10: Before you begin

Before attempting to enroll certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- Certificate group should have been created and associated with the policy. If not, the certificate will be created under the Default group.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 11: Request Structure

URL: /certificate/create

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered. (E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
csrGenerationSource	No	Specifies where the CSR is to be generated.	String	<p><b>Possible values:</b> appviewx, HSM, ENDPOINT, uploadCSR</p> <p><b>Default value:</b></p>

Name	Mandatory	Description	Field Type	Constraints
				appviewx
caConnectorInfo	Yes	Details related to Certificate Authority and CSR Parameters.	Any of the following - <ul style="list-style-type: none"> <li>• <a href="#">Sectigo</a> caConnectorInfo</li> <li>• <a href="#">DigiCert</a> caConnectorInfo</li> <li>• <a href="#">InCommon</a> caConnectorInfo</li> <li>• <a href="#">AppViewX</a> caConnectorInfo</li> <li>• <a href="#">EJBCA</a> caConnectorInfo</li> <li>• <a href="#">GlobalSign</a> caConnectorInfo</li> <li>• <a href="#">Trustwave</a> caConnectorInfo</li> <li>• <a href="#">Entrust</a> caConnectorInfo</li> <li>• <a href="#">Symantec</a> caConnectorInfo</li> <li>• <a href="#">LetsEncrypt</a> caConnectorInfo</li> <li>• <a href="#">GoDaddy</a> caConnectorInfo</li> <li>• <a href="#">Google</a> caConnectorInfo</li> <li>• <a href="#">Amazon</a> caConnectorInfo</li> <li>• <a href="#">Microsoft</a> Enterprise caConnectorInfo</li> </ul>	NA

Name	Mandatory	Description	Field Type	Constraints
			<ul style="list-style-type: none"> <li>Microsoft Standalone caConnectorInfo</li> <li>OpenTrust caConnectInfo</li> </ul>	
certificateGroup	No	Specifies the group under which the created certificate needs to be tagged.	certificateGroup	<b>Default certificateGroup:</b> Default
deviceDetails	No	Details on the device/ endpoint in which CSR will be generated.	deviceDetails	Required if the csrGenerationSource is <b>ENDPOINT</b> .
certificateHSMDetails	No	Details on the Hardware Security Module (HSM) device.	certificateHSMDetails	Required if the csrGenerationSource is <b>HSM</b> .
uploadCsrDetails	No	Details on the CSR.	uploadCsrDetails	Required if the csrGenerationSource is <b>uploadCSR</b> .
certificateFormat	No	Certificate format download details.	certificateFormat	NA

## certificateGroup

Name	Mandatory	Description	Field Type	Constraints
name	No	Specifies the group under which the created certificate needs to be tagged.	String	Group must already be present in AppViewX.

## deviceDetails

Name	Mandatory	Description	Field Type	Constraints
category	Yes	Specifies the device category.	String	<b>Possible values:</b> ADC, Server, Firewall
vendor	Yes	Vendor for the chosen device. For example, <b>Apache</b> is a vendor for <b>Server</b> category.	String	NA
deviceName	Yes	Name of the device as per AppViewX Device Inventory.	String	NA
csrFileName	Yes	Name of the CSR file that will be generated in the device.	String	NA
keyFileName	Yes	Name of the Key file that will be generated in the device.	String	NA
attributes	No	Additional attributes related to device.	attributes	NA

attributes

Name	Mandatory	Description	Field Type	Constraints
csrLocation	No	Location in the device where CSR will be created.	String	Required if <b>deviceDetails.category - Server</b> and <b>deviceDetails.vendor - Tomcat</b>
tenant	No	Name of the partition in the AVI device.	String	NA
partition	No	Name of the partition in the device.	String	Required if deviceDetails.category - <b>Firewall</b> and deviceDetails.vendor - <b>Fortinet</b>

certificateHSMDetails

Name	Mandatory	Description	Field Type	Constraints
type	Yes	Type of the HSM device.	String	<b>Possible values:</b> ADC, hsm

Name	Mandatory	Description	Field Type	Constraints
keyReference	Yes	Reference name for the key that will be mapped by the HSM device.	String	NA
hsmSettings	Yes	Configuration details for the HSM device.	hsmSettings	NA
vendor	Yes	Vendor for the chosen device. For example, F5.	String	NA
deviceName	Yes	Name of the device as per AppViewX Device Inventory.	String	NA

## hsmSettings

Name	Mandatory	Description	Field Type	Constraints
vendorType	Yes	Category of the vendor.	String	<b>Possible values:</b> Safenet, Thales, Fortanix
vendorSpecificSettings	Yes	Settings related to HSM vendor.	HSM vendorSpecificSettings	NA

## HSM vendorSpecificSettings

Name	Mandatory	Description	Field Type	Constraints
moduleId	No	Module Id	String	Applicable if hsmSettings.vendorType is <b>Thales</b> .

## uploadCsrDetails

Name	Mandatory	Description	Field Type	Constraints
category	Yes	Certificate category	String	<b>Possible values:</b> Server, Client, Code Signing
csrContent	Yes	The CSR content for certificate enrollment request.	String	NA

## certificateFormat

Name	Mandatory	Description	Field Type	Constraints
format	Yes	Certificate download format	String	Refer to the <a href="#">Possible values for Download Format</a> .
password	Yes	The field is mandatory for some parameters.	String	NA

## Possible values for Download Format

Certificate Extension	Value to be provided in payload	Password Required
.crt	CRT	No
.cert	CERT	No
.cer	CER	No
.pem	PEM	No
.der	DER	No
.cer	DERCER	No
.p7b	P7B	No
.p7c	P7C	No
.pk8	PK8	No
.pk12	PK12	Yes
.pfx	PFX	Yes
.jks	JKS	Yes

# Chapter 12: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the enrollment request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Open workorder request Id.	String

## Chapter 13: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Certificate submission triggered successfully.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>.	Check and ensure that valid value is provided for <<field name>> field in the request.
400 Bad Request	INVALID_CSR_GEN_SOURCE	Invalid csr generation source specified.	Check and ensure that valid value is provided for csrGenerationSource field in the request.
417 Bad Request	READ_GROUP	Group is given with read permission, so cannot perform any operation.	Please ensure that the certificateGroup has read write permission.
404 Expectation failed	GROUP_NOT_FOUND	Invalid group name specified.	Check and ensure that valid value is provided for the certificateGroup section in the request.
404 Not Found	CERT-CA-0001	CA settings are not available.	Check and ensure that valid value is provided for the caSettingName field in the request.

## Chapter 14: CA Specific Information

- Sectigo
- DigiCert
- InCommon
- AppViewX
- EJBCA
- GlobalSign
- Trustwave
- Entrust
- Symantec
- Let's Encrypt
- GoDaddy
- Google
- Amazon
- Microsoft Enterprise
- Microsoft Standalone
- OpenTrust Request Objects
- Sample Request/Response
- Entrust MPKI

### Sectigo

- Sectigo Request Objects
- Sample Request/Response

### Sectigo Request Objects

Sectigo caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate	String	Value should be Comodo Certificate Manager
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days	Boolean	Should be disabled if autoRegenerateEnabled is true.
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request	Integer	Value must be provided if isAutoRenewal is true
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days	Boolean	Should be disabled if isAutoRenewal is true.
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request	Integer	Value must be provided if autoRegenerateEnabled is true
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority	String	NA
certificateType	Yes	Name of the certificate product offered by the Certificate Authority	String	NA
description	No	Note about the certificate	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR	Sectigo csrParameters	NA
genericFields	No	Custom fields configured for the CA/Customer	genericFields	NA
vendorSpecificDetails	No	Data specific to the Sectigo vendor	Sectigo vendor	NA

Name	Mandatory	Description	Field Type	Constraints
			<a href="#">Specific Details</a>	
validityUnitValue	Yes	Validity value based on validityUnit. For example, if the expected validity is 1 year and validityUnit is Months, then the validityUnitValue should be 12	Integer	If validityUnit is not provided, then the validityUnitValue must be provided in days.
validityInDays	No	Specifies the validity in days	Integer	NA
validityUnit	No	Specifies the unit in which the validityUnitValue will be specified	String	Possible values - days, months, years
name	No	Specifies the name for the caConnector	String	NA

## Sectigo vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
serverType	Yes	Server category for which certificate is requested. For example, AOL	String	NA

## Sectigo csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as 'Strict'

Name	Mandatory	Description	Field Type	Constraints
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array of String	Possible values - Server, Client, Code Signing, Email
ellipticCurve	No	If the keyType chosen is EC, then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
enhancedSANTypes	No	Subject alternative names for the certificate.	<a href="#">Sectigo</a> <a href="#">enhancedSANTypes</a>	NA

[Sectigo enhancedSANTypes](#)

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo":
  {
    "certificateAuthority": "Comodo Certificate Manager",
    "isAutoRenewal": "true",
    "renewBefore": "30",
    "autoRegenerateEnabled": false,
    "caSettingName": "Sectigo_Test",
    "certificateType": "EliteSSL Certificate",
    "description": "",
    "csrParameters":
    {
      "commonName": "testcert2n.ams1907.com",
      "hashFunction": "SHA256",
      "keyType": "RSA",
      "bitLength": "2048",
      "certificateCategories":
      [
        "Server",
        "Client"
      ],
      "ellipticCurve": "",
      "enhancedSANTypes":
      {
        "dNSNames": []
      }
    }
  }
}
```

```

}
},
"genericFields":
{
"device_name_Comodo Certificate Manager": "TestDevice",
"vs_ip_Comodo Certificate Manager": "192.168.192.168"
},
"vendorSpecificDetails":
{
"serverType": "Apache-SSL (Ben-SSL, not Stronghold)"
},
"validityUnitValue": "365",
"validityInDays": "365",
"validityUnit": "days"
},
"certificateGroup":
{
"name": "Default"
},
"certificateFormat":
{
"format": "PEM",
"password": ""
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response":
{
"resourceId": "5e98047171eee023092ba006",
"requestId": "242"
},
"message": "Certificate submission triggered successfully."
}

```

```

"appStatusCode": null,
"tags": {},
"headers": null
}

```

## DigiCert

- [DigiCert Request Objects](#)
- [Sample Request/Response](#)

### DigiCert Request Objects

DigiCert caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>DigiCert</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in	String	

Name	Mandatory	Description	Field Type	Constraints
		AppViewX for the chosen Certificate Authority.		
divisionId	Yes	Digicert division id.	String	NA
certificateType	Yes	Certificate product type offered by the Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">DigiCert</a> <a href="#">csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
vendorSpecificDetails	No	Data specific to the DigiCert vendor.	<a href="#">DigiCert</a> <a href="#">vendorSpecificDetails</a>	NA
validityUnitValue	No	Validity value based on <a href="#">validityUnit</a> . For example, if the expected validity is <b>1</b> year and <a href="#">validityUnit</a> is <b>Months</b> , then the <a href="#">validityUnitValue</a> should be <b>12</b> .	Integer	Either <a href="#">validityUnit</a> and <a href="#">validityUnitValue</a> must be provided or <a href="#">validityInDays</a> must be provided.
validityInDays	No	Specifies the validity in days.	Integer	NA
validityUnit	No	Specifies the unit in which the <a href="#">validityUnitValue</a> will be specified.	String	Possible values - days, months, years.

## DigiCert csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common

Name	Mandatory	Description	Field Type	Constraints
				name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
streetAddress	No	Street address where the organization is located.	String	NA
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
postalCode	No	Postal code for the organization address.	String	NA
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.	String	Default value - the first value will be chosen from the policy.

Name	Mandatory	Description	Field Type	Constraints
		Should be chosen from the possible values configured in the Certificate Policy.		
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	DigiCert enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## DigiCert vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
caCertId	No	Certificate ID for the chosen Certificate Authority.	String	Required for DigiCert private product types. For example, 'Private SSL Plus'.
serverType	No	Server category for which certificate is requested. For example, Apache.	String	NA

## DigiCert enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA
iPAddresses	No	List of IP address to be considered as subject alternative names.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "ENDPOINT",
  "deviceDetails":
  {
    "vendor": "Apache",
    "category": "Server",
    "deviceName": "Apache server",
    "csrFileName": "test",
    "keyFileName": "test",
```

```

"attributes":
{
  "csrLocation": ""
}
},
"caConnectorInfo":
{
  "certificateAuthority": "DigiCert",
  "isAutoRenewal": false,
  "autoRegenerateEnabled": true,
  "regenerateBeforeInDays": "30",
  "caSettingName": "DigiCert",
  "divisionId": "80312",
  "certificateType": "Private SSL Plus",
  "description": "", "csrParameters":
{
  "commonName": "testcert8g.appviewx.plus",
  "organization": "AppViewX Inc.",
  "mailAddress": "test@test.com",
  "hashFunction": "SHA256",
  "keyType": "RSA",
  "bitLength": "2048",
  "certificateCategories": ["Server", "Client"],
  "ellipticCurve": "", "enhancedSANTypes":
{
  "dNSNames": []
}
},
  "genericFields":
{
    "device_name_DigiCert": "test_device",
    "vs_ip_DigiCert": "xxx.xxx.xxx.xxx"
  },
  "vendorSpecificDetails":
{
    "caCertId": "741B2D2F05C7F6543EFB",
    "serverType": "Apache" },

```

```

"validityUnitValue": "1",
"validityInDays": 365,
"validityUnit": "years"
},
"certificateGroup":
{
"name": "Default"
}
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response":
{
"resourceId": "5f4faf3e70040d33314f1142",
"requestId": "173"
},
"message": "Certificate submission triggered successfully.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## InCommon

- [InCommon Request Objects](#)
- [Sample Request/Response](#)

## InCommon Request Objects

InCommon caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>InCommon</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	NA
certificateType	No	Name of the certificate product offered by the Certificate Authority.	String	Must be provided for request with <b>InCommon</b> .
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<b>InCommon</b> <a href="#">csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
vendorSpecificDetails	No	Data specific to the Sectigo vendor	<b>InCommon</b> <a href="#">vendorSpecificDetails</a>	NA

Name	Mandatory	Description	Field Type	Constraints
validityInDays	Yes	Specifies the validity in days.	Integer	NA

## InCommon vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
serverType	Yes	Server category for which certificate is requested. For example, AOL.	String	NA

## InCommon csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.

Name	Mandatory	Description	Field Type	Constraints
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	InCommon enhancedSANTypes	NA

InCommon enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "InCommon",
    "isAutoRenewal": "false",
```

```

"autoRegenerateEnabled": true,
"regenerateBeforeInDays": "30",
"caSettingName": "Incommon_test_ca",
"certificateType": "EliteSSL Certificate",
"description": "",
"csrParameters": {
"commonName": "testcert.testdomain.com",
"hashFunction": "SHA256",
"keyType": "RSA",
"bitLength": "2048",
"certificateCategories": ["Server", "Client"],
"ellipticCurve": "",
"enhancedSANTypes":
"dNSNames": []
}
},
"genericFields": {
"device_name_InCommon": "test_device",
"vs_ip_InCommon": "192.168.142.162"
},
"vendorSpecificDetails": {
"serverType": "AOL"
},
"validityInDays": 365
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response":
{

```

```

"resourceId": "5f47b29938c8c041c2c7096b",
"requestId": "112
},
"message": "Certificate submission triggered successfully.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## AppViewX

- [AppViewX Request Objects](#)
- [Sample Request/Response](#)

### AppViewX Request Objects

AppViewX caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>AppViewX</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
caSettingName	Yes	Name of the <code>CASetting</code> that has been created in AppViewX for the chosen Certificate Authority.	String	NA
certificateProfileName	No	Determines the category of certificate that is being requested.	String	Possible values - Server, Client, CodeSigning  Default value - Server.
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">AppViewX</a> <a href="#">csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
validityInDays	Yes	Specifies the validity in days.	Integer	NA
validityUnitValue	No	Validity value based on validityUnit. For example, if the expected validity is 1 year and validityUnit is Months, then the validityUnitValue should be 12	Integer	Either validityUnit and validityUnitValue must be provided or validityInDays must be provided
validityUnit	No	Specifies the unit in which the validityUnitValue will be specified	String	Possible values - days, months, years
name	No	Specifies the name for the caConnector	String	NA

## AppViewX csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common

Name	Mandatory	Description	Field Type	Constraints
				name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from	String	Default value - the first value will

Name	Mandatory	Description	Field Type	Constraints
		the possible values configured in the Certificate Policy.		be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	AppViewX enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## AppViewX enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA

Name	Mandatory	Description	Field Type	Constraints
iPAddresses	No	List of IP address to be considered as subject alternative names.	Array of Sting	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "AppViewX",
    "isAutoRenewal": false,
    "autoRegenerateEnabled": true,
    "regenerateBeforeInDays": "30",
    "caSettingName": "AppViewX CA",
    "certificateProfileName": "Server",
    "description": "",
    "csrParameters": {
      "commonName": "testcert.testdomain.com",
      "organization": "AppViewX",
      "organizationUnit": "PE",
      "locality": "Plano",
      "state": "Texas",
      "country": "US",
      "mailAddress": "test@test.com",
      "hashFunction": "SHA256",
      "keyType": "RSA",
      "bitLength": "2048",
      "certificateCategories": ["Server", "Client"],
      "ellipticCurve": ""
    }
  }
}
```

```

"enhancedSANTypes": {
  "dNSNames": ["*.testcert4d.appviewx.com, testcert4b1.avx.com"],
  "IPAddresses": ["xxx.xxx.xxx.xxx"]
}
},
"genericFields": {
  "device_name_AppViewX": "test_device",
  "vs_ip_AppViewX": "192.168.142.162"
},
"validityInDays": 365
},
"certificateGroup": {
  "name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
  "response": {
    "resourceId": "5f4842fd70040d33314f0748",
    "requestId": "135"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}

```

## EJBCA

- [EJBCA Request Objects](#)
- [Sample Request/Response](#)

### EJBCA Request Objects

EJBCA caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Ejbca</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	Prerequisite - End entity profiles should have been fetched and updated in the specified CA setting.

Name	Mandatory	Description	Field Type	Constraints
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">EJBCA</a> <a href="#">csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
vendorSpecificDetails	Yes	Details specific to the EJBCA vendor.	<a href="#">EJBCA</a> <a href="#">vendorSpecificDetails</a>	NA
customAttributes	No	Custom fields configured for EJBCA	<a href="#">EJBCA</a> <a href="#">customAttributes</a>	NA
name	No	Specifies the name for the caConnector	String	NA

## EJBCA csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located.This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server,

Name	Mandatory	Description	Field Type	Constraints
				Client,  Code Signing,  Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	EJBCA enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## EJBCA vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
userName	No	User name configured at EJBCA end.	String	NA
endEntityProfileName	Yes	Name of the end entity profile configured at CA end.	String	NA
issuerCommonName	Yes	Common name of the Issuer configured at CA end.	String	NA
certificateProfileName	Yes	Name of the Certificate profile chosen from the certificate profiles available for the selected endEntityProfile.	String	NA

## EJBCA customAttributes

Name	Mandatory	Description	Field Type	Constraints
postalCode	No	postalCode is given as an example. This could be any customAttribute configured for EJBCA by the customer.	String	NA

## EJBCA enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	Should be a valid domain name.
rfc822Names	No	Email addresses to be considered as Subject Alternative Names.	Array of Sting	Must be valid email addresses.
iPAddresses	No	IP addresses to be considered as Subject Alternative Names.	Array of Sting	Must be valid ip addresses.
uniformResourceIdentifiers	No	URIs to be considered as Subject Alternative Names.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "EjbcA",
    "isAutoRenewal": false,
    "autoRegenerateEnabled": true,
    "regenerateBeforeInDays": "30",
    "caSettingName": "ejbca",
    "description": "",
    "csrParameters": {
```

```

"commonName": "testcert.testdomain.com",
"organization": "AppViewX",
"organizationUnit": "PE",
"locality": "Plano",
"state": "Texas",
"country": "US",
"mailAddress": "test@test.com",
"hashFunction": "SHA256",
"keyType": "RSA",
"bitLength": "2048",
"certificateCategories": ["Server", "Client"],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": ["testcert5.avx.com"],
"rfc822Names": ["test@testcert5.avx.com"],
"IPAddresses": ["xxx.xxx.xxx.xxx"],
"uniformResourceIdentifiers": ["http://testcert5.avx.com"]
}
},
"genericFields": {
"device_name_Ejbca": "test_device",
"vs_ip_Ejbca": "192.168.142.162"
},
"vendorSpecificDetails": {
"userName": "",
"endEntityProfileName": "APPVIEWX PROFILE",
"issuerCommonName": "Ejbca New Intermediate CA",
"certificateProfileName": "SERVER PROFILE"
},
"customAttributes": {}
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "resourceId": "5f4e5c2e70040d33314f0e9d",
    "requestId": "147"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## GlobalSign

- [GlobalSign Request Objects](#)
- [Sample Request/Response](#)

## GlobalSign Request Objects

GlobalSign caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>GlobalSign</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on <code>renewBefore</code> days.	Boolean	Should be disabled if <code>isAutoRenewal</code> is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if <code>autoRegenerateEnabled</code> is <b>true</b> .
caSettingName	Yes	Name of the <code>CASetting</code> that has been created in AppViewX for the chosen Certificate Authority.	String	NA
certificateType	Yes	Certificate product type offered by the Certificate Authority.	String	Possible values - Alpha SSL, Domain SSL, Organization SSL, Extended SSL.
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">GlobalSign</a> <a href="#">csrParameters</a>	NA
vendorSpecificDetails	No	Details specific to the GlobalSign vendor.	<a href="#">GlobalSign</a> <a href="#">vendorSpecificDetails</a>	Mandatory for the following certifiactTypes - Alpha SSL, Domain SSL, Extended SSL
validityInDays	Yes	Specifies the validity in days.	Integer	NA
name	No	Specifies the name for the <code>caConnector</code>	String	NA

## GlobalSign csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
streetAddress	No	Street address where the organization is located.	String	NA
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.	String	Default value - the first value will

Name	Mandatory	Description	Field Type	Constraints
		Should be chosen from the possible values configured in the Certificate Policy.		be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	GlobalSign enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## GlobalSign vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
domainAdminMailId	Yes	Local-part of the email id of the administrator/ group responsible for issuing/approving certificate requests. For example, <b>hostmaster@test.com</b> here <b>hostmaster</b> is the local-part.	String	NA
incorporatingAgencyRegistrationNumber	No	Registration no of the company.	String	Required if the certificateType is <b>extendedSSL</b> .
designation	No	Designation of the authorized signer.	String	Required if the certificateType is <b>extendedSSL</b> .
businessCategory	No	Category code for the business type.	String	Required if the certificateType is <b>extendedSSL</b> .

## GlobalSign enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array String	Applicable if the certificateType is - of Domain SSL/OrganizationSSL/ ExtendedSSL.
iPAddresses	No	List of IP address to be considered as subject alternative names.	Array String	Applicable if the certificateType is - of OrganizationSSL.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "GlobalSign",
    "autoRegenerateEnabled": false,
    "caSettingName": "Globalsign",
    "certificateType": "Alpha SSL",
    "description": "",
    "csrParameters": {
      "commonName": "testcert.testdomain.com",
      "organization": "AppViewX",
      "organizationUnit": "PE",
      "streetAddress": "Park Street",
      "locality": "Plano",
      "state": "Texas",
      "country": "US",
      "postalCode": "73301",
      "mailAddress": "test@test.com",
      "hashFunction": "SHA256",
      "keyType": "RSA",
      "bitLength": "2048",
      "certificateCategories": ["Server", "Client"],
      "ellipticCurve": "",
      "enhancedSANTypes": {
        "dNSNames": []
      }
    }
  },
  "vendorSpecificDetails": {
    "domainAdminMailId": "hostmaster",
    "incorporatingAgencyRegistrationNumber": ""
  }
}
```

```

"designation": "",
"businessCategory": "PO",
"orderId": ""
},
"validityInDays": 183
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response": {
"resourceId": "5f4e876e70040d33314f0f0b",
"requestId": "149"
},
"message": "Certificate submission triggered successfully.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## Trustwave

- [Trustwave Request Objects](#)
- [Sample Request/Response](#)

## Trustwave Request Objects

Trustwave caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Trustwave</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	NA
certificateType	Yes	Certificate product type offered by the Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">Trustwave</a> <a href="#">csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
validityInDays	Yes	Specifies the validity in days.	Integer	NA
name	No	Specifies the name for the caConnector	String	NA

Trustwave csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
streetAddress	No	Street address where the organization is located.	String	NA
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
postalCode	No	Postal code of the organization address.	String	NA
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	<a href="#">Trustwave enhancedSANTypes</a>	Value provided must be compliant with the Certificate Policy,

Name	Mandatory	Description	Field Type	Constraints
				if the policy is configured as <b>Strict</b> .

Trustwave enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "Trustwave",
    "isAutoRenewal": "true",
    "renewBefore": "30",
    "autoRegenerateEnabled": false,
    "caSettingName": "trustwave",
    "certificateType": "SecureTrust Organization Validation",
    "description": "",
    "csrParameters": {
      "commonName": "testcert.testdomain.com",
      "organization": "AppViewX",
      "organizationUnit": "PE",
      "streetAddress": "Park Street",
      "locality": "Plano",
      "state": "Texas",
      "country": "US",
      "postalCode": "75093",
```

```

"mailAddress": "test@test.com",
"hashFunction": "SHA256",
"keyType": "RSA",
"bitLength": "2048",
"certificateCategories": ["Server", "Client"],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": []
}
},
"genericFields": {
"device_name_Trustwave": "test_device",
"vs_ip_Trustwave": "xxx.xxx.xxx.xxx"
},
"validityInDays": 365
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response": {
"resourceId": "5f4e934570040d33314f0f1d",
"requestId": "151"
},
"message": "Certificate submission triggered successfully.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

# Entrust

- [Entrust Request Objects](#)
- [Sample Request/Response](#)

## Entrust Request Objects

Entrust caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Entrust</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	
divisionId	Yes	Digicert division id.	String	

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Certificate product type offered by the Certificate Authority.	String	
description	No	Note about the certificate.	String	
csrParameters	Yes	Parameters that are necessary for generating the CSR.	Entrust csrParameters	
genericFields	No	Custom fields configured for the CA/Customer.	genericFields	
vendorSpecificDetails	No	Data specific to the DigiCert vendor.	Entrust vendorSpecificDetails	
validityUnitValue	No	Validity value based on validityUnit. For example, if the expected validity is 1 year and validityUnit is <b>Months</b> , then the validityUnitValue should be <b>12</b> .	Integer	Either validityUnit and validityUnitValue must be provided or validityInDays must be provided.
validityInDays	No	Specifies the validity in days.	Integer	
customAttributes	No	Custom fields configured for Entrust.	Entrust customAttributes	
name	No	Specifies the name for the caConnector	String	

## Entrust csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the

Name	Mandatory	Description	Field Type	Constraints
				policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
streetAddress	No	Street address where the organization is located.	String	NA
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA,	String	Default value - the first value will

Name	Mandatory	Description	Field Type	Constraints
		DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.		be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	Entrust enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

Entrust vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
additionalEmails	No	Additional emailID to be associated with the request.	String	Should be a valid email address.

## Entrust customAttributes

Name	Mandatory	Description	Field Type	Constraints
email1	No	email1 is given as an example. This could be any customAttribute configured for Entrust by the customer. Multiple custom attributes could be configured.	String	NA

## Entrust enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA
iPAddresses	No	IP addresses to be considered as Subject Alternative Names.	Array of String	Must be valid ip addresses.
uniformResourceIdentifiers	No	URIs to be considered as Subject Alternative Names.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "Entrust",
    "isAutoRenewal": "true",
```

```

"renewBefore": "30",

"autoRegenerateEnabled": false,

"caSettingName": "Entrust",

"certificateType": "Standard",

"description": "",

"csrParameters": {

"commonName": "testcert.testdomain.com",

"mailAddress": "test@test.com",

"hashFunction": "SHA256",

"keyType": "RSA",

"bitLength": "2048",

"certificateCategories": ["Server", "Client"],

"ellipticCurve": "",

"enhancedSANTypes": {

"dNSNames": [],

"iPAddresses": [],

"uniformResourceIdentifiers": []

}

},

"genericFields": {

"device_name_Entrust": "test_device",

"vs_ip_Entrust": "xxx.xxx.xxx.xxx"

},

"vendorSpecificDetails": {

"additionalEmails": "test1@test.com"

},

"customAttributes": {

"email1": "test@test.com",

"text2": "ID12345",

"dropdown1": "Server Cert",

"number1": "1",

"date1": 1598985000000

},

"validityInDays": 90

},

"certificateGroup": {

```

```
"name": "Default"
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "resourceId": "5f4fcb1770040d33314f11f0",
    "requestId": "184"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Symantec

- [Symantec Request Objects](#)
- [Sample Request/Response](#)

## Symantec Request Objects

Symantec caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Symantec</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	NA
certificateType	Yes	Certificate product type offered by the Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">Symantec csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
vendorSpecificDetails	Yes	Details specific to the Symantec vendor.	<a href="#">Symantec vendorSpecificDetails</a>	NA
validityInDays	Yes	Specifies the validity in days.	Integer	NA
name	No	Specifies the name for the caConnector	String	NA

[Symantec csrParameters](#)

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located.This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
encryptedChallengePassword	Yes	Base64 encoded challenge password.	String	NA
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the <code>keyType</code> chosen is <b>EC</b> , then the <code>ellipticCurve</code> must be specified depending on the	String	Default value - the first value

Name	Mandatory	Description	Field Type	Constraints
		bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.		will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	Symantec enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## Symantec vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
serverType	Yes	Server category for which certificate is requested. For example, Red Hat.	String	Should be a valid email address.

## Symantec enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
```

```
"certificateAuthority": "Symantec",
"isAutoRenewal": "true",
"renewBefore": "30",
"autoRegenerateEnabled": false,
"caSettingName": "Symantec",
"certificateType": "Standard",
"description": "",
"csrParameters": {
  "commonName": "testcert.testdomain.com",
  "organization": "AppViewX",
  "organizationUnit": "PE",
  "locality": "Plano",
  "state": "Texas",
  "country": "US",
  "mailAddress": "test@test.com",
  "hashFunction": "SHA256",
  "encryptedChallengePassword": "QXBwVmld1hAMTiz",
  "keyType": "RSA",
  "bitLength": "2048",
  "certificateCategories": ["Server", "Client"],
  "ellipticCurve": "",
  "enhancedSANTypes": {
    "dNSNames": []
  }
},
"genericFields": {
  "device_name_Symantec": "test_device",
  "vs_ip_Symantec": "xxx.xxx.xxx.xxx"
},
"vendorSpecificDetails": {
  "serverType": "Red Hat"
},
"validityInDays": 365
},
"certificateGroup": {
  "name": "Default"
```

```
}
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "resourceId": "5f4fe53c70040d33314f122d",
    "requestId": "190"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Let's Encrypt

- [Lets Encrypt Request Objects](#)
- [Sample Request/Response](#)

## Lets Encrypt Request Objects

LetsEncrypt caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>LetsEncrypt</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	
description	No	Note about the certificate.	String	
csrParameters	Yes	Parameters that are necessary for generating the CSR.	LetsEncrypt csrParameters	
genericFields	No	Custom fields configured for the CA/Customer.	LetsEncryptgenericFields	
vendorSpecificDetails	Yes	Details specific to the Symantec vendor.	LetsEncrypt vendorSpecificDetails	
validityInDays	Yes	Specifies the validity in days.	Integer	
name	No	Specifies the name for the caConnector	String	

LetsEncrypt csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.

Name	Mandatory	Description	Field Type	Constraints
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	LetsEncrypt enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

LetsEncrypt genericFields

Name	Mandatory	Description	Field Type	Constraints
device_name_LetsEncrypt	No	Name of the device.	String	NA
vs_ip_LetsEncrypt	No	IP address of the application.	String	NA
f5_device	No	F5 device name/ IP.	String	NA
data_group_name	No	Data group name.	String	NA
infoblox_device	No	Infoblox device name/IP.	String	NA
validation_ip	No	Validation IP.	String	NA

#### LetsEncrypt vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
challengeType	Yes	Challenge validation type	String	Possible values:  DNS,  HTTP.

#### LetsEncrypt enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "LetsEncrypt",
    "isAutoRenewal": "false",
```

```

"autoRegenerateEnabled": true,
"regenerateBeforeInDays": "30",
"caSettingName": "LetsEncrypt",
"description": "",
"csrParameters": {
"commonName": "testcert.testdomain.net",
"hashFunction": "SHA256",
"keyType": "RSA",
"bitLength": "4096",
"certificateCategories": ["Server"],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": ["test.lab.appviewx.net"]
}
},
"genericFields": {
"device_name_LetsEncrypt": "",
"vs_ip_LetsEncrypt": "",
"f5_device": "",
"data_group_name": "",
"infoblox_device": "",
"validation_ip": ""
},
"vendorSpecificDetails": {
"challengeType": "DNS"
},
"validityInDays": 90
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values

## Sample Response:

```
{
  "response": {
    "resourceId": "5f510caa70040d33314f13d7",
    "requestId": "204"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## GoDaddy

- [GoDaddy Request Objects](#)
- [Sample Request/Response](#)

## GoDaddy Request Objects

GoDaddy caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>GoDaddy</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if <code>autoRegenerateEnabled</code> is <b>true</b> .
caSettingName	Yes	Name of the <code>CASetting</code> that has been created in AppViewX for the chosen Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">GoDaddy</a> <a href="#">csrParameters</a>	NA
vendorSpecificDetails	Yes	Details specific to the EJBCA vendor.	<a href="#">GoDaddy</a> <a href="#">vendorSpecificDetails</a>	NA
name	No	Specifies the name for the <code>caConnector</code>	String	NA

## GoDaddy csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located.This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values -Server,Code Signing.

Name	Mandatory	Description	Field Type	Constraints
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	GoDaddy enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## GoDaddy vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
firstName	No	User's First name configured at GoDaddy end.	String	NA
lastName	No	Last name of the user configured at CA end.	String	NA
emailAddress	No	Email address of the user to be configured at CA end.	String	Should be a valid email address.
phone	No	Phone number of the user.	String	NA
apiKey	Yes	API key to be used for the GoDaddy CA communication.	String	NA
apiSecret	Yes	API Secret to be used for the GoDaddy CA communication.	String	NA
customerNumber	Yes	Customer number obtained from user's GoDaddy portal.	String	NA

Name	Mandatory	Description	Field Type	Constraints
url	Yes	Base URL to be used for GoDaddy CA communication.	String	NA

## GoDaddy enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	Should be a valid domain name.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "certificateGroup": {
    "name": "Default"
  },
  "caConnectorInfo": {
    "name": "GoDaddy Deluxe UCC SSL SAN UCC OV SSL connector",
    "description": "",
    "certificateUuid": "",
    "certificateAuthority": "GoDaddy",
    "certificateType": "Deluxe UCC SSL - SAN (UCC_OV_SSL)",
    "csrParameters": {
      "commonName": "testgodaddy",
      "organization": "",
      "locality": "",
      "country": "",
      "enhancedSANTypes": {
        "dnsnames": [
```

```

"testgodaddy"
],
"dNSNames": [
"testgodaddy"
]
},
"organizationUnit": "",
"state": "",
"mailAddress": "",
"keyType": "RSA",
"bitLength": "1024",
"hashFunction": "SHA160",
"ellipticCurve": "",
"encryptedChallengePassword": "",
"certificateCategories": [
"Server",
"Client"
]
},
"renewBefore": 0,
"autoRegenerateEnabled": false,
"regenerateBeforeInDays": 0,
"inheritedFromGroup": false,
"caSettingName": "Godaddy",
"validityInDays": 365,
"vendorSpecificDetails": {
"slotSize": "10",
"orderId": ""
},
"whiteLabelledCA": false,
"existingUuid": "",
"validityUnit": "years",
"validityUnitValue": 1,
"certificateCategory": "Server and Client",
"isAutoRenewal": false
},
"csrGenerationSource": "appviewx",

```

```
"fields": []
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

Sample Response

```
{
  "response": {
    "resourceId": "60630a762be80b7ea8f03c06",
    "requestId": "155"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Google

- [Google Request Objects](#)
- [Sample Request/Response](#)

## Google Request Objects

Google caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Google</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if <code>autoRegenerateEnabled</code> is <b>true</b> .
caSettingName	Yes	Name of the <code>CASetting</code> that has been created in AppViewX for the chosen Certificate Authority.	String	NA
issuerName	Yes	Name of the Issuer configured at the CA end.	String	NA
issuerLocation	Yes	Location of the Issuer configured at the CA end.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">Google</a> <a href="#">csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
vendorSpecificDetails	Yes	Details specific to the Google vendor.	<a href="#">Google</a> <a href="#">vendorSpecificDetails</a>	NA
validityUnitValue	Yes	Validity value based on <code>validityUnit</code> . For example, if the expected validity is 1 year and <code>validityUnit</code> is <b>Months</b> , then the <code>validityUnitValue</code> should be <b>12</b> .	Integer	If <code>validityUnit</code> is not provided, then the <code>validityUnitValue</code> must be provided in <b>days</b> .
validityInDays	No	Specifies the validity in days.	Integer	NA
validityUnit	Yes	Specifies the unit in which the <code>validityUnitValue</code> will be specified.	String	Possible values - days, months, years.

Name	Mandatory	Description	Field Type	Constraints
name	No	Specifies the name for the caConnector	String	NA

## Google csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
EmailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	<a href="#">Google enhancedSANTypes</a>	Value provided must be compliant with the Certificate Policy,

Name	Mandatory	Description	Field Type	Constraints
				if the policy is configured as <b>Strict</b> .

## Google vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
certificateId	No	Unique name of the certificate maintained in Google CA.	String	If not filled, it will be filled with common name and the current timestamp.

## Google enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	Should be a valid domain name.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "Google",
    "autoRegenerateEnabled": false,
    "caSettingName": "Google",
    "issuerLocation": "us-east1",
    "issuerName": "AppViewX-Demo",
    "description": "",
    "csrParameters": {
```

```

"commonName": "testgooglecertificate",
"subjectAlternativeNames": [
"dNSNames"
],
"organization": "AppViewX",
"organizationUnit": "AppViewX",
"locality": "Texas",
"state": "Texas",
"country": "US",
"mailAddress": "test@appviewx.com",
"hashFunction": "SHA256",
"keyType": "RSA",
"bitLength": "2048",
"postalCode": "",
"certificateCategories": [
"Server",
"Client"
],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": [
"testgooglecertificate"
]
}
},
"genericFields": {
"device_name_Google": "",
"vs_ip_Google": ""
},
"vendorSpecificDetails": {
"certificateId": ""
},
"validityUnitValue": "1",
"validityInDays": 365,
"validityUnit": "years"
},
"certificateGroup": {

```

```
"name": "Default"
}
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "resourceId": "5f4e5c2e70040d33314f0e9d",
    "requestId": "148"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Amazon

- [Amazon Request Objects](#)
- [Sample Request/Response](#)

## Amazon Request Objects

Amazon caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Amazon</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .

Name	Mandatory	Description	Field Type	Constraints
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">Amazon csrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
name	No	Specifies the name for the caConnector	String	NA

## Amazon csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .

Name	Mandatory	Description	Field Type	Constraints
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	<a href="#">Amazon enhancedSANTypes</a>	Value provided must be compliant with the Certificate Policy,

Name	Mandatory	Description	Field Type	Constraints
				if the policy is configured as <b>Strict</b> .

## Amazon enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dnsNames	No	List of Subject Alternative names for the Certificate.	Array of String	Should be a valid domain name.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "Amazon",
    "isAutoRenewal": false,
    "autoRegenerateEnabled": false,
    "caSettingName": "Amazon",
    "certificateType": "",
    "description": "",
    "csrParameters": {
      "commonName": "testcert.testdomain.com",
      "hashFunction": "SHA256",
      "keyType": "RSA",
      "bitLength": "2048",
      "certificateCategories": [
        "Server",
        "Client"
      ]
    }
  }
}
```

```

],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": [
"testamazon.appviewx.com"
]
}
},
"genericFields": {
"device_name_Amazon": "",
"vs_ip_Amazon": ""
},
"vendorSpecificDetails": {
"orderId": ""
}
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response": {
"resourceId": "5f4e5c2e70040d33314f0e9d",
"requestId": "149"
},
"message": "Certificate submission triggered successfully.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## Microsoft Enterprise

- [Microsoft Enterprise Request Objects](#)
- [Sample Request/Response](#)

### Microsoft Enterprise Request Objects

Microsoft Enterprise caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Microsoft Enterprise</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is <b>true</b> .
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA

Name	Mandatory	Description	Field Type	Constraints
csrParameters	Yes	Parameters that are necessary for generating the CSR.	MS EnterprisecsrParameters	NA
genericFields	No	Custom fields configured for the CA/Customer.	genericFields	NA
vendorSpecificDetails	Yes	Details specific to the MS Enterprise vendor.	MS Enterprise vendorSpecificDetails	NA
name	No	Specifies the name for the caConnector	String	NA

## MS Enterprise csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as 'Strict'.
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
state	No	State or region where the organization is located.This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
EmailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
encryptedChallengePassword	Yes	Base64 encoded challenge password.	String	NA

Name	Mandatory	Description	Field Type	Constraints
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	MS Enterprise enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## MS Enterprise vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
templateName	Yes	Template name to issue the certificate with.	String	NA

## MS Enterprise enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	Should be a valid domain name.
iPAddresses	No	IP addresses to be considered as Subject Alternative Names.	Array of String	Must be valid ip addresses.
directoryNames	No	List of Directory names as Subject Alternative names for the Certificate.	Array of String	Should be a valid directory name.
rfc822Names	No	List of mail addresses as Subject Alternative names for the Certificate.	Array of String	Should be a valid mail address.
registeredIDs	No	List of registered ids as Subject Alternative names for the Certificate.	Array of String	Should be a valid registered id.
uniformResourceIdentifiers	No	List of URI as Subject Alternative names for the Certificate.	Array of String	Should be a valid URI.
otherNames	No	List of other names as Subject Alternative names for the Certificate.	Array of String	Should be a valid other name.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
```

```

"certificateAuthority": "Microsoft Enterprise",
"isAutoRenewal": false,
"autoRegenerateEnabled": false,
"caSettingName": "MS_ENT",
"certificateType": "",
"description": "",
"csrParameters": {
"commonName": "testcert.testdomain.com",
"organization": "AppViewX",
"organizationUnit": "AppViewX",
"locality": "Texas",
"state": "Texas",
"country": "US",
"mailAddress": "test@appviewx.com",
"hashFunction": "SHA256",
"encryptedChallengePassword": "QWRtaW5AMTlz",
"keyType": "RSA",
"bitLength": "2048",
"certificateCategories": [
"Server",
"Client"
],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": [
"testmscert.appviewx.com"
]
}
},
"genericFields": {
"device_name_Microsoft Enterprise": "",
"vs_ip_Microsoft Enterprise": ""
},
"vendorSpecificDetails": {
"templateName": "ServerCertificate"
}
},

```

```
"certificateGroup": {
  "name": "Default"
}
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "resourceId": "5f4e5c2e70040d33314f0e9d",
    "requestId": "150"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Microsoft Standalone

- [Microsoft Standalone Request Objects](#)
- [Sample Request/Response](#)

## Microsoft Standalone Request Objects

Microsoft Standalone caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be <b>Microsoft Standalone</b> .

Name	Mandatory	Description	Field Type	Constraints
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on <code>renewBefore</code> days.	Boolean	Should be disabled if <code>autoRegenerateEnabled</code> is <b>true</b> .
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if <code>isAutoRenewal</code> is <b>true</b> .
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on <code>renewBefore</code> days.	Boolean	Should be disabled if <code>isAutoRenewal</code> is <b>true</b> .
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if <code>autoRegenerateEnabled</code> is <b>true</b> .
caSettingName	Yes	Name of the <code>CASetting</code> that has been created in AppViewX for the chosen Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">MS EnterprisecsrParameters</a>	NA
genericFields	No	Custom fields configured for the CA/Customer.	<a href="#">genericFields</a>	NA
name	No	Specifies the name for the CA connector	String	NA

[MS Enterprise csrParameters](#)

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located.This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
EmailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
encryptedChallengePassword	Yes	Base64 encoded challenge password.	String	NA
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the	String	Default value - the first value

Name	Mandatory	Description	Field Type	Constraints
		bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.		will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	MS Standalone enhancedSANTypes	Value provided must be compliant with the Certificate Policy, if the policy is configured as <b>Strict</b> .

## MS Standalone enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	Should be a valid domain name.
iPAddresses	No	IP addresses to be considered as Subject Alternative Names.	Array of String	Must be valid ip addresses.
directoryNames	No	List of Directory names as Subject Alternative names for the Certificate.	Array of String	Should be a valid directory name.
rfc822Names	No	List of mail addresses as Subject Alternative names for the Certificate.	Array of String	Should be a valid mail address.
registeredIDs	No	List of registered ids as Subject Alternative names for the Certificate.	Array of String	Should be a valid registered id.

Name	Mandatory	Description	Field Type	Constraints
uniformResourceIdentifiers	No	List of URI as Subject Alternative names for the Certificate.	Array of String	Should be a valid URI.
otherNames	No	List of other names as Subject Alternative names for the Certificate.	Array of String	Should be a valid other name.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "Microsoft Standalone",
    "isAutoRenewal": false,
    "autoRegenerateEnabled": false,
    "caSettingName": "MS_STD",
    "csrParameters": {
      "commonName": "testcert.testdomain.com",
      "organization": "AppViewX",
      "organizationUnit": "AppViewX",
      "locality": "Texas",
      "state": "Texas",
      "country": "US",
      "mailAddress": "test@appviewx.com",
      "hashFunction": "SHA256",
      "encryptedChallengePassword": "QWRtaW5AMTlz",
      "keyType": "RSA",
      "bitLength": "2048",
      "certificateCategories": [
```

```

"Server",
"Client"
],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": [
"testmscert.appviewx.com"
]
}
},
"genericFields": {
"device_name_Microsoft Standalone": "",
"vs_ip_Microsoft Standalone": ""
}
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response": {
"resourceId": "5f4e5c2e70040d33314f0e9d",
"requestId": "151"
},
"message": "Certificate submission triggered successfully.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## OpenTrust Request Objects

### OpenTrust caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate	String	Value should be <b>OpenTrust</b> .
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days	Boolean	Should be disabled if autoRegenerateEnabled is true.
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request	Integer	Value must be provided if isAutoRenewal is true
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days	Boolean	Should be disabled if isAutoRenewal is true.
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request	Integer	Value must be provided if autoRegenerateEnabled is true
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority	String	NA
description	No	Note about the certificate	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR	OpenTrust csrParameters	NA

Name	Mandatory	Description	Field Type	Constraints
vendorSpecificDetails	Yes	Data specific to the Sectigo vendor	OpenTrust vendor Specific Details	NA
name	No	Specifies the name for the caConnector	String	NA
certificateProfileName	Yes	Name of the certificate profile configured in the CA portal	String	Certificate profiles will be fetched while saving the CA setting. One of those profiles must be chosen for enrollment.

## OpenTrust csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as 'Strict'.
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
streetAddress	No	Street address where the organization is located.	String	NA
locality	No	City where the organization is located.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
		This shouldn't be abbreviated.		
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
postalCode	No	Postal code for the organization address.	String	NA
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client,

Name	Mandatory	Description	Field Type	Constraints
				Code Signing, Email.
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as <b>'Strict'</b> .
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located. This shouldn't be abbreviated.	String	Default value - Value configured in the policy.

Name	Mandatory	Description	Field Type	Constraints
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email
ellipticCurve	No	If the keyType chosen is <b>EC</b> , then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the	String	Default value - the first value will be chosen from the policy.

Name	Mandatory	Description	Field Type	Constraints
		possible values configured in the Certificate Policy.		
Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as 'Strict'
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array of String	Possible values - Server, Client, Code Signing, Email

Name	Mandatory	Description	Field Type	Constraints
ellipticCurve	No	If the keyType chosen is EC, then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy
enhancedSANTypes	No	Subject alternative names for the certificate.	OpenTrust enhancedSANTypes	NA

## OpenTrust vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
formFields	Yes	Certificate parameters that are configured in the chosen certificate profile.  Parameters can be identified in the CA portal or the certificate enrollment page in the AppViewX UI.	OpenTrust formFields	CA settings must have been saved after choosing the required certificate profile.

## OpenTrust formFields

There could be any number of parameters configured in the Certificate Management Profile in the OpenTrust CA portal. One sample parameter has been specified in the below table.

Name	Mandatory	Description	Field Type	Constraints
commonName1	Mandatory constraint depends on the profile configuration in the CA portal.	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	NA

## OpenTrust enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA
iPAddresses	No	IP addresses to be considered as Subject Alternative Names.	Array of String	Must be valid ip addresses.
Name	Mandatory	Description	Field Type	Constraints
dNSNames	Depends on the profile configuration in the CA portal.	List of Subject Alternative names for the Certificate	Array of String	AlternativeName (DNS) must have been enabled in the CA Portal
iPAddresses	Depends on the profile configuration in the CA portal.	IP addresses to be considered as Subject Alternative Names	Array of String	AlternativeName (IP) must have been enabled in the CA Portal
uniformResourceIdentifiers	Depends on the profile configuration in the CA portal.	URIs to be considered as Subject Alternative Names	Array of String	AlternativeName (URI) must have been enabled in the CA Portal
rfc822Names	Depends on the profile configuration in the CA portal.	Email addresses to be considered as Subject Alternative Names	Array of String	Email as subject alternative name must have been enabled in the CA Portal

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
    "certificateAuthority": "OpenTrust",
    "isAutoRenewal": "true",
    "renewBefore": "30",
    "autoRegenerateEnabled": false,
    "caSettingName": "OpenTrust",
    "description": "Test certificate",
    "csrParameters": {
      "commonName": "testcert.testdomain.com",
      "hashFunction": "SHA256",
      "keyType": "RSA",
      "bitLength": "2048",
      "certificateCategories": ["Server", "Client"],
      "ellipticCurve": "",
      "enhancedSANTypes": {
        "dNSNames": []
      }
    },
    "vendorSpecificDetails": {
      "formFields": {
        "commonName1": "testcertot.appviewx.net",
        "subjectAltNameDNS1": "testcertot1.appviewx.net"
      }
    },
    "certificateProfileName": "server"
  },
  "certificateGroup": {
    "name": "Default"
  }
}
```

## Sample Response:

```
{
  "response": {
    "resourceId": "606357ec14974147f02c53ec",
    "requestId": "119"
  },
  "message": "Certificate submission triggered successfully.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Entrust MPKI

- [Entrust MPKI Request Objects](#)
- [Sample Request/Response](#)

## Entrust MPKI Request Objects

### Entrust MPKI caConnectorInfo

Name	Mandatory	Description	Field Type	Constraints
certificateAuthority	Yes	Name of the certificateAuthority that will issue the certificate.	String	Value should be Entrust MPKI.
isAutoRenewal	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if autoRegenerateEnabled is true.
renewBefore	No	Specifies the no of days prior to expiry for triggering the renewal request.	Integer	Value must be provided if isAutoRenewal is true.

Name	Mandatory	Description	Field Type	Constraints
autoRegenerateEnabled	No	If enabled, renewal will be triggered before expiry based on renewBefore days.	Boolean	Should be disabled if isAutoRenewal is true.
regenerateBeforeInDays	No	Specifies the no of days prior to expiry for triggering the regenerate request.	Integer	Value must be provided if autoRegenerateEnabled is true.
caSettingName	Yes	Name of the CASetting that has been created in AppViewX for the chosen Certificate Authority.	String	NA
description	No	Note about the certificate.	String	NA
csrParameters	Yes	Parameters that are necessary for generating the CSR.	<a href="#">Entrust MPKI</a> <a href="#">csrParameters</a>	NA
vendorSpecificDetails	Yes	Details specific to the Entrust vendor.	<a href="#">Entrust MPKI</a> <a href="#">vendorSpecificDetails</a>	NA
name	No	Specifies the name for the caConnector	String	NA

## Entrust MPKI csrParameters

Name	Mandatory	Description	Field Type	Constraints
commonName	Yes	Fully qualified domain name (FQDN) of the server for which certificate is requested.	String	Must be compliant with the common name specified in the policy, if the policy is set as 'Strict'.

Name	Mandatory	Description	Field Type	Constraints
organization	No	Legal name of the organization.	String	Default value - Value configured in the policy.
organizationUnit	No	Division or department of the organization handling the certificate.	String	Default value - Value configured in the policy.
locality	No	City where the organization is located.  This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
state	No	State or region where the organization is located.This shouldn't be abbreviated.	String	Default value - Value configured in the policy.
country	No	The two-letter code for the country where your organization is located.	String	Default value - Value configured in the policy.
mailAddress	No	Email address of the organization.	String	Default value - Value configured in the policy.
hashFunction	No	Hash function to be used in the Certificate. For example, SHA160.  Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.

Name	Mandatory	Description	Field Type	Constraints
keyType	No	Algorithm to be used for Key generation. For example, RSA, DSA, EC. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
bitLength	No	Bit length for the key is dependent on the key type chosen. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
certificateCategories	Yes	Purpose for which the generated certificate will be used.	Array	Possible values - Server, Client, Code Signing, Email
ellipticCurve	No	If the keyType chosen is EC, then the ellipticCurve must be specified depending on the bitlength selected. Should be chosen from the possible values configured in the Certificate Policy.	String	Default value - the first value will be chosen from the policy.
enhancedSANTypes	No	Subject alternative names for the certificate.	<a href="#">Entrust MPKI enhancedSANTypes</a>	Value provided must be compliant with the Certificate Policy, if the policy is

Name	Mandatory	Description	Field Type	Constraints
				configured as Strict.

## Entrust MPKI vendorSpecificDetails

Name	Mandatory	Description	Field Type	Constraints
caName	Yes	Name of the CA.	String	Should be a valid CA name configured in CA Settings.
certProfile	Yes	Certificate Profile name associated with the given caName.	String	Should be a valid Certificate Profile name configured in CA Settings.

## Entrust MPKI enhancedSANTypes

Name	Mandatory	Description	Field Type	Constraints
dNSNames	No	List of Subject Alternative names for the Certificate.	Array of String	NA
iPAddresses	No	IP addresses to be considered as Subject Alternative Names.	Array of String	Must be valid ip addresses.
uniformResourceIdentifiers	No	URIs to be considered as Subject Alternative Names.	Array of String	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "csrGenerationSource": "appviewx",
  "caConnectorInfo": {
```

```

"certificateAuthority": "Entrust MPKI",
"isAutoRenewal": "true",
"renewBefore": "30",
"autoRegenerateEnabled": false,
"caSettingName": "Entrust MPKI",
"description": "",
"csrParameters": {
"commonName": "testcert.testdomain.com",
"mailAddress": "test@test.com",
"hashFunction": "SHA256",
"keyType": "RSA",
"bitLength": "2048",
"certificateCategories": ["Server", "Client"],
"ellipticCurve": "",
"enhancedSANTypes": {
"dNSNames": [],
"IPAddresses": [],
"uniformResourceIdentifiers": []
}
},
"vendorSpecificDetails": {
"caName": "Test CA",
"certProfile": "Test CA Profile"
}
},
"certificateGroup": {
"name": "Default"
}
}

```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```

{
"response": {
"resourceId": "5f4fcb1770040d33314f11f0",

```

```
"requestId": "184"  
},  
"message": "Certificate submission triggered successfully.",  
"appStatusCode": null,  
"tags": {},  
"headers": null  
}
```

# Chapter 15: Common Payload Objects

genericFields

Name	Mandatory	Description	Field Type	Constraints
device_name_<certificateAuthority>	No	Device name of the Server for which certificate is requested.	String	In the field name, <certificateAuthority> has to be replaced by the value provided for certificateAuthority in caConnectorInfo.
vs_ip_<certificateAuthority>	No	IP address of the server for which certificate is requested.	String	In the field name, <certificateAuthority> has to be replaced by the value provided for certificateAuthority in caConnectorInfo.

## Chapter 16: Overview

The API will submit a request to create a certificate in Sync Mode. In sync mode, certificate will be generated immediately and sent in the response. Request payload varies depending on the Certificate Authority (CA) with which enrollment or creation is requested. There are two parameters that are different than async APIs. They are "isSync" and "ttl". If "isSync", that is, Sync Mode is true, it overrides the policy for certificate request approvals to create a new Certificate. You can use these two parameters for all the CA's mentioned in the async APIs.

# Chapter 17: Request Structure

API : /certificate/create

Type: POST

URL: [https://<APPVIEWX\\_GATEWAY\\_IP>:<APPVIEWX\\_GATEWAY\\_PORT>/avxapi/certificate/create?gwkey=f000ca01&gwsouce=external&isSync=true&tfl=300](https://<APPVIEWX_GATEWAY_IP>:<APPVIEWX_GATEWAY_PORT>/avxapi/certificate/create?gwkey=f000ca01&gwsouce=external&isSync=true&tfl=300)

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b>
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered(E.g. external).	String	NA
isSync	Query	If you want to use sync API, set the value as <b>true</b> . If you want to use the async API, set the value as <b>false</b> .  <b>Note: Default value is false .</b>	Boolean	Must be a valid boolean value : <b>true/false</b> .

Name	Param Type	Description	Field Type	Constraints
ttl	Query	Time to live for a response in seconds.  Default value is 300 seconds.	Integer	Must be a positive integer.
Payload	Body	Contains all the params to be sent in the request body for the post request	<a href="#">SyncRequestStructure</a>	NA

[SyncRequestStructure](#)

Name	Mandatory	Description	Field Type	Constraints
<a href="#">csrGenerationSource</a>	No	Specifies where the CSR is to be generated.	String	<b>Possible values:</b> appviewx, HSM, ENDPOINT, uploadCSR  <b>Default value:</b> appviewx
<a href="#">caConnectorInfo</a>	Yes	Details related to Certificate Authority and CSR Parameters	Any of the following - <ul style="list-style-type: none"> <li><a href="#">Sectigo caConnectorInfo</a></li> <li><a href="#">DigiCert caConnectorInfo</a></li> <li><a href="#">InCommon caConnectorInfo</a></li> <li><a href="#">AppViewX caConnectorInfo</a></li> <li><a href="#">EJBCA caConnectorInfo</a></li> <li><a href="#">GlobalSign caConnectorInfo</a></li> <li><a href="#">Trustwave caConnectorInfo</a></li> </ul>	

Name	Mandatory	Description	Field Type	Constraints
			<ul style="list-style-type: none"> <li>Entrust caConnectorInfo</li> <li>Symantec caConnectorInfo</li> <li>LetsEncrypt caConnectorInfo</li> <li>GoDaddy caConnectorInfo</li> <li>Google caConnectorInfo</li> <li>Amazon caConnectorInfo</li> <li>Microsoft Enterprise caConnectorInfo</li> <li>Microsoft Standalone caConnectorInfo</li> <li>Open Trust caConnectorInfo</li> </ul>	
certificateGroup	No	Specifies the group under which the created certificate needs to be tagged.	certificateGroup	<b>Default certificateGroup:</b> Default
deviceDetails	No	Details on the device/ endpoint in which CSR will be generated	deviceDetails	Required if the <b>csrGenerationSource</b> is <b>ENDPOINT</b>
certificateHSMDetails	No	Details on the Hardware Security Module (HSM) device	certificateHSMDetails	Required if the <b>csrGenerationSource</b> is <b>HSM</b>
uploadCsrDetails	No	Details on the CSR	uploadCsrDetails	Required if the <b>csrGenerationSource</b> is <b>uploadCSR</b>

Name	Mandatory	Description	Field Type	Constraints
certificateFormat	No	Certificate download format details.	certificateFormat	

## certificateGroup

Name	Mandatory	Description	Field Type	Constraints
name	No	Specifies the group under which the created certificate needs to be tagged.	String	Group must already be present in AppViewX

## deviceDetails

Name	Mandatory	Description	Field Type	Constraints
category	Yes	Specifies the device category	String	<b>Possible values:</b> ADC, Server, Firewall
vendor	Yes	Vendor for the chosen device. For example, <b>Apache</b> is a vendor for <b>Server</b> category	String	NA
deviceName	Yes	Name of the device as per AppViewX Device Inventory	String	NA
csrFileName	Yes	Name of the CSR file that will be generated in the device	String	NA
keyFileName	Yes	Name of the Key file that will be generated in the device	String	NA
attributes	No	Additional attributes related to device	attributes	NA

## attributes

Name	Mandatory	Description	Field Type	Constraints
csrLocation	No	Location in the device where CSR will be created	String	Required if <b>deviceDetails.category - Server</b> and <b>deviceDetails.vendor - Tomcat</b>

Name	Mandatory	Description	Field Type	Constraints
tenant	No	Name of the partition in the AVI device	String	NA
partition	No	Name of the partition in the device	String	Required if <b>deviceDetails.category</b> - <b>Firewall</b> and <b>deviceDetails.vendor</b> - <b>Fortinet</b>

## certificateHSMDetails

Name	Mandatory	Description	Field Type	Constraints
type	Yes	Type of the HSM device	String	<b>Possible values:</b> ADC, hsm
keyReference	Yes	Reference name for the key that will be mapped by the HSM device.	String	NA
hsmSettings	Yes	Configuration details for the HSM device	<a href="#">hsmSettings</a>	NA
vendor	Yes	Vendor for the chosen device. For example, F5.	String	NA
deviceName	Yes	Name of the device as per AppViewX Device Inventory	String	NA

## hsmSettings

Name	Mandatory	Description	Field Type	Constraints
vendorType	Yes	Category of the vendor	String	<b>Possible values:</b> Safenet, Thales, Fortanix
vendorSpecificSettings	Yes	Settings related to HSM vendor	<a href="#">HSM</a> <a href="#">vendorSpecificSettings</a>	NA

## HSM vendorSpecificSettings

Name	Mandatory	Description	Field Type	Constraints
moduleId	No	Module Id	String	Applicable if <b>hsmSettings.vendorType</b> is <b>Thales</b>

## uploadCsrDetails

Name	Mandatory	Description	Field Type	Constraints
category	Yes	Certificate category	String	<b>Possible values</b> - Server, Client, Code Signing
csrContent	Yes	The CSR content for certificate enrollment request	String	NA

## certificateFormat

Name	Mandatory	Description	Field Type	Constraints
format	Yes	Certificate download format.	String	<b>Refer to the</b> <a href="#">Possible values for Certificate Download Format</a>
password	Yes	The field is mandatory for some parameters.	String	NA

## Possible values for Certificate Download Format

Certificate Extension	Value to be provided in payload	Password Required
.crt	CRT	No
.cert	CERT	No
.cer	CER	No
.pem	PEM	No
.der	DER	No
.cer	DERCER	No
.p7b	P7B	No
.p7c	P7C	No
.pk8	PK8	No
.p12	P12	Yes
.pfx	PFX	Yes


Certificate Extension	Value to be provided in payload	Password Required
.jks	JKS	Yes

# Chapter 18: SyncResponseStructure

Response returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response attributes for the enrollment request.	SyncResponse
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

SyncResponse

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.  <b>Note:</b> resourceId in the above request can be found using the search API using commonName, serial No. or other search parameters.	String
requestId	Open work order request ID.	String
certificateContent	Content of certificate in Base64 encoded format.	String
encodedFormat	Certificate content encoded format.	String
certificateName	Certificate name.	String

# Chapter 19: Sample Request/Response

- Sample Request:
- Sample Response:

## Sample Request:

```
{
  "caConnectorInfo": {
    "caSettingName": "",
    "validityInDays": 365,
    "isAutoRenewal": "true",
    "renewBefore": "30",
    "autoRegenerateEnabled": false,
    "certificateType": "",
    "csrParameters": {
      "commonName": "",
      "organization": "",
      "organizationUnit": "",
      "locality": "",
      "state": "",
      "country": "",
      "mailAddress": "",
      "encryptedChallengePassword": "",
      "hashFunction": "",
      "keyType": "",
      "bitLength": "",
      "certificateCategories": [
        "Server",
        "Client"
      ],
      "ellipticCurve": "",
      "enhancedSANTypes": {
        "dNSNames": [],
        "iPAddresses": [],
        "uniformResourceIdentifiers": []
      }
    }
  }
}
```

```

}
}
"genericFields": {
"device_name_Entrust": "",
"vs_ip_Entrust": "",
"type": "",
"owner_email": "",
"environment": "",
"partition": "",
"devices": "",
"sync_group": "",
"requestor": "",
"project": ""
},
"vendorSpecificDetails": {
"avx_requester_fullname": "",
"avx_requester_emailid": "",
"avx_requester_phoneno": "",
"additionalEmails": ""
},
"customAttributes": {
"attribute1": ""
}
},
"certificateGroup": {
"name": "Default"
},
"certificateFormat": {
"format": "PEM",
"password": ""
}
}
}

```

## Sample Response:

```

{
"response": {

```

```
"resourceId": "<Certificate Object ID>",
"requestId": "<Certificate WF Request ID>",
"certificateContent": "<Encoded Certificate Content>",
"encodedFormat": "base64",
"certificateName": "<Name of the Certificate>"
},
"message": "Certificate create and download action performed successfully",
"appStatusCode": "<Error Code>",
"tags": {},
"headers": null
}
```

## Chapter 20: Overview

The API will submit a request to renew an existing certificate in async mode. Once the Renew a certificate API has been triggered successfully, the response will contain the resourceId, which can be used to trigger the [search](#) API to fetch the created certificate. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 21: Before you begin

Before attempting to renew certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 22: Request Structure

URL: /certificate/action

Type: PUT

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b>
gwkey	Query	Tenant Key	String	NA
gwsource	Query	Source from which the request is triggered	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
resourceId	No	Unique Id of the certificate .  Note:resourceId in the above request can be found using the search API using commonName, serial No. or other search parameters.	String	Required if the commonName and serialNumber are not specified.
commonName	No	Common name of the certificate	String	Required if resourceId is not specified.

Name	Mandatory	Description	Field Type	Constraints
serialNumber	No	Serial number of the certificate	String	Required if resourceId is not specified.
action	Yes	Action name for the renewal	String	Value should be Renew
challengePassword	No	Base64 encoded value of challenge password	String	Applicable and mandatory for Symantec Certificate Authority

## Chapter 23: Response Structure

202 Accepted returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the renewal request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	WorkOrder request Id.	String
message	Success message - Renew action triggered successfully.	NA

## Chapter 24: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	-	Renew action has been triggered successfully.	-
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName/serialNumber/resourceId</code> is correct.
400 Bad Request	INVALID_REQUEST	Please give valid common name and serial number or <code>resourceId</code> .	Please give valid common name and serial number or <code>resourceId</code> .
400 Bad Request	INVALID_REQUEST	Please provide a valid action.	Please provide a valid action.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - action.	Ensure that the <b>action</b> field is available in the request payload.
417 Expectation Failed	OPEN_WORK_ORDERS_FOUND	Since requested certificate's work order is in progress, cannot initiate another action.	Trigger the request once the open work order for the certificate is completed.

HTTP Status code	appStatusCode	Message	Possible remediation
406 Not Acceptable	CERT-VWF-0006	Life cycle action is unsupported by CA or another work order is in progress or certificate belongs to read group or is in Monitored status.	Ensure the following: <ul style="list-style-type: none"><li>• The CA supports the renew action.</li><li>• There is no workOrder in progress for the specified certificate.</li><li>• Certificate should not belong to read only group.</li><li>• Certificate is not in monitored status.</li></ul>

## Chapter 25: Sample Request/Response

- Sample Request:
- Sample Response:

### Sample Request:

```
{  
  "commonName": "testcert8g.appviewx.plus",  
  "serialNumber": "0D:A9:2D:8C:90:BB:90:B0:CE:7D:6A:76:BF:70:75:81",  
  "action": "Renew"  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

```
{  
  "response": {  
    "resourceId": "5f4faf3e70040d33314f1142",  
    "message": "Renew action triggered successfully.",  
    "requestId": "209"  
  },  
  "message": "Renew action has been triggered successfully",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

## Chapter 26: Overview

The API will submit a request to renew an existing certificate in Sync Mode. There are two parameters that are different than async apis. They are "isSync" and "ttl". If "isSync", that is, Sync Mode is true, it overrides the policy for certificate request approvals to Renew an existing Certificate. You can use these two parameters for all the CA's mentioned in the async APIs.


# Chapter 27: Request Structure

API : /certificate/renew


Method: PUT

URL: *https://<APPVIEWX\_GATEWAY\_IP>:<APPVIEWX\_GATEWAY\_PORT>/avxapi/certificate/renew?gwkey=f000ca01&gwsource=external&isSync=true&ttl=300*

Name	Type	Mandatory	Field Type	Description	Constraints
userName	Header	Yes	String	Username that is configured in AppViewX.	NA
password	Header	Yes	String	Password of that user.	NA
content-type	Header	Yes	String	Payload content-type with application/json value.	The value must be application/json.
gwkey	Queryparam	Yes	String	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	NA
gwsource	Queryparam	Yes	String	Source, from which the request is triggered. For example, external.	NA
isSync	Queryparam	No	boolean	The possible values are true and false. If you want to use sync API, set the value as <b>true</b> . If you want to use the async API, set the value as <b>false</b> .	Must be a valid boolean value: true or false.
ttl	Queryparam	No	int	Time to live for a response in seconds. Default value is 300 seconds.	Must be a positive integer.
body	Body	Yes	json	Refer to the sample request body.	NA

 **Note:** Default value is **false**.

Request Details

Name	Mandatory	Description	Field Type	Constraints
resourceId	Yes	Mongo Id of the certificate in the AppViewX database.  <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  <b>Note:</b> It refers to the resourceId field in the create certificate response. </div>	String	Either resourceId or serialNumber and commonName is mandatory.
commonName	Yes	Common name of the certificate.	String	
serialNumber	Yes	Serial number of the certificate.	String	
format	Yes	Certificate downloadable format.	String	Refer to the <a href="#">Possible values for Certificate Download Format</a> table.
password	Yes	The field is mandatory for some parameters.	String	

## Possible values for Download Format

Certificate Extension	Value to be provided in payload	Password Required
.crt	CRT	No
.cert	CERT	No
.cer	CER	No
.pem	PEM	No
.der	DER	No
.cer	DERCER	No
.p7b	P7B	No

Certificate Extension	Value to be provided in payload	Password Required
.p7c	P7C	No
.pk8	PK8	No
.pk12	PK12	Yes
.pfx	PFX	Yes
.jks	JKS	Yes

# Chapter 28: Sync Response Structure

Response returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the enrollment request.	SyncRenewResponse
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	

## SyncRenewResponse

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Open work order request ID.	String
certificateContent	Content of certificate in Base64 encoded format.	String
encodedFormat	Certificate content encoded format.	String
certificateName	Name of the Certificate.	String

## Chapter 29: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{  
  "resourceId": "<Mongo ID of the certificate in AppviewX Database>,"  
  "commonName": "<Common Name of the certificate>,"  
  "serialNumber": "<Serial Number of the certificate>,"  
  "certificateFormat": {  
    "format": "<Cert format - check possible values table>,"  
    "password": "<password>"  
  }  
}
```

### Sample Response:

```
{  
  "response": {  
    "resourceId": "<Certificate ID>,"  
    "requestId": "<Certificate WF Request ID>,"  
    "certificateContent": "<Encoded Certificate file>,"  
    "certificateName": "<Name of the certificate>,"  
    "encodedFormat": "base64"  
  },  
  "message": "Certificate Renew action executed successfully",  
  "appStatusCode": "<Error Code>,"  
  "tags": {},  
  "headers": null  
}
```

## Chapter 30: Overview

The API will submit a request to revoke an existing certificate in async mode. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 31: Before you begin

Before attempting to revoke a certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 32: Request Structure

URL: /certificate/action

Type: PUT

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered(E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
resourceId	No	Unique Id of the certificate.	String	Required if the commonName and serialNumber are not specified.
commonName	No	Common name of the certificate.	String	Required if resourceId is not specified.

Name	Mandatory	Description	Field Type	Constraints
serialNumber	No	Serial number of the certificate.	String	Required if resourceId is not specified.
action	Yes	Action name for the revocation.	String	Value should be <b>Revoke</b> .
reason	Yes	Reason for revocation.	String	NA

## Chapter 33: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the renewal request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Workorder request Id.	String
message	Success message - Revoke action triggered successfully.	NA

## Chapter 34: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	-	Revoke action has been triggered successfully.	-
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName/serialNumber/resourceId</code> is correct.
400 Bad Request	INVALID_REQUEST	Please give valid common name and serial number or resourceId.	Please give valid common name and serial number or resourceId
400 Bad Request	INVALID_REQUEST	Please provide a valid action.	Please provide a valid action
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - action.	Ensure that the action field is available in the request payload.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - reason.	Ensure that the reason field is available in the request payload.
417 Expectation Failed	OPEN_WORK_ORDERS_FOUND	Since requested certificate's work order is in progress, cannot initiate another action.	Trigger the request once the open worker for the

HTTP Status code	appStatusCode	Message	Possible remediation
			certificate is completed.
406 Not Acceptable	CERT-VWF-0006	Life cycle action is unsupported by CA or another work order is in progress or certificate belongs to read group or is in Monitored status.	Ensure the following - <ul style="list-style-type: none"> <li>• The CA supports the revoke action.</li> <li>• There is no workOrder in progress for the specified certificate.</li> <li>• Certificate should not belong to read only group</li> <li>• Certificate is not in monitored status</li> </ul>

# Chapter 35: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "resourceId": "5f4faf3e70040d33314f1142",
  "commonName": "testcert8g.appviewx.plus",
  "serialNumber": "0D:A9:2D:8C:90:BB:90:B0:CE:7D:6A:76:BF:70:75:81",
  "action": "Revoke",
  "reason": "Superseded"
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "resourceId": "5f4faf3e70040d33314f1142",
    "message": "Revoke action triggered successfully.",
    "requestId": "216"
  },
  "message": "Revoke action has been triggered successfully",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Chapter 36: Overview

The API will submit a request to revoke an existing certificate in Sync Mode. There are two parameters that are different than async APIs. They are "isSync" and "ttl". If "isSync", that is, Sync Mode is true, it overrides the policy for certificate request approvals to Revoke the Certificate. You can use these two parameters for all the CA's mentioned in the async APIs.


# Chapter 37: Request Structure

API: /certificate/revoke

Method: PUT

URL: `https://<APPVIEWX_GATEWAY_IP>:<APPVIEWX_GATEWAY_PORT>/avxapi/certificate/revoke?gwkey=f000ca01&gwsource=external&isSync=true&tfl=300`

Name	Type	Mandatory	Field Type	Description	Constraints
userName	Header	Yes	String	Username that is configured in AppViewX.	
password	Header	Yes	String	Password of that user.	
content-type	Header	Yes	String	Payload content-type with application/json value.	The value must be application/json.
gwkey	Queryparam	Yes	String	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	
gwsource	Queryparam	Yes	String	Source, from which the request is triggered. For example, external.	
isSync	Queryparam	No	boolean	The possible values are true and false. If you want to use sync API, set the value as <b>true</b> . If you want to use the async API, set the value as <b>false</b> .	Must be a valid boolean value: true or false.

 **Note:** Default value is **false**.

Name	Type	Mandatory	Field Type	Description	Constraints
ttl	Queryparam	No	Int	Time to live for a response in seconds. Default value is 300 seconds.	Must be a positive integer.
Body	Body	Yes	json	Refer to the sample request body.	

## SyncRevokeRequest

Name	Mandatory	Description	Field Type	Constraints
resourceId	Yes	Mongo Id of the certificate in the AppViewX database.  <b>Note:</b> It refers to the 'resourceId' field in the create certificate response. It can be found using the search API using commonName, serial No. or other search parameters.	String	Either resourceId or serialNumber and commonName is mandatory.
commonName	Yes	Common name of the certificate.	String	
serialNumber	Yes	Serial number of the certificate.	String	
reason	Yes	Reason for revoke.	String	
comments	Yes	Comments for Revocation	String	Comments field is mandatory for some revoke reasons alone. Refer <b>Valid Reason for CAs</b> table.

## Valid Reason for CAs

Certificate Authority	Reasons	Comments Required
Entrust	Superseded	No
	Cessation of operation	No
	Affiliation Changed	No
	Key compromised	Yes

## Chapter 38: Sync Revoke Response Structure

Response returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response attributes for the enrollment request.	<a href="#">RevokeResponseStructure</a>
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

RevokeResponseStructure

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Open work order request ID.	String
certStatus	Status of the certificate action request.	String

## Chapter 39: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{  
  "resourceId": "5fa27b7ffee3a70235a5816d",  
  "reason": "Superseded"  
}
```

### Sample Response:

```
{  
  "response": {  
    "resourceId": "<Certificate ID>",  
    "requestId": "<Certificate WF Request ID>",  
    "certStatus": "Revoked"  
  },  
  "message": "Certificate Revoke action executed successfully",  
  "appStatusCode": "<Error Code>",  
  "tags": {},  
  "headers": null  
}
```

## Chapter 40: Overview

The API will submit a request to reissue an existing certificate in async mode. Once the Reissue a certificate API has been triggered successfully, the response will contain the resourceId, which can be used to trigger the [search](#) API to fetch the created certificate. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 41: Before you begin

Before attempting to reissue a certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 42: Request Structure

URL: /certificate/action

Type: PUT

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
resourceId	No	Unique Id of the certificate.	String	Required if the commonName and serialNumber are not specified.
commonName	No	Common name of the certificate.	String	Required if resourceId is not specified.

Name	Mandatory	Description	Field Type	Constraints
serialNumber	No	Serial number of the certificate.	String	Required if resourceId is not specified.
action	Yes	Action name for the reissue.	String	Value should be <b>Reissue</b> .
reason	Yes	Reason for reissue request.	String	NA

# Chapter 43: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the renewal request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Workorder request Id.	String
message	Success message - Reissue action triggered successfully.	NA

## Chapter 44: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Reissue action has been triggered successfully.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName/serialNumber/resourceId</code> is correct.
400 Bad Request	INVALID_REQUEST	Please give valid common name and serial number or <code>resourceId</code> .	Please give valid common name and serial number or <code>resourceId</code> .
400 Bad Request	INVALID_REQUEST	Please provide a valid action.	Please provide a valid action.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - action.	Ensure that the action field is available in the request payload.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - reason.	Ensure that the reason field is available in the request payload.
417 Expectation Failed	OPEN_WORK_ORDERS_FOUND	Since requested certificate's work order is in progress, cannot initiate another action.	Trigger the request once the open worker for the

HTTP Status code	appStatusCode	Message	Possible remediation
			certificate is completed.
406 Not Acceptable	CERT-VWF-0006	Life cycle action is unsupported by CA or another work order is in progress or certificate belongs to read group or is in Monitored status.	<p>Ensure the following -</p> <ul style="list-style-type: none"> <li>• The CA supports the reissue action.</li> <li>• There is no workOrder in progress for the specified certificate.</li> <li>• Certificate should not belong to read only group.</li> <li>• Certificate is not in monitored status.</li> </ul>

# Chapter 45: Sample Request/Response

- Sample Request:
- Sample Response:

## Sample Request:

```
{  
  "commonName": "testcert8g.appviewx.plus",  
  "serialNumber": "08:BD:54:35:FD:81:AC:A4:45:2B:FC:9F:77:56:FE:2C",  
  "action": "Reissue",  
  "reason": "Superseded"  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{  
  "response": {  
    "resourceId": "5f51f7312fa95059a12b0aee",  
    "message": "Reissue action triggered successfully.",  
    "requestId": "238"  
  },  
  "message": "Reissue action has been triggered successfully",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

## Chapter 46: Overview

The API will submit a request to reissue an existing certificate in Sync Mode. There are two parameters that are different than async APIs. They are "isSync" and "ttl". If "isSync", that is, Sync Mode is true, it overrides the policy for certificate request approvals to Reissue the Certificate. You can use these two parameters for all the CA's mentioned in the async APIs.


# Chapter 47: Request Structure

**API:** /certificate/reissue


**Method:** PUT

**URL:** https://<APPVIEWX\_GATEWAY\_IP>:<APPVIEWX\_GATEWAY\_PORT>/avxapi/certificate/reissue?gwkey=f000ca01&gwsource=external&isSync=true&ttl=300

Name	Type	Mandatory	Field Type	Description	Constraints
userName	Header	Yes	String	Username that is configured in AppViewX.	NA
password	Header	Yes	String	Password of that user.	NA
content-type	Header	Yes	String	Payload content-type with application/json value.	The value must be application/json.
gwkey	Queryparam	Yes	String	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	NA
gwsource	Queryparam	Yes	String	Source, from which the request is triggered. For example, external.	NA
isSync	Queryparam	No	boolean	The possible values are true and false. If you want to use sync API, set the value as <b>true</b> . If you want to use the async API, set the value as <b>false</b> .	Must be a valid boolean value: true or false.
ttl	Queryparam	No	Int	Time to live for a response in seconds. Default value is 300 seconds.	Must be a positive integer.
body	Body	Yes	json	Refer to the sample request body.	

 **Note:** Default value is **false**.

SyncReIssueRequest

Name	Mandatory	Description	Field Type	Constraints
resourceId	Yes	Mongo Id of the certificate in the AppViewX database.  <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  <b>Note:</b> It refers to the 'resourceId' field in the create certificate response. It can be found using the search API using commonName, serial No. or other search parameters. </div>	String	Either resourceId or serialNumber and commonName is mandatory.
commonName	Yes	Common name of the certificate.	String	
serialNumber	Yes	Serial number of the certificate.	String	
reason	Yes	Reason for reissue.	String	
uploadCsrDetails	No	Details of the CSR	<a href="#">uploadCsrDetails</a>	
certificateFormat	Yes	Certificate download format details.	<a href="#">certificateFormat</a>	Refer to the certificateFormat table.

## uploadCsrDetails

Name	Mandatory	Description	Field Type	Constraints
category	Yes	Certificate category	String	<b>Possible values</b> - Server, Client, Code Signing
csrContent	Yes	The CSR content for certificate enrollment request	String	NA

## certificateFormat

Name	Mandatory	Description	Field Type	Constraints
format	Yes	Certificate download format.	String	<b>Refer to the <a href="#">Possible values for Certificate Download Format</a></b>
password	Yes	The field is mandatory for some parameters.	String	NA

## Possible values for Certificate Download Format

Certificate Extension	Value to be provided in payload	Password Required
.crt	CRT	No
.cert	CERT	No
.cer	CER	No
.pem	PEM	No
.der	DER	No
.cer	DERCER	No
.p7b	P7B	No
.p7c	P7C	No
.pk8	PK8	No
.p12	P12	Yes
.pfx	PFX	Yes
.jks	JKS	Yes

# Chapter 48: Sync Response Structure

Response returns string of type application/json with the following body params

Name	Description	Field Type
Response	Contains the response attributes for the enrollment request.	<a href="#">SyncRelssueResponse</a>
Message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
Tags	More info in case of failure response.	NA

SyncRelssueResponse

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Open work order request ID.	String
certificateContent	Content of certificate in Base64 encoded format.	String
encodedFormat	Certificate content encoded format.	String
certificateName	Name of the Certificate.	String

# Chapter 49: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "resourceId": "<Mongo ID of the certificate in AppviewX Database>",
  "commonName": "<Common Name of the certificate>",
  "serialNumber": "<Serial Number of the certificate>",
  "reason": "<Reason for reissue>",
  "uploadCsrDetails": {
    "csrContent": "",
    "additionalSANValues": {
      "dNSNames": [],
      "iPAddresses": [],
      "uniformResourceIdentifiers": []
    }
  },
  "certificateFormat": {
    "format": "<Cert format - check possible values table>",
    "password": "<password>"
  }
}
```

## Sample Response:

```
{
  "response": {
    "resourceId": "<Certificate ID>",
    "requestId": "<Certificate WF Request ID>",
    "certificateContent": "<Encoded Certificate file>",
    "certificateName": "<Name of the certificate>",
    "encodedFormat": "base64"
  },
}
```

```
"message": "Certificate Reissue action executed successfully"  
"appStatusCode": "<Error Code>",  
"tags": {},  
"headers": null  
}
```

## Chapter 50: Overview

The API will submit a request to regenerate an existing certificate in async mode. Once the Regenerate a certificate API has been triggered successfully, the response will contain the resourceId, which can be used to trigger the [search](#) API to fetch the created certificate. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 51: Before you begin

Before attempting to regenerate a certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 52: Request Structure

URL: /certificate/action

Type: PUT

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
resourceId	No	Unique Id of the certificate.	String	Required if the commonName and serialNumber are not specified.

Name	Mandatory	Description	Field Type	Constraints
commonName	No	Common name of the certificate.	String	Required if resourceId is not specified.
serialNumber	No	Serial number of the certificate.	String	Required if resourceId is not specified.
action	Yes	Action name for the regenerate request.	String	Value should be <b>Regenerate</b> .
reason	No	Reason for regenerate request.	String	NA

# Chapter 53: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the renewal request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Workorder request Id.	String
message	Success message - Regenerate action triggered successfully.	NA

## Chapter 54: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Regenerate action has been triggered successfully.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName/serialNumber/resourceId</code> is correct.
400 Bad Request	INVALID_REQUEST	Please give valid common name and serial number or <code>resourceId</code> .	Please give valid common name and serial number or <code>resourceId</code> .
400 Bad Request	INVALID_REQUEST	Please provide a valid action.	Please provide a valid action.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - action.	Ensure that the <code>action</code> field is available in the request payload.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - reason.	Ensure that the <code>reason</code> field is available in the request payload.
417 Expectation Failed	OPEN_WORK_ORDERS_FOUND	Since requested certificate's work order is in progress, cannot initiate another action.	Trigger the request once the open <code>workOrder</code> for

HTTP Status code	appStatusCode	Message	Possible remediation
			the certificate is completed.
406 Not Acceptable	CERT-VWF-0006	Life cycle action is unsupported by CA or another work order is in progress or certificate belongs to read group or is in Monitored status.	<p>Ensure the following -</p> <ul style="list-style-type: none"> <li>• The CA supports the <b>regenerate</b> action.</li> <li>• There is no workOrder in progress for the specified certificate.</li> <li>• Certificate should not belong to read only group.</li> <li>• Certificate is not in monitored status.</li> </ul>

# Chapter 55: Sample Request/Response

- Sample Request:
- Sample Response:

## Sample Request:

```
{  
  "resourceId": "5f5641a7e74e187a7e3d5266",  
  "commonName": "testcert8g.appviewx.plus",  
  "serialNumber": "08:BD:54:35:FD:81:AC:A4:45:2B:FC:9F:77:56:FE:2C",  
  "action": "Regenerate",  
  "reason": "Key compromised"  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{  
  "response": {  
    "resourceId": "5f56482ce74e187a7e3d52a2",  
    "message": "Regenerate action triggered successfully.",  
    "requestId": "237"  
  },  
  "message": "Regenerate action has been triggered successfully",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

## Chapter 56: Overview


The API will submit a request to regenerate a certificate in Sync Mode. There are two parameters that are different than async APIs. They are "isSync" and "ttl". If "isSync", that is, Sync Mode is true, it overrides the policy for certificate request approvals to Regenerate the Certificate. You can use these two parameters for all the CA's mentioned in the async APIs.

# Chapter 57: Request Structure


**API:** /certificate/regenerate

**Method:** PUT

**URL:** https://<APPVIEWX\_GATEWAY\_IP>:<APPVIEWX\_GATEWAY\_PORT>/avxapi/certificate/regenerate?gwkey=f000ca01&gwsouce=external&isSync=true&tfl=300

Name	Type	Mandatory	Field Type	Description	Constraints
userName	Header	Yes	String	Username that is configured in AppViewX.	NA
password	Header	Yes	String	Password of that user.	NA
content-type	Header	Yes	String	Payload content-type with application/json value.	The value must be application/json.
gwkey	Queryparam	Yes	String	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	NA
gwsouce	Queryparam	Yes	String	Source, from which the request is triggered. For example, external.	NA
isSync	Queryparam	No	boolean	The possible values are true and false. If you want to use sync API, set the value as <b>true</b> . If you want to use the async API, set the value as <b>false</b> .  <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 5px; display: inline-block;">  <b>Note:</b> Default value is <b>false</b>.         </div>	Must be a valid boolean value: true or false.
tfl	Queryparam	No	Int	Time to live for a response in seconds. Default value is 300 seconds.	Must be a positive integer.
body	Body	Yes	json	Refer to the sample request body.	NA

## Request Details

Name	Mandatory	Description	Field Type	Constraints
resourceId	Yes	Mongo Id of the certificate in the AppViewX database.  <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  <b>Note:</b> It refers to the resourceId field in the create certificate response. It can be found using the search API using commonName, serial No. or other search parameters. </div>	String	Either resourceId or serialNumber and commonName is mandatory.
commonName	Yes	Common name of the certificate.	String	
serialNumber	Yes	Serial number of the certificate.	String	
format	Yes	Certificate downloadable format.	String	Refer to the <a href="#">Possible values for Certificate Download Format</a> table.
password	Yes	The field is mandatory for some parameters.	String	

## Possible values for Certificate Download Format

Certificate Extension	Value to be provided in payload	Password Required
.crt	CRT	No
.cert	CERT	No
.cer	CER	No
.pem	PEM	No
.der	DER	No
.cer	DERCER	No
.p7b	P7B	No
.p7c	P7C	No

Certificate Extension	Value to be provided in payload	Password Required
.pk8	PK8	No
.p12	P12	Yes
.pfx	PFX	Yes
.jks	JKS	Yes

# Chapter 58: Sync regenerate Response Structure

Response returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the enrollment request.	<a href="#">SyncReGenerateResponse</a>
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

## SyncReGenerateResponse

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Open work order request ID.	String
certificateContent	Content of certificate in Base64 encoded format.	String
encodedFormat	Certificate content encoded format.	String
certificateName	Name of the Certificate.	String

## Chapter 59: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{  
  "resourceId": "<Mongo ID of the certificate in AppviewX Database>,"  
  "commonName": "<Common Name of the certificate>,"  
  "serialNumber": "<Serial Number of the certificate>,"  
  "certificateFormat": {  
    "format": "<Cert format - check possible values table>,"  
    "password": "<password>"  
  }  
}
```

### Sample Response:

```
{  
  "response": {  
    "resourceId": "<Certificate ID>,"  
    "requestId": "<Certificate WF Request ID>,"  
    "certificateContent": "<Encoded Certificate file>,"  
    "certificateName": "<Name of the certificate>,"  
    "encodedFormat": "base64"  
  },  
  "message": "Certificate Regenerate action executed successfully",  
  "appStatusCode": "<Error Code>,"  
  "tags": {},  
  "headers": null  
}
```

## Chapter 60: Overview

The API will submit a request to suspend an existing certificate. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 61: Before you begin

Before attempting to suspend a certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 62: Request Structure

URL: /certificate/action

Type: PUT

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
resourceId	No	Unique Id of the certificate.	String	Required if the commonName and serialNumber are not specified.
commonName	No	Common name of the certificate.	String	Required if resourceId is not specified.

Name	Mandatory	Description	Field Type	Constraints
serialNumber	No	Serial number of the certificate.	String	Required if resourceId is not specified.
action	Yes	Action name for the suspend.	String	Value should be <b>Suspend</b> .
reason	Yes	Reason for suspend request.	String	NA

# Chapter 63: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the renewal request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Workorder request Id.	String
message	Success message - Suspend action triggered successfully.	NA

## Chapter 64: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Suspend action has been triggered successfully.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName/serialNumber/resourceId</code> is correct.
400 Bad Request	INVALID_REQUEST	Please give valid common name and serial number or <code>resourceId</code> .	Please give valid common name and serial number or <code>resourceId</code> .
400 Bad Request	INVALID_REQUEST	Please provide a valid action.	Please provide a valid action.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - action.	Ensure that the <code>action</code> field is available in the request payload.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - reason.	Ensure that the <code>reason</code> field is available in the request payload.
417 Expectation Failed	OPEN_WORK_ORDERS_FOUND	Since requested certificate's work order is in progress, cannot initiate another action.	Trigger the request once the open worker for the

HTTP Status code	appStatusCode	Message	Possible remediation
			certificate is completed.
406 Not Acceptable	CERT-VWF-0006	Life cycle action is unsupported by CA or another work order is in progress or certificate belongs to read group or is in Monitored status.	<p>Ensure the following -</p> <ul style="list-style-type: none"> <li>• The CA supports the suspend action.</li> <li>• There is no workOrder in progress for the specified certificate.</li> <li>• Certificate should not belong to read only group</li> <li>• Certificate is not in monitored status</li> </ul>

## Chapter 65: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "resourceId": "5f5641a7e74e187a7e3d5266",
  "commonName": "testcert5d.avx.com",
  "serialNumber": "7E:94:56:98:57:F2:0E:3B",
  "action": "Suspend",
  "reason": "Superseded"
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

```
{
  "response": {
    "resourceId": "5f4e5c2e70040d33314f0e9d",
    "message": "Suspend action triggered successfully.",
    "requestId": "241"
  },
  "message": "Suspend action has been triggered successfully",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Chapter 66: Overview

The API will submit a request to reinstate a suspended certificate. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 67: Before you begin

Before attempting to reinstate a certificate from a particular CA through AppViewX, the user has to ensure the following:

- The CA setting should have been configured in AppViewX for the CA.
- Connectivity to the CA via the chosen setting should be working fine.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 68: Request Structure

URL: /certificate/action

Type: PUT

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
resourceId	No	Unique Id of the certificate.	String	Required if the commonName and serialNumber are not specified.
commonName	No	Common name of the certificate.	String	Required if resourceId is not specified.

Name	Mandatory	Description	Field Type	Constraints
serialNumber	No	Serial number of the certificate.	String	Required if resourceId is not specified.
action	Yes	Action name for the reinstate request.	String	Value should be <b>Reinstate</b> .
reason	Yes	Reason for the request.	String	NA

# Chapter 69: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the renewal request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
resourceId	Identifier of the certificate record that has been created.	String
requestId	Workorder request Id.	String
message	Success message - Reinstate action triggered successfully.	NA

## Chapter 70: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	-	Reinstate action has been triggered successfully.	-
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName/serialNumber/resourceId</code> is correct.
400 Bad Request	INVALID_REQUEST	Please give valid common name and serial number or <code>resourceId</code> .	Please give valid common name and serial number or <code>resourceId</code> .
400 Bad Request	INVALID_REQUEST	Please provide a valid action.	Please provide a valid action.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - action.	Ensure that the <code>action</code> field is available in the request payload.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - reason.	Ensure that the <code>reason</code> field is available in the request payload.
417 Expectation Failed	OPEN_WORK_ORDERS_FOUND	Since requested certificate's work order is in progress, cannot initiate another action.	Trigger the request once the open worker for the

HTTP Status code	appStatusCode	Message	Possible remediation
			certificate is completed.
406 Not Acceptable	CERT-VWF-0006	Life cycle action is unsupported by CA or another work order is in progress or certificate belongs to read group or is in Monitored status.	<p>Ensure the following -</p> <ul style="list-style-type: none"> <li>• The CA supports the Reinstate action.</li> <li>• There is no workOrder in progress for the specified certificate.</li> <li>• Certificate should not belong to read only group.</li> <li>• Certificate is not in monitored status.</li> </ul>

# Chapter 71: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{  
  "resourceId": "5f4faf3e70040d33314f1142",  
  "commonName": "testcert8g.appviewx.plus",  
  "serialNumber": "0D:A9:2D:8C:90:BB:90:B0:CE:7D:6A:76:BF:70:75:81",  
  "action": "Reinstate",  
  "reason": "Test"  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{  
  "response": {  
    "resourceId": "5f4faf3e70040d33314f1142",  
    "message": "Reinstate action triggered successfully.",  
    "requestId": "216"  
  },  
  "message": "Reinstate action has been triggered successfully",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

## Chapter 72: Overview

The API allows the user to search for one or more certificates against the given searchable input. The complete detail of the matching certificate(s) will be returned.

# Chapter 73: Request Structure

URL: /certificate/search

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered(E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
input	Yes	Search parameters such as keyword or free text search.	input	NA
filter	Yes	Pagination and sort params for the search.	filter	NA

input

Name	Mandatory	Description	Field Type	Constraints
category	No	Specifies the certificate category. Possible categories are Server, Client, Code Signing and Device.	String	Possible values - Server Client Code Signing Device Default value - Server.
resourceId	No	Id of the certificate.	String	NA
freeSearch	No	Key for free text search. Search will be performed on all cert fields based on the free text input provided.	String	NA
keywordSearch	No	Search by certificate attributes based on the predefined key words.	keywordSearch	NA

keywordSearch

Search keywords and their description.

Name	Mandatory	Description	Field Type	Constraints
subject:o	No	Subject organization name.	String	NA
subject:ou	No	Subject organization unit.	String	NA
subject:cn	No	Subject common name.	String	NA

Name	Mandatory	Description	Field Type	Constraints
subject:l	No	Subject location.	String	NA
subject:st	No	Subject state.	String	NA
subject:c	No	Subject country.	String	NA
certversion	No	Certificate version.	String	NA
certserialno	No	Certificate serial no.	String	Serial no of the certificate to be specified in the same format as in example- 1C:DB:14:3D:5C:98:EC:00:D0:C1:D1:63:78:B4:AA:42
certissuer	No	Issuer common name.	String	
certstatus	No	Certificate status.	String	Possible values -  Managed  New Certificate  Monitored  Discovered
certgroup	No	Certificate group.	String	
expirydate	No	Certificate expiry date in MM/DD/YYYY format.	String	Example - 03/26/2022
expiryyear	No	Certificate expiry year.	String	Example - 2022
expirymonth	No	Certificate expiry month.	String	Example - 03
discoverysource	No	Any of the following -	String	Applicable for discovered certificates.

Name	Mandatory	Description	Field Type	Constraints
		<ul style="list-style-type: none"> <li>• Device name for certificates discovered from device.</li> <li>• For the certificates discovered from CA, the value will be &lt;&lt;<b>CA name</b>&gt;&gt;::&lt;&lt;<b>CA setting name</b>&gt;&gt;. For example, if CA name is Ejbca and CA setting name is <b>Setting1</b> then the value is <b>Ejbca::Setting1</b>.</li> </ul>		
ca	No	Certificate Authority name.	String	This is fixed, how to make user understand what are the options available.
application	No	CA name/ auto enrollment protocol name/ device vendor name.	String	<p>Applicable for certificates discovered from <b>CA/device</b> and also the certificates enrolled via <b>auto enrollment protocols</b>.</p> <p>Possible values for the auto enrollment protocols -</p> <p>SCEP</p> <p>EST</p> <p>ACME</p>

Name	Mandatory	Description	Field Type	Constraints
				SCEP_MS_INTUNE
issuer	No	Issuer Common Name.	String	Need to mention to either get it from our inventory or certificate data
hash	No	Hash function.	String	Possible values - SHA160, SHA224, SHA256, SHA384, SHA512
key	No	Key Type.	String	Possible values - RSA, EC, DSA
keylength	No	Bit length depending on the key type chosen.	String	Possible values - To be taken from defined CA policy
caSettings	No	CA setting name.	String	Possible values - To be taken from defined CA policy
ellipticcurve	No	Elliptic curve name.	String	Possible values - To be taken from defined CA policy

filter

Name	Mandatory	Description	Field Type	Constraints
max	No	The number of entries from the start index to be made available. For example, if the response is expected to have skipped the first 100 and show the next 50, then the start index has to be 101 and max has to be 50.	Integer	Default value is <b>100</b> .
start	No	Start index from which the response has to be available. For example, if the response has to skip the first 100 and show the next 50, then the start index has to be 101.	Integer	Default value is <b>0</b> .
sortOrder	No	Ascending or descending order of sort.	String	Possible values - <b>asc, desc</b> . Default value is asc.

Name	Mandatory	Description	Field Type	Constraints
sortColumn	No	Sort by column.	String	Default value is <b>_id</b> .  Default value is <b>asc</b> .

# Chapter 74: Response Structure

200 OK returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes.	response
message	Failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
response	Contains the response attributes.	Inventory response
message	<p>If matching records are found for the search query - Matching results found for the given input.</p> <p>If no records are available for the search query - No matching records found for the given input.</p>	NA

Inventory response

Name	Description	Field Type
objects	Certificate metadata	Array of Certificate json.
totalRecords	Total no of records matching the search criteria.	String
obtainedRecords	No of records fetched for the query.	String
obtainedRecordRange	Index of the records fetched.	obtainedRecordRange

obtainedRecordRange

Name	Description	Field Type
start	Start index of the fetched record.	String
end	End index of the fetched record.	String

## Chapter 75: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
200 OK	NA	<p>If matching records are found for the search query -</p> <p><b>Matching results found for the given input.</b></p> <p>If no records are available for the search query -</p> <p><b>No matching records found for the given input.</b></p>	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	INVALID_REQUEST	please input a valid sort order.	Ensure that valid value is provided for <code>sortOrder</code> .
400 Bad Request	INVALID_REQUEST	Invalid request specified.	<p>Possible causes -</p> <ul style="list-style-type: none"> <li>• Either <code>filter</code> is not available or <code>filter</code> value is invalid in the <code>payload</code>.</li> <li>• Value provided for <code>filter.max</code> is invalid.</li> </ul>

## Chapter 76: Sample request/response

- Sample Request:
- Sample Response:

### Sample Request:

```
{
  "input":{
    "category":"Server",
    "keywordSearch": {
      "certserialno":"7E:94:56:98:57:F2:0E:3B"
    }
  },
  "filter":{
    "max":"100",
    "start":"1",
    "sortColumn":"commonName",
    "sortOrder":"desc"
  }
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

```
{
  "response": {
    "statusCode": 200,
    "response": {
      "objects": [
        {
          "commonName": "testcert5d.avx.com",
          "serialNumber": "7E:94:56:98:57:F2:0E:3B",
          "issuerCommonName": "Ejbc New Intermediate CA",
          "status": "Managed",
        }
      ]
    }
  }
}
```

```

"associatedObjects": [],
"discoverySources": [],
"subjectOrganization": "AppViewX",
"subjectOrganizationUnit": "PE",
"subjectLocality": "Plano",
"subjectState": "Texas",
"subjectCountry": "US",
"issuerOrganization": "",
"issuerOrganizationUnit": "",
"issuerLocality": "",
"issuerState": "",
"issuerCountry": "",
"version": "3",
"validFrom": 1598943490000,
"validTo": 1607583490000,
"validFor": "90 day(s) 18 hr(s) 8 min(s)",
"keyAlgorithmAndSize": "RSA 2048",
"signatureAlgorithm": "SHA256withRSA",
"signatureHashAlgorithm": "SHA256",
"keyUsage": "DigitalSignature, NonRepudiation, KeyEncipherment",
"extendedKeyUsage": "Client Authentication(1.3.6.1.5.5.7.3.2) Server Authentication(1.3.6.1.5.5.7.3.1) ",
"basicConstraints": "Subject Type=End entity, Path Length=none",
"group": "Default",
"subjectAlternativeNames": [
"URL : http://testcert5.avx.com",
"Email address : test@testcert5.avx.com",
"DNS : testcert5.avx.com",
"IP Address : 192.168.142.162"
],
"complianceStatus": "Compliant",
"applications": [],
"expiryStatus": "Valid",
"permission": "RW",
"category": "Server",
"uuid": "4f4e23cefea2297a4e7d6b202ceee5ba47a20c89",
"certificateAuthority": "Ejbca",
"authorityKeyIdentifier": "69:15:BC:24:60:6B:73:96:8B:F3:3A:ED:C4:38:EB:58:A6:3F:4B:5B",

```

```

"subjectKeyIdentifier": "0D:31:76:0B:F1:41:AD:A6:39:6B:79:A6:43:8D:62:07:CC:F5:94:FB",
"authorityInfoAccess": [],
"certificatePolicies": [],
"crldistributionPoints": [
  "CrldistributionPoint : [ name : , url : http://192.168.12.139:8080/ejbca/publicweb/webdist/certdist?cmd=crl&issuer=CN=Ejbca%20New%20Intermediate%20CA ]"
],
"thumbprintAlgorithm": "SHA-1",
"thumbPrint": "9F:8E:0D:B6:A6:25:D4:05:E3:2B:5E:F7:85:1B:79:83:CB:EA:3C:DB",
"type": "Others",
"genericFields": {
  "vs_ip_Ejbca": "192.168.142.162",
  "device_name_Ejbca": "test_device"
},
"csrGenerationSource": "appviewx",
"newConnectors": [],
"csrAvailable": true,
"autoRenewDate": "",
"missingParamsForAutoRenew": "CSR parameters are available for certificate renewal",
"suspendedCertificate": false,
"mailAddress": "test@test.com",
"streetAddress": "",
"postalCode": "",
"requestIds": [
  "R147",
  "R241"
],
"publicKey":
"30:82:01:0A:02:82:01:01:00:8B:64:08:1A:2B:40:B9:2F:7C:48:65:78:98:B3:A6:87:9D:E2:49:CF:25:A5:27:14:EE:A0:45:CE:E3:32:21:DC:8D:A5:AE:78:36:EF:5A:9
1:D6:7E:D9:F4:04:C5:A9:9A:38:F7:D4:C6:50:B2:4C:96:1C:AE:69:23:19:7C:45:87:07:37:4C:2C:07:25:87:6C:93:D0:B6:D0:5D:55:FC:29:A0:48:6A:75:CF:12:6C:41
:AE:90:BD:E7:89:C5:C4:8D:81:78:3D:DD:0B:29:61:92:55:0E:5A:8F:44:8F:44:CE:07:D8:11:95:7F:03:33:DD:86:E0:9A:69:42:AD:C9:67:44:91:FF:B2:02:ED:1A:16:
9C:B5:CC:03:92:3B:1A:17:52:D5:32:56:BF:52:D1:7C:8F:96:43:18:6E:43:22:55:32:01:4D:44:AA:52:13:6A:BB:B6:1B:DD:8D:76:D3:C6:C2:D0:D9:E0:9E:B8:4C:DD:
0A:8D:5C:5E:20:02:79:20:26:84:7F:35:B7:EF:E0:98:D1:FC:37:D4:1B:B8:4C:07:64:00:E8:18:DC:63:98:85:56:F3:D0:F7:E4:EF:F1:C1:0F:B3:F2:86:48:73:36:A5:5D
:AC:A9:A9:DE:F9:62:44:D0:C8:8D:F1:03:5F:3E:03:08:C8:0C:92:00:CF:72:55:02:03:01:00:01",
"issuedByRootCertificate": false,
"privatekeyAvailable": true,
"resourceId": "5f4e5c2e70040d33314f0e9d"
}

```

```
],  
  "totalRecords": 1,  
  "obtainedRecords": 1,  
  "obtainedRecordRange": {  
    "start": 1,  
    "end": 1  
  }  
},  
  "message": "Matching results found for the given input.",  
  "appStatusCode": null,  
  "tags": null  
},  
  "message": null,  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

# Chapter 77: Overview

The API will add various attributes like name, label, etc. to the certificate.

# Chapter 78: Request Structure

URL: /certificate/attribute

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
	Yes	List of the certificate attributes.	Array of Attribute	Array cannot be empty. At Least one entry should be added.

Attribute

Name	Mandatory	Description	Field Type	Constraints
name	Yes	Name of the custom attribute.	String	Must not start with special characters. No special characters except -, _ are allowed.
label	Yes	Label to be shown in the UI for the custom attribute.	String	Must not start with special characters. No special characters except -, _ are allowed.
value	No	Default value for the attribute.	String	NA
mandatory	No	Specifies whether the attribute is mandatory or not.	Boolean	NA
helpInfo	No	Help info about the field.	String	NA

# Chapter 79: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	

response

Name	Description	Field Type
successfullFields	List of fields that have been added successfully.	Array of CertificateGenericField
failedFields	List of fields that have not been added successfully.	Array of CertificateGenericField

## Chapter 80: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	-	<p>If all fields are added -</p> <p><b>Certificate attribute added successfully</b></p> <p>If the addition of some fields have failed -</p> <p><b>Certificate attribute added successfully. Failed fields are [ &lt;&lt;failed field names&gt;&gt; ]</b></p> <p>If the addition of all fields failed -</p> <p><b>Certificate attribute addition failed.</b></p>	<p>Possible reasons for the field addition failure</p> <ul style="list-style-type: none"> <li>• Name or label not provided for the failed field.</li> <li>• Value provided for the name or label does not meet the expected pattern.</li> <li>• Attribute with the provided name already exists.</li> </ul> <p>Validate the given input based on the above mentioned reasons.</p>
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - payload.	Payload array cannot be empty. At least one attribute must be provided.
400 Bad Request	avx-common-028	Invalid / Incorrect payload.	Check and ensure that the value provided for all the fields in the payload is proper.

# Chapter 81: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
[
{
  "name": "externalemail",
  "label": "External_Email",
  "value": "test@test.com",
  "mandatory": true,
  "helpInfo": "The external email id"
},
{
  "name": "priority",
  "label": "priority",
  "value": "1",
  "mandatory": false
}
]
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": {
    "failedFields": [],
    "successfullFields": [
      {
        "id": "externalemailcustomAttributecustomAttribute",
        "type": "text",
        "value": "test@test.com",
        "values": null,

```

```
"label": "External_Email",
"searchKey": "externalemail",
"errorCode": null,
"mandatory": true,
"validation": null,
"divClasses": null,
"classNames": null,
"category": "certAttributes",
"certificateAuthority": null,
"regexValue": null,
"helpInfo": "The external email id",
"name": "externalemail",
"mandatoryFor": null,
"_id": "externalemail"
},
{
  "id": "prioritycustomAttributecustomAttribute",
  "type": "text",
  "value": "1",
  "values": null,
  "label": "priority",
  "searchKey": "priority",
  "errorCode": null,
  "mandatory": false,
  "validation": null,
  "divClasses": null,
  "classNames": null,
  "category": "certAttributes",
  "certificateAuthority": null,
  "regexValue": null,
  "helpInfo": null,
  "name": "priority",
  "mandatoryFor": null,
  "_id": "priority"
}
]
},
```

```
"message": "Certificate attribute added successfully",  
"appStatusCode": null,  
"tags": {},  
"headers": null  
}
```

# Chapter 82: Overview

The API will update the existing certificate attributes.

# Chapter 83: Request Structure

URL: /certificate/attribute

Type: PUT

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
	Yes	List of the certificate attributes	Array of Attribute	Array cannot be empty. At Least one entry should be added.

Attribute

Name	Mandatory	Description	Field Type	Constraints
name	Yes	Name of the existing custom attribute.	String	NA
label	Yes	Label to be shown in the UI for the custom attribute.	String	Must not start with special characters. No special characters except -, _ are allowed.
value	No	Default value for the attribute.	String	NA
mandatory	No	Specifies whether the attribute is mandatory or not.	Boolean	NA
helpInfo	No	Help info about the field.	String	NA

# Chapter 84: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

response

Name	Description	Field Type
updatedGenericFields	List of fields that have been updated successfully.	Array of CertificateGenericField
failedGenericFields	List of fields that have not been added successfully.	Array of CertificateGenericField

## Chapter 85: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Certificate attribute updated successfully.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - payload.	<code>Payload</code> array cannot be empty. At least one attribute must be provided.
400 Bad Request	avx-common-028	Invalid / Incorrect payload.	Check and ensure that the value provided for all the fields in the <code>payload</code> is proper.

## Chapter 86: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
[
{
  "name": "approver",
  "label": "approver",
  "value": "cert approver",
  "mandatory": true
}
]
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

```
{
  "response": {
    "updatedGenericFields": [
      {
        "id": "approvercustomAttribute",
        "type": "text",
        "value": "cert approver",
        "values": null,
        "label": "approver",
        "searchKey": "approver",
        "errorCode": null,
        "mandatory": true,
        "validation": null,
        "divClasses": null,
        "classNames": null,
        "category": "certAttributes",
```

```
"certificateAuthority": null,  
"regexValue": null,  
"helpInfo": null,  
"name": "approver",  
"mandatoryFor": null,  
"_id": "approver"  
}  
],  
"failedGenericFields": [],  
"addedGenericFields": []  
},  
"message": "Certificate attribute updated successfully",  
"appStatusCode": null,  
"tags": {},  
"headers": null  
}
```

# Chapter 87: Overview

The API will fetch all the certificate attributes.

# Chapter 88: Request Structure

URL: /certificate/attribute

Type: GET

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external)	String	NA

# Chapter 89: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
	List of certificate attributes.	Array of CertificateGenericField

## Chapter 90: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Certificate attributes fetched successfully or  No certificate attributes available.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.

# Chapter 91: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

No payload for GET request

## Sample Response:

```
{
  "response": [
    {
      "id": "approvercustomAttributecustomAttribute",
      "type": "text",
      "value": "cert approver",
      "values": null,
      "label": "approver",
      "searchKey": "approver",
      "errorCode": null,
      "mandatory": true,
      "validation": null,
      "divClasses": null,
      "classNames": null,
      "category": "certAttributes",
      "certificateAuthority": null,
      "regexValue": null,
      "helpInfo": null,
      "name": "approver",
      "mandatoryFor": null,
      "_id": "approver"
    }
  ],
  "message": "Certificate attributes fetched successfully",
  "appStatusCode": null,
}
```

```
"tags": {},  
"headers": null  
}
```

# Chapter 92: Overview

- Request Structure
- Response Structure
- Status codes
- Sample Request/Response

## Request Structure

URL: */certificate/attribute*

Type: DELETE

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b> .
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external)	String	NA
name	Query	Name of the certificate attribute that has to be deleted.	String	NA

## Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Message indicating whether the field has been deleted successfully or not. Message - <b>Deleted Successfully - [approver1]. Failed to delete - []</b> .	String
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

## Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Certificate attribute deleted successfully.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for name query param is correct.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - name.	Ensure that the name query param is available in the request.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

No payload for Delete request.

### Sample Response:

```
{  
  "response": "Deleted Successfully - [approver2]. Failed to delete - []",  
  "message": "Certificate attribute deleted successfully",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

## Chapter 93: Overview

The API will download a certificate in PEM format. Optionally, the user can request the certificate to be downloaded along with the key.

# Chapter 94: Request Structure

URL: `/certificate/download`

Type: GET

Name	ParameterType	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	Mandatory
gwsource	Query	Source from which the request is triggered (E.g. external)	String	Mandatory
resourceId	Query	Unique Id of the certificate to be downloaded.	String	Mandatory
keyRequired	Query	Specifies whether private key needs to be downloaded.	Boolean	Optional
isCACertificate	Query	Specifies whether the certificate being downloaded is a CA certificate.	Boolean	Optional

# Chapter 95: Response Structure

## Success Response

200 Ok response with attachment of type pem.

Response content-type - application/octet-stream

## Error Response

Error response will have following parameters:

Name	Description	Field Type
message	Failure description	String
appStatusCode	Application specific status code for the error response.	String
tags	More info in case of failure response.	NA

## Chapter 96: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>resourceId</code> param refers to the id of an existing certificate.
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
417 Expectation Failed	NOT_A_VALID_CERT	Invalid certificate.	Certificates that have not yet been issued cannot be downloaded. Ensure that the <code>resourceId</code> refers to the certificate which has been issued.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <code>resourceId</code> .	Ensure that the <code>resourceId</code> param is available in query params.
417 Expectation Failed	CERT-KEY-0006	Private key access is restricted in <<Default>> policy.	Retry to download after enabling the <b>Enable Access to Private Key</b> flag in the Certificate Policy.

# Chapter 97: Sample Response

Downloaded file will be a pem file and the content of the downloaded file will be as mentioned below:

```
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmijZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDEwNzAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxZDZANBgNVBAoTBkFtYXpvcjEzZmBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo91ghYPI0hAEvrAlhtOgQ3pOsqTQnroBvo3bSMgHFzZM
9O6ll8c+6zf1tRn4SWiw3te5dugdYZ6k/ol2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWlbcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrlqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBbYEFiQYzIU07LwMJQuCFmxc71QTgoIMA0GCSqGSib3DQEBcWUA
A4IBAQCYY8jdaQZChGsV2USggNiMOruYou6r4IK5lpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjImCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhbl1Bjijt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWIJbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bidZwgJcJmApzyMZFo6lQ6XU
5Msl+yMRQ+hDKXJioaldXgJkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----
```

## Chapter 98: Overview

The API will download a certificate in the requested format. The user can request for private key download in requested format. Key can either be requested with or without encryption. Also, the user can specify whether the complete certificate chain needs to be downloaded.

# Chapter 99: Request Structure

URL: `/certificate/download/format`

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
format	Yes	Format required.	String	Format to download Possible values are:

Name	Mandatory	Description	Field Type	Constraints
				1. CRT 2. CERT 3. CER 4. PEM 5. DER 6. DERCER 7. P7B 8. P7C 9. PK8 10. P12 11. PFX 12. JKS
password	No	Password for the certificate. Applicable for the following formats.  1. P12 2. PFX 3. JKS 4. PK8	String	Mandatory if the format is -  1. P12 2. PFX 3. JKS
commonName	Yes	Common name of the certificate.	String	NA
serialNumber	Yes	Serial number of the certificate.	String	NA
isChainRequired	No	Specifies whether the complete certificate chain has to be	Boolean	NA

Name	Mandatory	Description	Field Type	Constraints
		downloaded or the end certificate alone has to be downloaded.		
isKeyEncryptionRequired	No	Specifies whether the private key needs to be encrypted using password.	Boolean	<p>This is applicable for PK8 format.</p> <p>If <b>true</b>, make sure that the Certificate policy allows private key access.</p> <p>If <b>true</b>, then password must be provided in the request payload.</p>

# Chapter 100: Response Structure

## Success Response

200 Ok response with attachment of request format.

Response content-type - application/octet-stream

## Error Response

Error response will have following parameters:

Name	Description	Field Type
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

## Chapter 101: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
404 Not Found	NO_RECORDS_FOUND	No matching records found.	Check and ensure that the value provided for <code>commonName</code> and <code>serialNumber</code> pertain to that of a valid existing certificate.
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
417 Expectation Failed	NOT_A_VALID_CERT	Invalid certificate.	Certificates that have not yet been issued cannot be downloaded. Ensure that the <code>resourceId</code> refers to the certificate which has been issued.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <code>&lt;&lt;field&gt;&gt;</code> .	Ensure that the <code>&lt;&lt;field&gt;&gt;</code> is available in the request payload.
400 Bad Request	INVALID_FORMAT	Invalid format specified.	Ensure that a valid value is provided for the format field in the request.
417 Expectation Failed	PASSWORD_FORMAT_WRONG	Password must have at least eight characters and contain at least one lowercase letter, one uppercase letter, one numeric digit, and one special character.	Password provided in the request payload must have at least eight characters and contain at least one lowercase letter, one uppercase letter, one numeric digit, and one special character.

HTTP Status code	appStatusCode	Message	Possible remediation
417 Expectation Failed	CERT-KEY-0006	Private key access is restricted in <<Default>> policy.	Retry to download after enabling the Enable Access to Private Key flag in the Certificate Policy.
417 Expectation Failed	CERT-HLST-0007	Failed to download as the given extension is not supported.	Ensure that a valid value is provided for the format field in the request.

## Chapter 102: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{  
  "commonName": "testcert7a.appviewx.com",  
  "serialNumber": "07:2B:84:01:27:5A:77:63:38:C8:72:09:BE:CD:1F:D9:78:BA:BD",  
  "format": "P12",  
  "password": "Password@123",  
  "isChainRequired": true,  
  "isKeyEncryptionRequired": false  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

File will be downloaded.

## Chapter 103: Overview

The fields are part of APIs to add, update and get certificate attributes. Since this structure is common to these APIs they are grouped together in this section.

# Chapter 104: Common Response Objects

## CertificateGenericField

Name	Description	Field Type
<code>_id</code>	Unique identifier for the attribute.	String
<code>name</code>	Name of the attribute.	String
<code>type</code>	Type of the attribute.	String
<code>value</code>	Default value for the attribute.	String
<code>label</code>	Label to be shown in the UI for the custom attribute.	String
<code>searchKey</code>	Keyword for searching the certificate based on the attribute.	String
<code>mandatory</code>	Specifies whether the attribute is mandatory or not.	Boolean
<code>category</code>	Category the attribute which will always be certAttributes for the certificate attributes.	String
<code>helpInfo</code>	Help info about the field.	String

# Chapter 105: Overview

The API will push the certificate and its private keys to the firewall profile and associate that to the server profiles. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 106: Before you Begin

Before attempting to push certificate to a particular Firewall device through AppViewX, the user has to ensure the following:

- Firewall devices should be added in AppViewX.
- The device should be in Managed state.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 107: Request Structure

URL: */certificate/pushToDevice*

Type: POST

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external)	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
certificateId	Yes	Resource id of the certificate	String	

Name	Mandatory	Description	Field Type	Constraints
selectedProfiles	Yes	Firewall profile id	String	<ul style="list-style-type: none"> <li>If Profile connector specified then certificate will be pushed and bound to the profile</li> <li>If Default connector specified then certificate will be only pushed</li> </ul>
certificateDetails	Yes	Certificate details for the Firewall devices	Any of the following - <ul style="list-style-type: none"> <li>Fortinet certificate details</li> <li>Paloalto certificate details</li> </ul>	NA
pushDetails	No	Other push details	pushDetails	NA

## Fortinet certificate details

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Certificate type in which format the certificate to be pushed in a firewall device.	String	Value should be PEM-.pem
certificateName	Yes	Certificate name in which name the certificate to be pushed	String	The certificate file name should not start and end with special characters except @, #, \$, %, ^, &, -, {, }, [, ],  , :, <, >, (, ), ;, . It should

Name	Mandatory	Description	Field Type	Constraints
		in a firewall device.		not be the same as intermediate file name and CA file name.
privateKeyPassword	No	Password for the private key pushed to the firewall device	String	NA
pushRootAndIntermediateCertificates	Yes	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
rootCertificateFileName	Yes	Certificate name in which name the root certificate to be pushed in a firewall device.	String	CA file name should not start and end with special characters. No special characters except !, @, #, \$, %, ^, &, _ , { , } , [ , ] ,   , ; , < , > , ( , ) , ; , . and should not be the same as certificate file name and CA file name.
intermediateCertificateFileName	Yes	Certificate name in which name the intermediate certificates to be pushed in a firewall device.	List of Strings	Intermediate file names should not start and end with special characters. No special characters except !, @, #, \$, %, ^, &, _ , { , } , [ , ] ,   , ; , < , > , ( , ) , ; , . and should not be the same as certificate file name and CA file name.

Palo Alto certificate details

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Certificate type in which format the certificate to be pushed in a firewall device.	String	Value should be PEM-.pem or PKCS12-.p12
certificateName	Yes	Certificate name in which name the certificate to be pushed in a firewall device.	String	The certificate file name should not start and end with special characters except @, #, \$, %, ^, &, _, {, }, [, ],  , :, <, >, (, ), ;, . It should not be the same as intermediate file name and CA file name.
pushRootAndIntermediateCertificates	Yes	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
rootCertificateFileName	Yes	Certificate name in which name the root certificate to be pushed in a firewall device.	String	CA file name should not start and end with special characters. No special characters except !, @, #, \$, %, ^, &, _, {, }, [, ],  , :, <, >, (, ), ;, . and should not be the same as certificate file name and CA file name.
intermediateCertificateFileName	Yes	Certificate name in which name the intermediate	List of Strings	Intermediate file names should not start and end with special characters. No special characters except !, @, #, \$, %, ^, &, _, {, }, [, ],  , :, <, >, (, ), ;, . and should

Name	Mandatory	Description	Field Type	Constraints
		certificates to be pushed in a firewall device.		not be the same as certificate file name and CA file name.

## pushDetails

Name	Mandatory	Description	Field Type	Constraints
preValidationScriptPath	No	Location of the pre push script	String	NA
postValidationScriptPath	No	Location of the post push script	NA	NA
overwrite	No	Whether the certificate needs to be overwritten if already available	Boolean (true or false)	NA
pushAutomatically	No	Whether the certificate needs to be pushed automatically when renewed	Boolean (true or false)	NA

# Chapter 108: Response Structure

202 Accepted returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the push request	List of response
message	Success message of the action or failure description in case of error	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

response

Name	Description	Field Type
requestId	Request Id for push action for the application connector	String
connectorId	Application connector Id	String

## Chapter 109: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	1 connector(s) saved and push operation has been triggered.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>	Check and ensure that valid value is provided for <<field name>> field in the request.
417 Expectation failed	FIELD_VALUE_INVALID	Invalid value - <<field name>>	Check and ensure that a valid value is provided for the <<field name>> field in the request.
404 Not Found	NO_RECORDS_FOUND	No matching records found - certificate not found.	Please provide the correct value for the field <code>certificateId</code>
400 Bad Request	INVALID_REQUEST	<code>selectedProfiles</code> are already available in the specified certificate	Provide different value for the <code>selectedProfiles</code> field.
417 Expectation failed	CERT-APP-0016	Connector with profiles {} already exists	Profile connector already available for the selected certificate. Please change the <code>certificateId</code> or delete the existing connector.
500	avx-common-011	Error while processing	NA

HTTP Status code	appStatusCode	Message	Possible remediation
Internal Server Error			

# Chapter 110: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "certificateDetails": {
    "certificateType": "PEM-.pem",
    "certificateName": "testfile",
    "pushRootAndIntermediateCertificates": true,
    "rootCertificateFileName": "testrootfile.pem",
    "intermediateCertificateFileName": [
      "testinterfilelevel1.pem"
    ]
  },
  "pushDetails": {
    "preValidationScriptPath": "",
    "postValidationScriptPath": "",
    "overwrite": false,
    "pushAutomatically": false
  },
  "certificateId": "5f5b1fac77534426423bab57",
  "selectedProfiles": [
    "xxx.xxx.xx.xxx:@Template_001:@shared:@IKE Gateway Profile:@TEST"
  ]
}
```

## Sample Response:

```
{
  "response": [
    {
      "requestId": "3",
      "connectorId": "xxx.xxx.xx.xxx:@Template_001:@shared:@IKE Gateway Profile:@TEST:@389ed6f96387e3002cc6ac0678c07ffe70459abf"
    }
  ]
}
```

```
}  
],  
"message": "1 connector(s) saved and push operation has been triggered.",  
"appStatusCode": null,  
"tags": {},  
"headers": null  
}
```

# Chapter 111: Overview

The API will rollback the certificate and it's private keys to the firewall device and associate that to the firewall profiles. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 112: Before you begin

Before attempting to rollback a certificate to a particular Firewall device through AppViewX, the user has to ensure the following:

- Firewall devices should be added in AppViewX.
- The device should be in Managed state.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 113: Request Structure

URL: /certificate/rollback

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsources	Query	Source from which the request is triggered(E.g. external).	String	NA
autoApproval	Query	Auto approval needed for the action or not.	String	Value should be <b>yes</b> .
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
applicationConnectorIds	Yes	Application connector id.	String	NA

# Chapter 114: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the rollback request.	List of response
message	Success message of the action or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	

response

Name	Description	Field Type
requestId	Request Id for rollback action for the application connector.	String
connectorId	Application connector Id.	String

## Chapter 115: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	-	App connector rollback action initiated for 1 connector(s).	-
202 Accepted	-	Operation cannot be completed for one or more devices as the 'Resource' allocated to you does not have write permission.	User does not have access for the device.
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>.	Check and ensure that valid value is provided for <<field name>> field in the request.
417 Expectation failed	FIELD_VALUE_INVALID	Invalid value - <<field name>>.	Check and ensure that valid value is provided for the

HTTP Status code	appStatusCode	Message	Possible remediation
			<<field name>> field in the request.
417 Expectation failed	CERT-APP-0012	Application connector ids cannot be empty.	Please provide value for the field <code>applicationConnectorIds</code> .
417 Expectation failed	ERR_APPLICATION_CONNECTOR_LIST_RETRIVAL	Unable to retrieve connector information.	Connector may not be available. Please provide correct value for the field <code>applicationConnectorIds</code> .
417 Expectation failed	ERR_APP_CONNECTORS_NOT_FOUND	Application connector(s) not found.	Connector may not be available. Please provide correct value for the field <code>applicationConnectorIds</code> .
417 Expectation failed	ERR_INITIALIZE_ROLLBACK_REQUEST	Unable to initialize rollback request.	
500  Internal Server Error	avx-common-011	Error while processing.	

## Chapter 116: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{  
  "applicationConnectorIds": [ "xx.xxx:@clientsl-insecure-compatible:@Common:@c46ec8a04da701721159ce0c3cf772367ade58cb"  
]  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

```
{  
  "response": "Success",  
  "message": "App connector rollback action initiated for 1 connector(s).",  
  "appStatusCode": null,  
  "tags": null,  
  "headers": null  
}
```

# Chapter 117: Overview

This API will fetch the available profile details for a particular Server device.

# Chapter 118: Before you begin

Before attempting to fetch available profiles for a particular Server device through AppViewX, the user has to ensure the following:

- Server devices should be added in AppViewX.
- The device should be in Managed state.

# Chapter 119: Request Structure

URL: /certificate/profiles

Type: GET

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	The value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external)	String	NA
category	Query	Specifies the device category. It is a mandatory field.	String	<b>Possible values:</b> ADC, Server, Firewall
vendor	Query	Vendor for the chosen device. For example, <b>Apache</b> is a vendor for the Server category. It is a mandatory field.	String	NA
certificateUuid	Query	Uuid of the certificate	String	NA
deviceName	Query	Name of the device as per AppViewX Device Inventory	String	NA
inventory	Query	Name of AppViewX inventory where the certificate is present	String	<b>Possible values:</b>

Name	Type	Description	Field Type	Constraints
				Server, client, code signing, device

# Chapter 120: Response Structure

202 Accepted returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response attributes for the fetch profiles request	List of <a href="#">response</a>
message	Success message of the action or failure description in case of error	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

response

Name	Description	Field Type
objects	List of available device profile Ids.	List of String
totalRecords	Total number of profiles fetched.	Integer
obtainedRecords	Total number of profiles fetched.	Integer
obtainedRecordRange	Range of record found.	Object

## Chapter 121: Status Codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	<<No. of profile Ids>> record(s) found	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	The mandatory field is missing or invalid - <<field name>>	Check and ensure that valid value is provided for <<field name>> field in the request.
404 Not Found	NO_RECORDS_FOUND	No matching records found	Please provide correct values for all the fields.
500 Internal Server Error	avx-common-011	Error while processing	NA

# Chapter 122: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

Sample Request:

No payload for GET request

URL: *https://<APPVIEWX\_GATEWAY\_IP>:<APPVIEWX\_GATEWAY\_PORT>/avxapi/certificate/profiles?gwkey=f000ca01&gwsorce=external&category=Server&vendor=<Vendor>&certificateUid=<resourceId>&deviceName=<deviceName>&inventory=server*

## Sample Response:

```
{
  "response":{
    "objects":[
      "ApacheLinux",
      "ApacheLinux:@/etc/apache2:@xxx.xxx.xxx.xxx:@443:@/etc/apache2/sites-enabled/test.conf",
      "ApacheLinux:@/etc/apache2:@_default_:@443:@/etc/apache2/sites-available/default-ssl.conf",
      "ApacheLinux:@/etc/apache2:@_default_:@443:@/etc/apache2/sites-enabled/random-ssl.conf"
    ],
    "totalRecords":4,
    "obtainedRecords":4,
    "obtainedRecordRange":{
      "start":0,
      "end":0
    }
  },
  "message":"4 record(s) found",
  "appStatusCode":null,
  "tags":{
  },
}
```

```
"headers":null  
}
```

## Chapter 123: Overview

The API will push the certificate and its private keys to the Server device and associate that to the server profiles. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 124: Before you begin

Before attempting to push certificate to a particular Server device through AppViewX, the user has to ensure the following:

- Server devices should be added in AppViewX.
- The device should be in Managed state.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 125: Request Structure

URL: `/certificate/pushToDevice`

Type: `POST`

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
certificateId	Yes	Resource id of the certificate.	String	

Name	Mandatory	Description	Field Type	Constraints
<a href="#">selectedProfiles</a>	Yes	Server profile id.	String	<ul style="list-style-type: none"> <li>• If Profile connector specified then certificate will be pushed and bound to the profile.</li> <li>• If Default connector specified then certificate will be only pushed.</li> </ul>
<a href="#">certificateDetails</a>	Yes	Certificate details for the Server devices.	Any of the following - <ul style="list-style-type: none"> <li>• <a href="#">Certificate details for IIS Server</a></li> <li>• <a href="#">Certificate details for Microsoft PC Server</a></li> <li>• <a href="#">Certificate details for Microsoft Server</a></li> <li>• <a href="#">Apache Linux certificate details</a></li> <li>• <a href="#">Tomcat Linux certificate details</a></li> <li>• <a href="#">Weblogic Linux certificate details.</a></li> <li>• <a href="#">Websphere Linux certificate details</a></li> <li>• <a href="#">IPlanet certificate details</a></li> </ul>	NA

Certificate details for IIS Server

Name	Mandatory	Description	Field Type	Constraints
locationType	Yes	Determines whether certificate has to be pushed into Certificate store or Centralized file system	String	Allowed values <ul style="list-style-type: none"> <li>• Certificate store</li> <li>• File system</li> </ul>
keyStoreLocation	Yes	If locationType Certificate store chosen, keyStoreLocation determines under which sub store the certificate has to be pushed	String	Mandatory for locationType Certificate store <p>Allowed values</p> <ul style="list-style-type: none"> <li>• Personal</li> <li>• Web Hosting</li> </ul>
certificateLocation	No	If locationType File system chosen, the actual folder location where pfx file has to be placed	String	Applicable only if IIS is managed with Centralized file system <p>If certificateLocation left blank, location specified during IIS device addition taken as location</p>
certificateFileName	Yes	Certificate friendly name with which certificate to be pushed into Certificate store / Certificate file name with which the certificate to be pushed into File system	String	The certificate file name should not start and end with special characters except @, #, \$, %, ^, &, _, {, }, [, ],  , :, <, >, (, ), ;, .

Name	Mandatory	Description	Field Type	Constraints
bindType	No	Determines Update certificate binding only or Create new site binding or update site binding (port, hostname, SNI)	String	<p>Applicable for locationType Certificate store</p> <p>Allowed values</p> <ul style="list-style-type: none"> <li>• Update certificate only</li> <li>• Update site binding</li> <li>• Create new site binding</li> </ul>
IsSNIEnabled	No	The Server Name Indication option to be mapped in IIS site binding	Boolean (true or false)	<p>Applicable when bindType is Update site binding or Create new site binding,</p> <p>Applicable for IIS versions &gt;= 8.0</p>
hostName	No	The hostname has to be updated in site binding	String	Applicable when bindType is Update site binding or Create new site binding
port	No	The port has to be updated in site binding	String	<p>Applicable when bindType is Update site binding or Create new site binding,</p> <p>Mandatory for bindType Create new site binding</p>

Name	Mandatory	Description	Field Type	Constraints
				Allowed values between 1-65535
pushRootAndIntermediateCertificates	Yes	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	

## Certificate details for Microsoft PC Server

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	The certificate format in which certificate has to be pushed	String	Allowed values • PKCS12-.pfx
certificateFileName	Yes	Certificate friendly name with which certificate to be pushed into Certificate store	String	The certificate file name should not start and end with special characters except @, #, \$, %, ^, &, -, {, }, [, ],  , :, <, >, (, ), ;, .
pfxPassword	Yes	The temporary password used during certificate push	String	The value has to be base64 encoded
pushRootAndIntermediateCertificates	Yes	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA

## Certificate details for Microsoft Server

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	The certificate format in which certificate has to be pushed	String	Allowed values <ul style="list-style-type: none"> <li>• PEM-.crt</li> <li>• PEM-.cer</li> <li>• PEM-.pem</li> <li>• DER-.cer</li> <li>• DER-.der</li> <li>• JKS-.jks</li> <li>• PKCS7-.p7b</li> <li>• PKCS7-.p7c</li> <li>• PKCS12-.p12</li> <li>• PKCS12-.pfx</li> </ul>
pushLocation	Yes	The folder location in which certificate and private key has to be pushed	String	NA
certificateFileName	Yes	The file name of certificate	String	Has to be suffixed with the format to be pushed. Ex: test.appviewx.com.pfx
privateKeyFileName	Yes	The file name of private key	String	Has to be suffixed with the format to be pushed. Ex: test.appviewx.com.key  Applicable for types PEM-.crt, PEM-.cer, PEM-.pem, DER-.cer, DER-.der, PKCS7-.p7b, PKCS7-.p7c

Name	Mandatory	Description	Field Type	Constraints
pfxPassword	No	The password of pfx/p12 file	String	The value has to be base64 encoded  Applicable for certificate types PKCS12-pfx,PKCS-p12
aliasName	No	The end entity certificate alias name for jks file	String	Applicable for certificate type JKS-.jks
keyStorePassword	No	The jks file password	String	The value has to be base64 encoded  Applicable for certificate type JKS-.jks
privateKeyPassword	No	The private key entry password for jks file	String	The value has to be base64 encoded  Applicable for certificateType JKS-.jks
pushRootAndIntermediateCertificates	Yes	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
rootCertificateFileName	Yes	Certificate name in which name the root certificate to be pushed	String	CA file name should not start and end with special characters. No special characters except !, @, #, \$, %, ^, &, _, {, }, [, ],  , :, <, >, (, ), ;, . and should not be the same

Name	Mandatory	Description	Field Type	Constraints
		in a firewall device.		as certificate file name and CA file name.
intermediateCertificateFileName	Yes	Certificate name in which name the intermediate certificates to be pushed in a firewall device.	List of Strings	Intermediate file names should not start and end with special characters. No special characters except !, @, #, \$, %, ^, &, _, {, }, [, ],  , :, <, >, (, ), ;, . and should not be the same as certificate file name and CA file name.

## Certificate details for Apache Server

Name	Mandatory	Description	Field Type	Constraints
isManualPush	No	If selected, user can customize certificate and key push location	Boolean (true or false)	NA
certificateType	Yes	Certificate type in which format the certificate to be pushed in a server device.	String	<b>Possible Values:</b> PEM-.crt PEM-.cer PEM-.pem
certificateLocation	No. Mandatory if "isManualPush" is true	A user defined directory	String	NA

Name	Mandatory	Description	Field Type	Constraints
		name in which the certificate will be pushed.		
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an Apache server.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
keyLocation	No. Mandatory if "isManualPush" is true	A user defined key directory name in which the keys will be pushed into an Apache server	String	NA
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an Apache server.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName

Name	Mandatory	Description	Field Type	Constraints
isRestartRequired	No	Option to restart the service after push	Boolean (true or false)	NA
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
trustLocation	No. Mandatory if “pushRootAndIntermediateCertificates” is true.	A user defined directory name in which the root and intermediate certificates will be pushed.	String	NA
rootCertificateFileName	No. Mandatory if “pushRootAndIntermediateCertificates” is true.	Certificate name in which name the root certificate to be pushed in a server device.	String	Intermediate file or bundle name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the

Name	Mandatory	Description	Field Type	Constraints
				certificate file name
intermediateCertificateFileName	No. Mandatory if “pushRootAndIntermediateCertificates” is true.	Certificate name in which name the intermediate certificates to be pushed in a server device.	List of Strings	Should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name.
privateKeyInDevice	No	If users need to push the private key in the device.	Boolean (true or false)	NA

## Certificate details for Tomcat Server

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Certificate type in which format the certificate to be pushed in a server device.	String	<b>Possible Values:</b> PEM-.crt PEM-.cer PEM-.pem JKS-.jks
certificateLocation	Yes	A user defined directory	String	NA

Name	Mandatory	Description	Field Type	Constraints
		name in which the certificate will be pushed.		
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an Apache server.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
keyLocation	Yes	A user defined key directory name in which the keys will be pushed into an Apache server	String	NA
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an Apache server.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName

Name	Mandatory	Description	Field Type	Constraints
keyStoreLocation	Yes	A user defined store location in which the keys and certificates will be pushed.	String	Applicable for certificate type JKS-.jks. Enter the keystore location with the extension as .jks or .keystore
keyStorePassword	Yes	The jks file password	String	Applicable for certificate type JKS-.jks
aliasName	No	The end entity certificate alias name for jks file	String	Applicable for certificate type JKS-.jks
privateKeyInDevice	No	If users need to push the private key in the device.	Boolean (true or false)	Applicable for certificate type JKS-.jks
privateKeyLocation	No. Mandatory if "privateKeyInDevice" is true.	A user defined directory name in which the private key will be pushed into the device.	String	Applicable for certificate type JKS-.jks

Name	Mandatory	Description	Field Type	Constraints
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	Applicable for all certificate types.
rootCertificateFileName	No. Mandatory if “pushRootAndIntermediateCertificates” is true.	Certificate name in which name the root certificate to be pushed in a server device.	String	Intermediate file or bundle name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name
intermediateCertificateFileName	No. Mandatory if “pushRootAndIntermediateCertificates” is true.	Certificate name in which name the intermediate certificates to be pushed in a server device.	List of Strings	Should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name.
trustStoreLocation	No. Mandatory if “pushRootAndIntermediateCertificates” is true	A user defined store location	String	Applicable for certificate type JKS-.jks. Provide the truststore

Name	Mandatory	Description	Field Type	Constraints
		in which the CA certificates will be pushed.		location with the extension as .jks or .truststore
trustStorePassword	No. Mandatory if "pushRootAndIntermediateCertificates" is true	The jks file password	String	Applicable for certificate type JKS-.jks

## Certificate details for Websphere Server

Name	Mandatory	Description	Field Type	Constraints
isManualPush	No	If selected, user can customize certificate and key push location	Boolean (true or false)	NA
certificateType	Yes	Certificate type in which format the certificate to be pushed in a server device.	String	<b>Possible Values:</b>  JKS-.jks  PKCS12-.p12
keyStoreLocation	Yes	A user defined store location in which the keys and certificates will be pushed.	String	Enter the keystore location with the extension as .jks or .keystore
keyStorePassword	Yes	The jks file password	String	Keystore password must have at least 6 characters.
trustStoreLocation	No	A user defined store location in which the CA certificates will be pushed.	String	Provide the truststore location with the extension as .jks or .truststore. Applicable only for certificate type JKS-.jks.

Name	Mandatory	Description	Field Type	Constraints
trustStorePassword	No	The jks file password	String	Truststore password must have at least 6 characters. Applicable only for certificate type JKS-.jks.
privateKeyInDevice	No	If users need to push the private key in the device.	Boolean (true or false)	NA
keyLocation	No	A user-defined key directory name in which the keys will be pushed into an Apache server	String	NA
aliasName	No	The end entity certificate alias name for jks file	String	NA

## Certificate details for Weblogic Server

Name	Mandatory	Description	Field Type	Constraints
isManualPush	No	If selected, user can customize certificate and key push location	Boolean (true or false)	NA
certificateType	Yes	Certificate type in which format the certificate to be pushed in a server device.	String	<b>Possible Values:</b> JKS-.jks
keyStoreLocation	Yes	A user defined store location in which the keys and certificates will be pushed.	String	Enter the keystore location with the extension as .jks or .keystore
keyStorePassword	Yes	The jks file password	String	Keystore password must have at least 6 characters.
trustStoreLocation	Yes	A user defined store location in which the CA certificates will be pushed.	String	Provide the truststore location with the extension as .jks or .truststore

Name	Mandatory	Description	Field Type	Constraints
trustStorePassword	Yes	The jks file password	String	Truststore password must have at least 6 characters.
privateKeyInDevice	No	If users need to push the private key in the device.	Boolean (true or false)	NA
keyLocation	No	A user defined key directory name in which the keys will be pushed into an Apache server	String	NA
aliasName	No	The end entity certificate alias name for jks file	String	NA

## Certificate details for IPlanet Server

Name	Mandatory	Description	Field Type	Constraints
isManualPush	No	If selected, user can customize certificate and key push location	Boolean (true or false)	NA
certificateType	Yes	Certificate type in which format the certificate to be pushed in a server device.	String	<b>Possible Values:</b> PKCS12-.pfx
dbPath	Yes	Database Location in server where certificates will be pushed.	String	NA
dbPassword	No	Password required to import certificates to database in server.	String	NA
isRestartRequired	No	Option to restart the service after push	Boolean (true or false)	NA

Name	Mandatory	Description	Field Type	Constraints
privateKeyInDevice	No	If users need to push the private key in the device.	Boolean (true or false)	NA
keyLocation	No	A user defined key directory name in which the keys will be pushed into an Apache server	String	NA

## pushDetails

Name	Mandatory	Description	Field Type	Constraints
preValidationScriptPath	No	Location of the pre push script.	String	NA
postValidationScriptPath	No	Location of the post push script.	String	NA
overwrite	No	Whether the certificate needs to be overwritten if already available.	Boolean (true or false)	Enabled by default for IIS vendor.
pushAutomatically	No	Whether the certificate needs to be pushed automatically when renewed.	Boolean (true or false)	

# Chapter 126: Response Structure

202 Accepted returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the push request	List of response
message	Success message of the action or failure description in case of error	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	

response

Name	Description	Field Type
requestId	Request Id for push action for the application connector	String
connectorId	Application connector Id	String

## Chapter 127: Status Codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	-	1 connector(s) saved and push operation has been triggered.	-
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>.	Check and ensure that valid value is provided for <<field name>> field in the request.
417 Expectation failed	FIELD_VALUE_INVALID	Invalid value - <<field name>>.	Check and ensure that valid value is provided for the <<field name>> field in the request.
404 Not Found	NO_RECORDS_FOUND	No matching records found - certificate not found.	Please provide correct value for the field <code>certificateId</code> .
400 Bad Request	INVALID_REQUEST	<code>selectedProfiles</code> are already available in the specified certificate.	Provide different value for the <code>selectedProfiles</code> field.
417 Expectation failed	CERT-APP-0016	Connector with profiles {} already exists.	Profile connector already available for the selected certificate. Please change the <code>certificateId</code> or delete the existing connector.
500	avx-common-011	Error while processing.	

HTTP Status code	appStatusCode	Message	Possible remediation
Internal Server Error			

# Chapter 128: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)
- [Sample Request/Response For Apache Server](#)
- [Sample Request/Response for IPlanet Server](#)
- [Sample Request/Response for Tomcat Server](#)
- [Sample Request/Response for Weblogic Server](#)
- [Sample Request/Response for Websphere Server](#)

## Sample Request:

### Sample Request

```
{
  "certificateDetails": {
    "certificateType": "PEM-.pem",
    "certificateName": "testfile",
    "pushRootAndIntermediateCertificates": true,
    "rootCertificateFileName": "testrootfile.pem",
    "intermediateCertificateFileName": [
      "testinterfilelevel1.pem"
    ]
  },
  "pushDetails": {
    "preValidationScriptPath": "",
    "postValidationScriptPath": "",
    "overwrite": false,
    "pushAutomatically": false
  },
  "certificateId": "5f5b1fac77534426423bab57",
  "selectedProfiles": [
    "xxx.xxx.xx.xxx:@Template_001:@shared:@IKE Gateway Profile:@TEST"
  ]
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{
  "response": [
    {
      "requestId": "3",
      "connectorId": "xxx.xxx.xx.xxx:@Template_001:@shared:@IKE Gateway Profile:@TEST:@389ed6f96387e3002cc6ac0678c07ffe70459abf"
    }
  ],
  "message": "1 connector(s) saved and push operation has been triggered.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

## Sample Request/Response For Apache Server

### Sample Request:

```
{
  "certificateDetails": {
    "certificateType": "PEM-.pem",
    "certificateLocation": "/opt/automation",
    "certificateFileName": "apachecertkeytest",
    "keyLocation": "/opt/automation",
    "privateKeyFileName": "apachecertkeytest.key",
    "isRestartRequired": true,
    "pushRootAndIntermediateCertificates": true,
    "trustLocation": "/opt/automation",
    "rootCertificateFileName": "apacheroottest.pem",
    "intermediateCertificateFileName": [
      "apacheintermediatetest.pem"
    ],
    "privateKeyInDevice": true
  }
}
```

```

},
"certificateId": "605dbd66ccb1c49e108e67e6",
"selectedProfiles": [
"ApacheLinux:@/etc/apache2:@_default_:@443:@/etc/apache2/sites-available/default-ssl.conf"
]
}

```

## Sample Response

```

{
"response":[
{
"requestId":"54",
"connectorId":"ApacheLinux:@/etc/apache2:@_default_:@443:@/etc/apache2/sites-available/default-ssl.conf:@51fb6d24534926ea1d89d32bfe59b25c26d84b5e"
}
],
"message":"1 connector(s) saved and push operation has been triggered.",
"appStatusCode":null,
"tags":{
},
"headers":null
}

```



**Note:** Above given sample request and response can be followed for all three certificate types (PEM-.crt, PEM-.cer, PEM-.pem) for the Apache server. **certificateId** in the above request can be found using the search API using commonName, serial No. or other search parameters. **resourceId** present in Search API response is equivalent to **certificateId** here.

## Sample Request/Response for IPlanet Server

### Sample Request

```

{
"certificateDetails": {
"isManualPush":true,
"certificateType": "PKCS12-.pfx",
"dbPath": "/opt/automation/db",

```

```

"dbPassword": "YXBwd==26",
"isRestartRequired":false,
"privateKeyInDevice": true,
"privateKeyLocation":"/opt/automation/keys",
},
"certificateId": "605dbd66ccb1c49e108e67e6",
"selectedProfiles": [
"iPlanet:@8443:@localhost:@localhost:"
]
}

```

## Sample Response

```

{
"response":[
{
"requestId":"60",
"connectorId":"iPlanet:@8443:@localhost:@localhost:@51fb6d24534926ea1d89d32bfe59b25c26d84b5e"
}
],
"message":"1 connector(s) saved and push operation has been triggered.",
"appStatusCode":null,
"tags":{
},
"headers":null
}

```



**Note:** Above given sample request and response can be followed for PKCS12 certificate type (PKCS12-.pfx) for the IPlanet server. **certificateId** in the above request can be found using the search API using commonName, serial No. or other search parameters. **resourceId** present in Search API response is equivalent to **certificateId** here.

## Sample Request/Response for Tomcat Server

### Sample Request

```

{
"certificateDetails": {

```

```

"certificateType": "PEM-.pem",
"certificateLocation": "/opt/automation",
"certificateFileName": "tomcatcertkeytest",
"keyLocation": "/opt/automation",
"privateKeyFileName": "tomcatcertkeytest.key",
"pushRootAndIntermediateCertificates": true,
"rootCertificateFileName": "tomcatroottest.pem",
"intermediateCertificateFileName": [
"tomcatintermediatetest.pem"
]
},
"certificateId": "605dbd66ccb1c49e108e67e6",
"selectedProfiles": [
"TomcatLinux:@8443:@localhost:@localhost:"
]
}

```

## Sample Response

```

{
"response":[
{
"requestId":"57",
"connectorId":"TomcatLinux:@8443:@localhost:@localhost:@51fb6d24534926ea1d89d32bfe59b25c26d84b5e"
}
],
"message":"1 connector(s) saved and push operation has been triggered.",
"appStatusCode":null,
"tags":{
},
"headers":null
}

```



**Note:** Above given sample request and response can be followed for all three certificate types (PEM-.crt, PEM-.cer, PEM-.pem) for the Tomcat server. **certificateId** in the above request can be found using the search API using commonName, serial No. or other search parameters. **resourceId** present in Search API response is equivalent to **certificateId** here.

# Sample Request/Response for Weblogic Server

## Sample Request

```
{
  "certificateDetails": {
    "isManualPush":true,
    "certificateType": "JKS-.jks",
    "keyStoreLocation": "/opt/automation/key.jks",
    "keyStorePassword": "YXBwd==26",
    "trustStoreLocation": "/opt/automation/trust.jks",
    "trustStorePassword": "YXBwd==26"
    "privateKeyInDevice": true,
    "privateKeyLocation":"/opt/automation/keys",
    "aliasName": "sampleKeystore"
  },
  "certificateId": "605dbd66ccb1c49e108e67e6",
  "selectedProfiles": [
    "Weblogic:@8443:@localhost:@localhost:"
  ]
}
```

## Sample Response

```
{
  "response":[
    {
      "requestId":"58",
      "connectorId":"Weblogic:@8443:@localhost:@localhost:@51fb6d24534926ea1d89d32bfe59b25c26d84b5e"
    }
  ],
  "message":"1 connector(s) saved and push operation has been triggered.",
  "appStatusCode":null,
  "tags":{
  },
  "headers":null
}
```



**Note:** Above given sample request and response can be followed for JKS certificate type (JKS.jks) for the Weblogic server. **certificateId** in the above request can be found using the search API using commonName, serial No. or other search parameters. **resourceId** present in Search API response is equivalent to **certificateId** here.

## Sample Request/Response for Websphere Server

### Sample Request

```
{
  "certificateDetails": {
    "isManualPush": true,
    "certificateType": "JKS-jks",
    "keyStoreLocation": "/opt/automation/key.jks",
    "keyStorePassword": "YXBwd==26",
    "trustStoreLocation": "/opt/automation/trust.jks",
    "trustStorePassword": "YXBwd==26",
    "privateKeyInDevice": true,
    "privateKeyLocation": "/opt/automation/keys",
    "aliasName": "sampleKeystore"
  },
  "certificateId": "605dbd66ccb1c49e108e67e6",
  "selectedProfiles": [
    "WebsphereLinux:@8443:@localhost:@localhost:"
  ]
}
```

### Sample Response

```
{
  "response": [
    {
      "requestId": "58",
      "connectorId": "WebsphereLinux:@8443:@localhost:@localhost:@51fb6d24534926ea1d89d32bfe59b25c26d84b5e"
    }
  ],
  "message": "1 connector(s) saved and push operation has been triggered.",
  "appStatusCode": null,
}
```

```
"tags":{
},
"headers":null
}
```



**Note:** Above given sample request and response can be followed for JKS certificate type (JKS.jks) for the Websphere server. **certificateId** in the above request can be found using the search API using commonName, serial No. or other search parameters. **resourceId** present in Search API response is equivalent to **certificateId** here.

## Sample Request

```
{
"certificateDetails": {
"isManualPush":true,
"certificateType": "PKCS12-.p12",
"keyStoreLocation": "/opt/automation/key.jks",
"keyStorePassword": "YXBwd==26",
"privateKeyInDevice": true,
"privateKeyLocation":"/opt/automation/keys",
"aliasName": "sampleKeystore"
},
"pushDetails": {
"preValidationScriptPath": "",
"postValidationScriptPath": "",
"pushAutomatically": false
},
"certificateId": "605dbd66ccb1c49e108e67e6",
"selectedProfiles": [
"WebsphereLinux:@8443:@localhost:@localhost:"
]
}
```

## Sample Response

```
{
"response":[
{
"requestId":"59",
```

```
"connectorId":"WebsphereLinux:@8443:@localhost:@localhost:@51fb6d24534926ea1d89d32bfe59b25c26d84b5e"
}
],
"message":"1 connector(s) saved and push operation has been triggered.",
"appStatusCode":null,
"tags":{
},
"headers":null
}
```



**Note:** Above given sample request and response can be followed for PKCS12 certificate type (PKCS12-.p12) for the Websphere server. **certificateld** in the above request can be found using the search API using commonName, serial No. or other search parameters. **resourceId** present in Search API response is equivalent to **certificateld** here.

## Chapter 129: Overview

The API will rollback the certificate and it's private keys to the server device and associate that to the server profiles. Please refer to [After you are done](#) section to Approve and Implement the request.

# Chapter 130: Before you begin

Before attempting to rollback a certificate to a particular Server device through AppViewX, the user has to ensure the following:

- Server devices should be added in AppViewX.
- The device should be in Managed state.
- **Approval is not required:** Users can enable this mode by setting the 'Certificate Requests Need Approval?' flag to false in the Certificate Policy.
- **Approval is required:** If the approval setting in the policy cannot be changed, users can approve specific requests by following the [After you are done](#) section.

# Chapter 131: Request Structure

URL: /certificate/rollback

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'.
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsources	Query	Source from which the request is triggered(E.g. external)	String	NA
autoApproval	Query	Auto approval needed for the action or not.	String	Value should be <b>yes</b> .
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
applicationConnectorIds	Yes	Application connector id.	String	NA

# Chapter 132: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the rollback request.	List of response
message	Success message of the action or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
requestId	Request Id for rollback action for the application connector.	String
connectorId	Application connector Id.	String

## Chapter 133: Status Codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	App connector rollback action initiated for 1 connector(s).	NA
202 Accepted	NA	Operation cannot be completed for one or more devices as the 'Resource' allocated to you does not have write permission.	User does not have access to the device.
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>.	Check and ensure that valid value is provided for <<field name>> field in the request.
417 Expectation failed	FIELD_VALUE_INVALID	Invalid value - <<field name>>.	Check and ensure that valid value is provided for the

HTTP Status code	appStatusCode	Message	Possible remediation
			<<field name>> field in the request.
417 Expectation failed	CERT-APP-0012	Application connector ids cannot be empty.	Please provide value for the field <code>applicationConnectorIds</code> .
417 Expectation failed	ERR_APPLICATION_CONNECTOR_LIST_RETRIVAL	Unable to retrieve connector information.	Connector may not be available. Please provide correct value for the field <code>applicationConnectorIds</code> .
417 Expectation failed	ERR_APP_CONNECTORS_NOT_FOUND	Application connector(s) not found.	Connector may not be available. Please provide correct value for the field <code>applicationConnectorIds</code> .
417 Expectation failed	ERR_INITIALIZE_ROLLBACK_REQUEST	Unable to initialize rollback request.	NA
500 Internal Server Error	avx-common-011	Error while processing.	NA

## Chapter 134: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{  
  "applicationConnectorIds": [ "xx.xxx:@clientssl-insecure-compatible:@Common:@c46ec8a04da701721159ce0c3cf772367ade58cb"  
]  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

### Sample Response:

```
{  
  "response": "Success",  
  "message": "App connector rollback action initiated for 1 connector(s).",  
  "appStatusCode": null,  
  "tags": null,  
  "headers": null  
}
```

## Chapter 135: Overview

- Once an action has been triggered, a request will be created for that.
- Requests have to be approved using the [Approve request](#) action.
- Requests have to be approved using the [Implement request](#) action.

# Chapter 136: Approve a Request

- Request Structure
- Response Structure
- Status codes
- Sample Request/Response

## Request Structure

URL: */certificate/workorder/update*

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b> .
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
data	Yes	Action data.	Data	NA
header	Yes	Request details.	Header	NA

## Data

Name	Mandatory	Description	Field Type	Constraints
task_action	Yes	Action to be performed.	Number	1 - approve 2 - reject

## Header

Name	Mandatory	Description	Field Type	Constraints
request_id	Yes	Id of the request.	String	NA
workorder_id	Yes	Id of the work order.	String	Value should be <b>0</b> .
task_id	Yes	Id of the task to be performed.	String	Value should be <b>review_1</b> .

## Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the approve request.	response
message	Success message of the action or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
requestId	Request Id for approve action for the application connector.	String
message	Success message of the action or failure description in case of error.	String

## Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
200 OK	NA	Request submitted to workflow engine for processing workflow request 14.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
404 Not Found	engine-db-011	Request not found.	Check and ensure that valid value is provided for request_id field in the request.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "data": {
    "task_action": 1
  },
  "header": {
    "request_id": "14",
    "workorder_id": "0",
    "task_id": "review_1"
  }
}
```

```
}  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{  
  "response": {  
    "message": "Request submitted to workflow engine for processing workflow request 14",  
    "requestId": "14"  
  },  
  "message": "Success",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

# Chapter 137: Implement Request

- Request Structure
- Response Structure
- Status codes
- Sample Request/Response

## Request Structure

URL: */certificate/workorder/update*

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if <code>username</code> and <code>password</code> are not provided.
username	Header	AppViewX login username.	String	Required if <code>sessionId</code> is not provided.
password	Header	AppViewX login password.	String	Required if <code>sessionId</code> is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b> .
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	

Payload

Name	Mandatory	Description	Field Type	Constraints
data	Yes	Action data.	Data	
header	Yes	Request details.	Header	

## Data

Name	Mandatory	Description	Field Type	Constraints
task_action	Yes	Action to be performed.	Number	1 - implement 2 - reject

## Header

Name	Mandatory	Description	Field Type	Constraints
request_id	Yes	Id of the request.	String	
workorder_id	Yes	Id of the work order.	String	Value should be <b>0</b> .
task_id	Yes	Id of the task to be performed.	String	Value should be <b>review_1</b> .

## Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the implement request.	response
message	Success message of the action or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	

## response

Name	Description	Field Type
requestId	Request Id for implement action for the application connector.	String
message	Success message of the action or failure description in case of error.	String

## Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
200 OK	-	Request submitted to workflow engine for processing workflow request 14.	-
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid username and password or valid sessionId is provided as header param.
404 Not Found	engine-db-011	Request not found.	Check and ensure that valid value is provided for request_id field in the request.

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

### Sample Request:

```
{
  "data": {
    "task_action": 1
  },
  "header": {
    "request_id": "14",
    "workorder_id": "0",
    "task_id": "review_2"
  }
}
```

```
}  
}
```



**Note:** Please refer to the "Request Structure" to identify the changeable values.

## Sample Response:

```
{  
  "response": {  
    "message": "Request submitted to workflow engine for processing workflow request 14",  
    "requestId": "14"  
  },  
  "message": "Success",  
  "appStatusCode": null,  
  "tags": {},  
  "headers": null  
}
```

## Chapter 138: Overview

The API will download the private key of a certificate in password protected PEM format. The user can request for private key.

# Chapter 139: Request Structure

URL: /certificate/privatekey/download

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
uuid	Yes	Format required.	String	Uuid of the certificate
password	No	Password for encrypting the private key in the response.	String	NA

# Chapter 140: Response Structure

## Success Response

```
{
  "response": {
    "status": "Success",
    "zipFileName": null,
    "certContents": null,
    "privateKeyPemEncoded": "-----BEGIN ENCRYPTED PRIVATE
KEY-----\nMIIE9jAoBgoqhkiG9w0BDAEDMBoEFEC4EngLMd1kbUJjHxgNyvApBAMFAgIIAASC\nBMhv4AkofevTbEa0BZ3HLOI/WIHelmYmHtwC26SDI1JGJWK
UN1O0W52rIn30mjZG\niUxiu7F3ozQgrKBCHuM+IKxx4zYbmgukyodJtbzJ4EohmapiZQHJSf4DiJiHHa8X\nZ/r
+sfNoRunhy3r/0SS0INzQvnm5dqSATkHTtrcD82NsKlx4GScFtG5yKhoAsS6nEJQag7THPAbwaRPMl54UZBsFnmkG638utC511EWdbD0Z12L4zxhzPNxgnYF
NQEg\n12552kj5jGNZgojxZ52BpWe0ZfDKH+bvt7LC9M62n4d5vrvivfBd6R7ZMJDAD0M\nnn2vlttgoKe9Nbxq/5oraGJt+8q/sWIGWehx/L8Ue/weCzXtmectQd69M4id
K7QQ\nAwOh/cdosut6uQUGzlrH3n59INWbkJ0HzlunJjLcvJzrhr191SYmMU0HNPjaVptD\n
+UErWYqs4y1XEwOn1jrNQ8cBA0VvkwHau1juWc0hoY9s6tv573dnjrnAtiVotw4\nl3CsQINMAXSHizpaM6pKXh8mCHfgZqzDmbmDTTDc9A37ekgnDfSRCEgzR
KQl4bJT\noXYRJFTyjGqT4HdC9pKaj3B+IHfWae5iWOB5kCdaE9qCc9NjN8WzPed3U3ltNNzC\nneW3mYihP+
+XCB9uNHylZ9jCXh42RCCW/omVlgZR9VBvhrq3tmk8MCYp+cFdo2KsH\nnC+MAVL6NPI4jcGBNt2JUMG5coPXttF2OsHqucH95WqUDUBF5+ZmpvJSuCFb8
SEc\nvAH35lwJnanZs4hPlIgywkRcldu+Mv
+lseOqFQvRypeRKPLGIWqZQZ83J5P2ENh\nnhQLv9hrYLKzpo4oMk4GTFelNvYnayAa5sqAOACoI8QZRmjUPVffG3KEQ6F3vOTdX\n1BrpeSH6bolqEUTcBpc
agf+1XdoZ4f3h76K0yVH1KVE8Gz2uVqwm33UThVU2ra\nn0kWLxO2tz3tN0m21MLQhU4bWDMT5Yz5GdsLLQqkizZ7vwfnYT6xtTPTJLre0X+
\n927vNjnhVgtdydtSImf+m
+tH3knZ35K8MCBH7ILV+Ov0dj4DSbd/44eMuBOP7KW\nu8u4FK0Mys4EeBc2nH9tQ/HTSDCRJGj8syncRHZGGCRRaUfIBXy0F9InbaoTCNnI\nSngeW/ZjuSBHH0
csNxxJZS3rr62Pnl14IOueM5cklLa02LqON/gdRZwXQm7i0sOTEC\nnvhX4Y3RgJ8+IZYx3MyeNkuUBFOuyRcl3my/C6K9okMyQhAgXr16YR/v\nioKihX8/\nRH0RC9
OrSy6ad0ILRhye9/giEJ2JBfkq8QwwRUO1peD7lrzde6gD8c9XRmbshgu\nndU00dmL0a26H/h
+/T9UNs5D1KJUROHxeCX2ywakw/k1hR9xfiQsYJ6FVXGC38POW\nVnOK4gmRnGQPfDsQZJ32bm+cyqQ2nmXYHxGQO1S0k+U5JlmPk9EruaWal2rud8wO\nnd
AcT/ZUxaBXXjwRnOAr142Hh939KuMcb7IK7as0Ku2xoBxLSGFikaQXC\nnLVavs1C\nnGns0ZrN9rDwua7e7VTug08oI0H0Fvqt5cgQAaFTDjM6Usy7AoKJVOP4j4Us
W78m\nTRAnJLlfdxCB5b315bMMg5uTpZRDclybY0=\n-----END ENCRYPTED PRIVATE KEY-----\n",
    "success": true,
    "log": false
  },
  "message": "private key downloaded successfully",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

Response content-type - application/json.

## Error Response

Error response will have following params:

Name	Description	Field Type
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
privateKeyPemEncoded	Private key of certificate in password encrypted pem.	String
success	True-if download is successful.  False-if error in downloading.	Boolean (true or false)
tags	More info in case of failure response	NA

## Chapter 141: Status Codes

HTTP Status code	appStatusCode	Message	Possible remediation
417 Expectation Failed	CERT-GEN-0015	Field cannot be null/empty	Please verify if all the mandatory fields have valid value in the request.it should not be null or empty.
417 Expectation Failed	CERT-KEY-0006	Private key access is restricted in <policy> policy	Please enable private key access in the mentioned policy
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid	Please verify if all the mandatory fields have valid value in the request.
500 Internal Server Error	CERT-GEN-0027	certificate private key download failed.	Internal Server Error.

# Chapter 142: Overview

The API will delete the certificates using UUID.

Universal Unique Identifier(UUID) is a unique id for each certificate. This id remains the same even if we generate the certificate numerous times. It is generated on the basis of certain attributes and properties of the certificate.

# Chapter 143: Request Structure

URL: <https://192.168.150.63:31443/avxapi/certificate/delete?>

[gwsource=external&commonName={commonName}&serialNumber={serialNumber}](https://192.168.150.63:31443/avxapi/certificate/delete?gwsource=external&commonName={commonName}&serialNumber={serialNumber})

Type: DELETE

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b>
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
certIds	Yes	Format required.	Array	uuid of the certificates to be deleted

# Chapter 144: Response Structure

## Success Response

```
{
  {
    "response": {
      "code": "CERTIFICATE_DELETION_SUCCESSFULL",
      "message": "Selected certificate(s) has been deleted successfully from AppViewX inventory."
    },
    "message": null,
    "appStatusCode": null,
    "tags": {},
    "headers": null
  }
}
```

Response content-type - application/json.

## Error Response

Error response will have the following params:

Name	Description	Field Type
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

## Chapter 145: Status Codes

HTTP Status code	appStatusCode	Message	Possible remediation
417 Expectation Failed	CERT-CA-0047	Cert Ids cannot be empty	Please provide UUIDs in the request to delete certificates
404 Not found	CERT-INV-0021	Requested certificate is not available in the inventory.	Provide uuids that are available in the inventory.
417 Expectation Failed	CERT-HLST-0019	The operation cannot be completed as {} selected certificate(s) are in Read-Only access groups	Please change the permission the required group
500 Internal Server Error	CERT-GEN-0028	Certificate Limit Exceed.	Cannot delete more than 100 certificates.

# Chapter 146: Push and Bind Certificate to a ADC Profile

- [Before You Begin](#)
- [Fetch available profiles API](#)
- [Request Structure](#)
- [Response for Push and Bind ASYNC API](#)

## Before You Begin

Before attempting to push a certificate to a particular ADC device through AppViewX, the user has to ensure the following:

- ADC devices should be added in AppViewX.
- The device should be in Managed state.

## Fetch available profiles API

HTTP Method: GET

URL: */certificate/profiles*

Query Parameters : category, vendor, deviceName

### Request Details

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided

Name	Type	Description	Field Type	Constraints
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b>
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external).	String	<b>Possible values:</b> WEB, external
category	Query	Specifies the device category. It is a mandatory field.	String	<b>Possible values:</b> ADC, Server, Firewall
vendor	Query	Vendor for the chosen device. For example, <b>F5</b> is a vendor for the ADC category. It is a mandatory field.	String	<b>Possible values:</b> F5, Citrix, AVI, A10, AmazonELB, NginxPlus, HAProxy
certificateUuid	Query	Resource id of the certificate	String	This can be obtained from search api
deviceName	Query	Name of the device as per AppViewX Device Inventory	String	NA
inventory	Query	Name of AppViewX inventory where certificate is present	String	<b>Possible values:</b> Server, client, code signing, device

#### Sample Request

```
https://<appviewx node ip>:<gateway port>/avxapi/certificate/profiles?
gwkey=f000ca01&category=ADC&vendor=F5&deviceName=F5_device&gwsource=external
```

## Response Details

Name	Description	Field Type
response	Contains the mentioned response attributes for the fetch profiles request <ul style="list-style-type: none"> <li>• objects</li> <li>• totalRecords</li> <li>• obtainedRecords</li> <li>• obtainedRecordRange</li> </ul>	List of response
objects	List of available device profile Ids.	List of String
totalRecords	Total number of profiles fetched.	Integer
obtainedRecords	Total number of profiles fetched.	Integer
obtainedRecordRange	Range of record found.	Object
message	Success message of the action or failure description in case of error	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

## Sample Response

```
{
  "response": {
    "objects": [
      "device_name",
      "device_name:@apm-default-serverssl:@Common",
      "device_name:@clientssl-insecure-compatible:@Common",
      "device_name:@clientssl-secure:@Common",
      "device_name:@clientssl:@Common",
      "device_name:@crypto-client-default-serverssl:@Common",
      "device_name:@crypto-server-default-clientssl:@Common",
      "device_name:@pcoip-default-serverssl:@Common",
    ]
  }
}
```

```

"device_name:@serverssl-insecure-compatible:@Common",
"device_name:@serverssl:@Common",
"device_name:@splitsession-default-clientsl:@Common",
"device_name:@splitsession-default-serverssl:@Common",
"device_name:@wom-default-clientsl:@Common",
"device_name:@wom-default-serverssl:@Common"
],
"totalRecords": 14,
"obtainedRecords": 14,
"obtainedRecordRange": {
"start": 0,
"end": 0
}
},
"message": "14 record(s) found",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## Request Structure

**URL:** /certificate/pushToDevice

**Type:** POST

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided

Name	Type	Description	Field Type	Constraints
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be <b>'application/json'</b>
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

## Payload

Name	Mandatory	Description	Field Type	Constraints
certificateId	Yes	Resource id of the certificate	String	NA
certificateUuid	Yes	Certificate UUID	String	Mandatory if certificateId is not present
selectedProfiles	Yes	ADC device profile id	String	<ul style="list-style-type: none"> <li>If Profile connector specified then certificate will be pushed and bound to the profile</li> <li>If Default connector specified then certificate will be only pushed</li> </ul> <p>These can be obtained from <a href="#">Fetch Available Profiles API</a></p>
certificateDetails	Yes	Certificate details for the Server devices	Any of the following -	

Name	Mandatory	Description	Field Type	Constraints
			<ul style="list-style-type: none"> <li>• <a href="#">Certificate details for F5</a></li> <li>• <a href="#">Certificate details for Citrix</a></li> <li>• <a href="#">Certificate details for A10</a></li> <li>• <a href="#">Certificate details for AVI</a></li> <li>• <a href="#">Certificate details for AmazonELB</a></li> <li>• <a href="#">Certificate details for NginxPlus</a></li> <li>• <a href="#">Certificate details for HAProxy</a></li> </ul>	

Certificate details for F5

Name	Mandatory	Description	Field Type	Constraints
<a href="#">certificateType</a>	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt
<a href="#">certificateFileName</a>	Yes	A user defined file name in which the certificate will be	String	The certificate file name should not start and end with special characters except -, ., _. It should not be

Push and Bind Certificate to a ADC Profile

Name	Mandatory	Description	Field Type	Constraints
		pushed into an F5 device.		the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an F5 device.	String	The key file name should not start and end with special characters except -, ., _ . Key file name should be same as certificateFileName
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
rootCertificateFileName	<ul style="list-style-type: none"> <li>• No.</li> <li>• Yes,if “pushRootAndIntermediateCertificates” is true.</li> </ul>	Certificate name in which name the intermediate and root certificates to be pushed in a F5 device.	String	Intermediate file or bundle name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name

Sample Request

```
{
  "certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles": [
    "device_name:@default:@clientsl"
  ],
  "certificateDetails": {
    "certificateFileName": "test123.crt",
    "certificateType": "PEM-.crt",
    "privateKeyFileName": "test123.key",
    "pushRootAndIntermediateCertificates": true,
    "rootCertificateFileName": "test123_root.crt"
  }
}
```

### Sample Response

```
{
  "response": [
    {
      "requestId": "156",
      "connectorId": "device_name:@default:@clientsl:@9eb94a53963d1ae326dbf4cda6077f55baa8476e"
    }
  ],
  "message": "1 connector(s) saved and push operation has been triggered.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

### Certificate details for Citrix

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt,

Push and Bind Certificate to a ADC Profile

Name	Mandatory	Description	Field Type	Constraints
				PEM-.cer, PEM-.pem, DER-.der, DER-.cer
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an F5 device.	String	The certificate file name should not start and end with special characters except -, ., _ . It should not be the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an F5 device.	String	The key file name should not start and end with special characters except -, ., _ . Key file name should be same as certificateFileName
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA

Name	Mandatory	Description	Field Type	Constraints
intermediateCertificateFileName	Yes,if "pushRootAndIntermediateCertificates" is true.	Certificate name in which name the intermediate certificates to be pushed in a citrix device.	List<String>	Intermediate file name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name
rootCertificateFileName	Yes,if "pushRootAndIntermediateCertificates" is true.	Certificate name in which name the root certificates to be pushed in a citrix device.	String	Root file name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name

Sample Request

```
{
  "certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles": [
    "device_name:@default:@sslvs"
  ],
  "certificateDetails": {
    "certificateFileName": "test123.crt",
    "certificateKeyPairName": "test123",
    "certificateType": "PEM-.crt",
```

```

"intermediateCertificateFileName": [
  "test123_inter.crt"
],
"privateKeyFileName": "test123.key",
"pushRootAndIntermediateCertificates": true,
"rootCertificateFileName": "test123_root.crt",
"sniCert": "false",
"sniStatus": "false"
}
}
    
```

### Sample Response

```

{
  "response": [
    {
      "requestId": "156",
      "connectorId": "device_name:@default:@clntssl:@9eb94a53963d1ae326dbf4cda6077f55baa8476e"
    }
  ],
  "message": "1 connector(s) saved and push operation has been triggered.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
    
```

### Certificate details for AVI

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.pem
certificateFileName	Yes	A user defined file name in which the	String	The certificate file name should not start and end with special characters

Push and Bind Certificate to a ADC Profile

Name	Mandatory	Description	Field Type	Constraints
		certificate will be pushed into an AVI device.		except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an AVI device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
intermediateCertificateFileNames	Yes of pushRootAndIntermediateCertificates is set true	Intermediate file name to be pushed into device	List<String>	NA
rootCertificateFileName	Yes of pushRootAndIntermediateCertificates is set true	Root file name to be pushed into device	String	NA

Sample Request

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",

```

```

"selectedProfiles":[
"device_name:@serverssl-insecure-compatible:@Common"
],
"certificateDetails":{
"certificateType":"PEM-.pem",
"certificateFileName":"certificate_push.pem",
"privateKeyFileName":"certificate_push.key",
"pushRootAndIntermediateCertificates": true,
"intermediateCertificateFileNames":["test_cert_inter.crt"],
"rootCertificateFileName": "test_cert_ca.crt"
}
}

```

### Sample Response

```

{
"response": [
{
"requestId": "156",
"connectorId": "device_name:@default:@clntssl:@9eb94a53963d1ae326dbf4cda6077f55baa8476e"
}
],
"message": "1 connector(s) saved and push operation has been triggered.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

### Certificate details for A10

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt, PEM-.pem,

Push and Bind Certificate to a ADC Profile

Name	Mandatory	Description	Field Type	Constraints
				PEM-.cer, DER-.der, DER-.cer, PKCS#7-.p7b, PKCS#7-.p7c, PKCS#7-.p12, PKCS#7-.pfx
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an A10 device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an A10 device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
pfxPassword	Yes if pfx and pkcs12 type are selected	A password for the pfx type of files	String	This should be Base64 encoded

Name	Mandatory	Description	Field Type	Constraints
pushRootAndIntermediateCertificates	<ul style="list-style-type: none"> <li>No</li> </ul>	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
intermediateCertificateFileNames	Yes of pushRootAndIntermediateCertificates is set true	Intermediate file name to be pushed into device	List<String>	NA

Sample Request for non pfx and p12 cert types

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
    "certificateType":"PEM-.crt",
    "certificateFileName":"certificate_push.crt",
    "privateKeyFileName":"certificate_push.key",
    "pushRootAndIntermediateCertificates": true,
    "intermediateCertificateFileNames":["test_cert_ca.crt"]
  }
}
```

Sample Request for pfx and p12 cert types

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
```

```

"certificateType": "PKCS-.pfx",
"certificateFileName": "certificate_push.pfx",
"pushRootAndIntermediateCertificates": true,
"intermediateCertificateFileNames": [ "test_cert_ca.crt" ],
"pfxPassword": "cGFzc3dvcmQ="
}
}

```

### Sample Response

```

{
"response": [
{
"requestId": "156",
"connectorId": "device_name:@default:@clntssl:@9eb94a53963d1ae326bf4cda6077f55baa8476e"
}
],
"message": "1 connector(s) saved and push opertaion has been triggered.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

### Certificate details for AmazonELB

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.pem
certificateLocation	Yes	<ul style="list-style-type: none"> <li>• ACM</li> <li>• IAM</li> </ul> Select one of these to input the certificate location	String	The key file name should not start and end with special characters except -, ., _ . Key file name should be same as certificateFileName

Name	Mandatory	Description	Field Type	Constraints
certCARReferenceld	String	This is applicable if ACM is selected.  Reference ID to input the certificate location	String	No special characters except ('=', '/', ',', '!', '@', '-') are allowed.
certificateFileName	Yes if certificate location selected is IAM	This is mandatory if IAM is selected. A user defined file name in which the certificate will be pushed into an AmaonELB device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
pushRootAndIntermediateCertificates	• No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
certificateLocation	Yes	<ul style="list-style-type: none"> <li>• ACM</li> <li>• IAM</li> </ul> Select one of these to input the certificate location	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName

## Sample request if certificate location is ACM

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
    "certCARReferenceld": "arn:aws:iam::2313:amazonELB"
    "certificateLocation": "ACM"
  }
}
```

```

"certificateType": "PEM-.pem"

"pushRootAndIntermediateCertificates": true

}
    
```

Sample request if certificate location is IAM

```

{

"certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",

"selectedProfiles": [

"device_name: @serverssl-insecure-compatible: @Common"

],

"certificateDetails": {

"certificateFileName": "sample.pem",

"certificateLocation": "IAM",

"certificateType": "PEM-.pem",

"pushRootAndIntermediateCertificates": true

}

}
    
```

Sample Response

```

{

"response": [

{

"requestId": "156",

"connectorId": "device_name: @default: @clntssl: @9eb94a53963d1ae326dbf4cda6077f55baa8476e"

}

],

"message": "1 connector(s) saved and push operation has been triggered.",

"appStatusCode": null,

"tags": {},

"headers": null

}
    
```

Certificate details for NginxPlus

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt,

Name	Mandatory	Description	Field Type	Constraints
				PEM-.cer, PEM-.pem
certificateLocation	Yes	A user defined file name in which the certificate will be pushed into an NginxPlus device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
keyLocation	Yes	A user defined file name in which the keys will be pushed into an NginxPlus device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
privateKeyInDevice	• No	Private key to in device or not	Boolean (true or false)	NA

### Sample Request

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
    "certificateType":"PEM-.pem",
    "certificateLocation":"/home/appviewx",
    "keyLocation":"/home/appviewx",
    "privateKeyInDevice": true
  }
}
```

### Sample Response

```
{
  "response": [
```

```
{
  "requestId": "156",
  "connectorId": "device_name:@default:@clntssl:@9eb94a53963d1ae326dbf4cda6077f55baa8476e"
}
],
"message": "1 connector(s) saved and push operation has been triggered.",
"appStatusCode": null,
"tags": {},
"headers": null
}
```

Certificate details for HAProxy

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.pem
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an HAproxy device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyLocation	Yes, if privateKeyInDevice is selected	A user defined file name in which the keys will be pushed into an HAproxy device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
privateKeyInDevice	No	Private key in device or not	Boolean (true or false)	NA

Sample Request

```
{
  "certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles": [
```

```

"device_name:@serverssl-insecure-compatible:@Common"
],
"certificateDetails":{
"certificateType":"PEM-.pem",
"certificateFileName":"/opt/haproxy/certificate_push.pem",
"privateKeyLocation": true,
"privateKeyLocation": "/opt/haproxy/test_cert.key"
}
}

```

### Sample Response

```

{
"response": [
{
"requestId": "156",
"connectorId": "device_name:@default:@clientsl:@9eb94a53963d1ae326bf4cda6077f55baa8476e"
}
],
"message": "1 connector(s) saved and push operation has been triggered.",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## Response for Push and Bind ASYNC API

Response Details: (Common for all vendors)

Name	Description	Schema
response	Contains Request id and connector id.	Object
message	Success or failure messages	String
appStatusCode	Application specific status code for the response.	String
tags	More info in case of failure response	Object
headers	Details of the response headers if any	Object

## Response Details :

## Sample response (Common for all vendors)

```
{
  "response": [
    {
      "requestId": "156",
      "connectorId": "device_name:@default:@clntssl:@9eb94a53963d1ae326dbf4cda6077f55baa8476e"
    }
  ],
  "message": "1 connector(s) saved and push operation has been triggered.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

# Chapter 147: Push and Bind Certificate to ADC Devices

- [Before you begin](#)
- [Fetch available profiles API](#)
- [Request for Push and Bind SYNC API](#)
- [Response for Push and Bind SYNC API](#)

## Before you begin

Before attempting to push and bind certificate to a an ADC device through AppViewX, the user has to ensure the following:

- ADC devices should be added and should be in **Managed** in AppViewX.
- A valid certificate should be used to push the certificate.
- Profiles or virtual servers should be available. This can be fetched using [Fetch Available Profiles API](#).

## Fetch available profiles API

HTTP Method: GET

URL: */certificate/profiles*

Query Parameters : category, vendor, deviceName

### Request Details

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided

Name	Type	Description	Field Type	Constraints
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external).	String	<b>Possible values:</b> WEB, external
category	Query	Specifies the device category. It is a mandatory field.	String	<b>Possible values:</b> ADC, Server, Firewall
vendor	Query	Vendor for the chosen device. For example, F5 is a vendor for the ADC category. It is a mandatory field.	String	<b>Possible values:</b> F5, Citrix, AVI, A10, AmazonELB, NginxPlus, HAProxy
certificateUuid	Query	Resource id of the certificate	String	This can be obtained from search api
deviceName	Query	Name of the device as per AppViewX Device Inventory	String	NA
inventory	Query	Name of AppViewX inventory where certificate is present	String	<b>Possible values:</b> Server, client, code signing, device

## Sample Request

*https://<appviewx node ip>:<gateway port>/avxapi/certificate/profiles?  
gwkey=f000ca01&category=ADC&vendor=F5&deviceName=F5\_device&gwsource=external*

## Response Details

Name	Description	Field Type
response	Contains the mentioned response attributes for the fetch profiles request <ul style="list-style-type: none"> <li>• objects</li> <li>• totalRecords</li> <li>• obtainedRecords</li> <li>• obtainedRecordRange</li> </ul>	List of response
objects	List of available device profile Ids.	List of String
totalRecords	Total number of profiles fetched.	Integer
obtainedRecords	Total number of profiles fetched.	Integer
obtainedRecordRange	Range of record found.	Object
message	Success message of the action or failure description in case of error	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

## Sample Response

```
{
  "response": {
    "objects": [
      "device_name",
      "device_name:@apm-default-serverssl:@Common",
      "device_name:@clientssl-insecure-compatible:@Common",
      "device_name:@clientssl-secure:@Common",
      "device_name:@clientssl:@Common",
    ]
  }
}
```

```

"device_name:@crypto-client-default-serverssl:@Common",
"device_name:@crypto-server-default-clientssl:@Common",
"device_name:@pcoip-default-serverssl:@Common",
"device_name:@serverssl-insecure-compatible:@Common",
"device_name:@serverssl:@Common",
"device_name:@splitsession-default-clientssl:@Common",
"device_name:@splitsession-default-serverssl:@Common",
"device_name:@wom-default-clientssl:@Common",
"device_name:@wom-default-serverssl:@Common"
],
"totalRecords": 14,
"obtainedRecords": 14,
"obtainedRecordRange": {
"start": 0,
"end": 0
}
},
"message": "14 record(s) found",
"appStatusCode": null,
"tags": {},
"headers": null
}

```

## Request for Push and Bind SYNC API

URL: */certificate/profiles/push*

Type: POST

Query Parameters : ttl

Name	Mandatory	Description	Schema
certificateId	Yes	Either certificateId or certificateUuid is mandatory	String
certificateUuid	Yes	Either certificateId or certificateUuid is mandatory	String
selectedProfiles	Yes	Profiles to push the certificate	List<String>
certificateDetails	Yes	Details of certificate to push	json



**Note:** `ttl` is a query param which is **time to live**, if the total time taken exceeds this then the request would be **in progress** in the response, if the operation is successful within this `ttl` then the response would be **success**.

Sample URL :

`https://<appviewx node id>:<gateway port>/avxapi/certificate/profiles/push?gwkey=f000ca01&gwsources=external&ttl=120`

F5 Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an F5 device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an F5 device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName

Name	Mandatory	Description	Field Type	Constraints
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
rootCertificateFileName	<ul style="list-style-type: none"> <li>• No.</li> <li>• Yes,if “pushRootAndIntermediateCertificates” is true.</li> </ul>	Certificate name in which name the intermediate and root certificates to be pushed in a F5 device.	String	Intermediate file or bundle name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name

Sample Request for F5

```
{
  "certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles": [
    "device_name:@default:@clientssl"
  ],
  "certificateDetails": {
    "certificateFileName": "test123.crt",
    "certificateType": "PEM-.crt",
    "privateKeyFileName": "test123.key",
    "pushRootAndIntermediateCertificates": true,
    "rootCertificateFileName": "test123_root.crt"
  }
}
```

```
}
}
```

Citrix Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt, PEM-.pem, PEM-.cer, DER-.der, DER-.cer
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an Citrix device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an Citrix device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName

Name	Mandatory	Description	Field Type	Constraints
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
rootCertificateFileName	<ul style="list-style-type: none"> <li>• No.</li> <li>• Yes,if “pushRootAndIntermediateCertificates” is true.</li> </ul>	Certificate name in which name the intermediate and root certificates to be pushed in a Citrix device.	String	Intermediate file or bundle name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name
sniCert	No	To enable SNI Push for Certificate	Boolean (true/false)	NA
sniStatus	No	To enable SNI in Virtual Server	Boolean (true/false)	NA

Sample Request

```
{
  "certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles": [
```

```

"device_name": "@default:@sslvs"
],
"certificateDetails": {
"certificateFileName": "test123.crt",
"certificateKeyPairName": "test123",
"certificateType": "PEM-.crt",
"intermediateCertificateFileName": [
"test123_inter.crt"
],
"privateKeyFileName": "test123.key",
"pushRootAndIntermediateCertificates": true,
"rootCertificateFileName": "test123_root.crt",
"sniCert": "false",
"sniStatus": "false"
}
}

```

### AVI Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b>  PEM-.pem
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an AVI device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.

Name	Mandatory	Description	Field Type	Constraints
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an AVI device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA
intermediateCertificateFileNames	Yes, if “pushRootAndIntermediateCertificates” is true.	Intermediate certificate file name to be pushed into device	List<String>	Intermediate file or bundle name should not start and end with special characters. No special characters except (-, _, .) are allowed and should not be the same as the certificate file name
rootCertificateFileName	<ul style="list-style-type: none"> <li>• No.</li> <li>• Yes,if “pushRootAndIntermediateCertificates” is true.</li> </ul>	Certificate name in which name the intermediate and root	String	Root file name should not start and end with special characters. No special characters except

## Push and Bind Certificate to ADC Devices

Name	Mandatory	Description	Field Type	Constraints
		certificates to be pushed in a AVI device.		(-, _, .) are allowed and should not be the same as the certificate file name

### Sample Request

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
    "certificateType":"PEM-.pem",
    "certificateFileName":"certificate_push.pem",
    "privateKeyFileName":"certificate_push.key",
    "pushRootAndIntermediateCertificates": true,
    "intermediateCertificateFileNames":["test_cert_inter.crt"],
    "rootCertificateFileName": "test_cert_ca.crt"
  }
}
```

### A10 Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt, PEM-.pem, PEM-.cer, DER-.der,

Push and Bind Certificate to ADC Devices

Name	Mandatory	Description	Field Type	Constraints
				DER-.cer, PKCS#7-.p7b, PKCS#7-.p7c, PKCS#7-.p12, PKCS#7-.pfx
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an A10 device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyFileName	Yes	A user defined file name in which the keys will be pushed into an A10 device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
pfxPassword	Yes if pfx and pkcs12 type are selected	A password for the pfx type of files	String	This should be Base64 encoded
pushRootAndIntermediateCertificates	• No	Root and Intermediate certificate also needs to be	Boolean (true or false)	NA

Name	Mandatory	Description	Field Type	Constraints
		pushed or not		
intermediateCertificateFileNames	<ul style="list-style-type: none"> <li>• Yes of pushRootAndIntermediateCertificates is set true</li> </ul>	Intermediate file name to be pushed into device	List<String>	NA

### Sample Request

#### Sample request for non pfx and p12 cert types

```
{ "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[ "device_name:@serverssl-insecure-compatible:@Common" ],
  "certificateDetails":{"certificateType":"PEM-.crt",
  "certificateFileName":"certificate_push.crt",
  "privateKeyFileName":"certificate_push.key", "pushRootAndIntermediateCertificates":
  true, "intermediateCertificateFileNames":["test_cert_ca.crt"] } }
```

#### Sample request for pfx and p12 cert types:

```
{
"certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
"selectedProfiles":[
"device_name:@serverssl-insecure-compatible:@Common"
],
"certificateDetails":{"
"certificateType":"PKCS-.pfx",
"certificateFileName":"certificate_push.pfx",
"pushRootAndIntermediateCertificates": true,
"intermediateCertificateFileNames":["test_cert_ca.crt"],
"pfxPassword":"cGFZc3dvcnQ="
}
}
```

## AmazonELB Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b>  PEM-.pem
certificateLocation	Yes	<ul style="list-style-type: none"> <li>• ACM</li> <li>• IAM</li> </ul> Select one of these to input the certificate location	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
certCAReferenceId	String	This is applicable if ACM is selected.  Reference ID to input the certificate location	String	No special characters except ('=', '/', ',', '!', '@', '-') are allowed.
certificateFileName	Yes if certificate location selected is IAM	This is mandatory if IAM is selected. A user defined file name in which the certificate will be pushed into an AmazonELB device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
pushRootAndIntermediateCertificates	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA

## Sample Request if certificate location is ACM

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
```

```
"selectedProfiles":[
"device_name:@serverssl-insecure-compatible:@Common"
],
"certificateDetails":{
"certCAReferenceId": "arn:aws:iam::2313:amazonELB"
"certificateLocation": "ACM"
"certificateType": "PEM-.pem"
"pushRootAndIntermediateCertificates": true
}
```

Sample Request if certificate location is IAM

```
{
"certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
"selectedProfiles":[
"device_name:@serverssl-insecure-compatible:@Common"
],
"certificateDetails":{
"certificateFileName": "sample.pem",
"certificateLocation": "IAM",
"certificateType": "PEM-.pem",
"pushRootAndIntermediateCertificates": true
}
```

NginxPlus Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b> PEM-.crt, PEM-.cer, PEM-.pem
certificateLocation	Yes	A user defined file name in which the	String	The certificate file name should not start and end with special characters except

Name	Mandatory	Description	Field Type	Constraints
		certificate will be pushed into an NginxPlus device.		-, ., _. It should not be the same as CA file name and intermediate file name.
keyLocation	Yes	A user defined file name in which the keys will be pushed into an NginxPlus device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
privateKeyInDevice	No	Private key to in device or not	Boolean (true or false)	NA

### Sample Request

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
    "certificateType":"PEM-.pem",
    "certificateLocation":"/home/appviewx",
    "keyLocation":"/home/appviewx",
    "privateKeyInDevice": true
  }
}
```

### HA proxy Request

Name	Mandatory	Description	Field Type	Constraints
certificateType	Yes	Type of the certificate	String	<b>Possible Cert Types:</b>

Name	Mandatory	Description	Field Type	Constraints
				PEM-.pem
certificateFileName	Yes	A user defined file name in which the certificate will be pushed into an HAproxy device.	String	The certificate file name should not start and end with special characters except -, ., _. It should not be the same as CA file name and intermediate file name.
privateKeyLocation	Yes, if privateKeyInDevice is selected	A user defined file name in which the keys will be pushed into an HAproxy device.	String	The key file name should not start and end with special characters except -, ., _. Key file name should be same as certificateFileName
privateKeyInDevice	No	Root and Intermediate certificate also needs to be pushed or not	Boolean (true or false)	NA

### Sample Request

```
{
  "certificateUuid":"9eb94a53963d1ae326dbf4cda6077f55baa8476e",
  "selectedProfiles":[
    "device_name:@serverssl-insecure-compatible:@Common"
  ],
  "certificateDetails":{
    "certificateType":"PEM-.pem",
    "certificateFileName":"/opt/haproxy/certificate_push.pem",
    "privateKeyLocation": true,
    "privateKeyLocation": "/opt/haproxy/test_cert.key"
  }
}
```

## Response for Push and Bind SYNC API

Response Details: (Common for all vendors)

Name	Description	Schema
certificate	Details about the selected certificate which was used to push into the ADC devices	Object
deviceProfiles	Detail about the device profiles to which the certificate is binded	List<Object>
applicationConnectors	Details about the application connectors to which push operation took place	List<Object>
success	Push operation succeeded or not.  Possible values- true or false	Boolean (true or false)
messages	Details of the push operation	List<String>

### Response Details

```
{
  "response": {
    "certificate": <Selected Certificate>,
    "deviceProfiles": <Selected Profiles>,
    "applicationConnectors": <Application connectors with its work order status>,
    "success": <Status of the operation>,
    "messages": <Response message for the push action performed>
  },
  "message": null,
  "appStatusCode": <Error Code>,
  "tags": {},
  "headers": null
}
```

### Sample response (Common for all vendors)

```
{
  "response": {
    "certificate": {
      "commonName": "appviewx_test",
```

```

"serialNumber": "C4:DA:38:94:6B:A7:08:92:FB:A5:31:89:60:C6:D3:E7",
"issuerCommonName": "AppViewX Intermediate CA",
"status": "Managed",
"avxStatus": null,
"associatedObjects": [
"device_name::serverssl-insecure-compatible:Common:server-ssl"
],
"discoverySources": [
"device_name"
],
"subjectOrganization": "",
"subjectOrganizationUnit": "",
"subjectLocality": "",
"subjectState": "",
"subjectCountry": "",
"issuerOrganization": "AppViewX Inc",
"issuerOrganizationUnit": "",
"issuerLocality": "Seattle",
"issuerState": "Washington",
"issuerCountry": "US",
"version": "3",
"validFrom": 1617025560000,
"validTo": 1648561560000,
"validFor": "364 day(s) 23 hr(s) 57 min(s)",
"keyAlgorithmAndSize": "RSA 1024",
"signatureAlgorithm": "SHA160withRSA",
"signatureHashAlgorithm": "SHA160",
"keyUsage": "DigitalSignature, KeyEncipherment",
"extendedKeyUsage": "Server Authentication(1.3.6.1.5.5.7.3.1) Client Authentication(1.3.6.1.5.5.7.3.2) ",
"basicConstraints": "Subject Type=End entity, Path Length=none",
"group": "Default",
"subjectAlternativeNames": [
"DNS : appviewx_test"
],
"complianceStatus": "Compliant",
"applications": [],
"expiryStatus": "Valid",

```

```

"permission": null,
"category": "Server",
"uuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
"id": "6061da0e92d28c50bd40336b",
"certificateAuthority": "AppViewX",
"authorityKeyIdentifier": "D0:D7:4B:D0:82:85:A4:98:70:6E:75:97:26:C6:A3:14:A5:C0:31:4D",
"subjectKeyIdentifier": "9D:1D:77:A0:CA:35:C5:5B:D4:3A:70:1A:AB:B0:2A:DD:CD:61:D9:80",
"issuerSerialNumber": "2C:F4:8E:5F:5F:D2:C2:F2:8A:DB:5A:D4:A1:06:A5:B2",
"authorityInfoAccess": [
  "AuthorityInfoAccess : [ accessMethod : 1.3.6.1.5.5.7.48.1, alternativeName : , url :
  https://172.18.0.157:31443/avxapi/controller/avxocsp?issuerserialnumber=59755839906513824709803510723863029170 ]"
],
"certificatePolicies": [],
"crDistributionPoints": [
  "CrDistributionPoint : [ name : , url : https://172.18.0.157:31443/avxapi/controller/avxcrl?crlFileName=59755839906513824709803510723863029170.crl ]"
],
"thumbprintAlgorithm": "SHA-1",
"thumbPrint": "64:E8:0C:89:5F:6A:F1:2D:09:47:8C:5B:D5:DC:1B:CE:42:78:06:E7",
"type": "Others",
"genericFields": {
  "vs_ip_AppViewX": "",
  "device_name_AppViewX": ""
},
"certAttributes": {
  "test": "test"
},
"validFromDate": null,
"validToDate": null,
"discoveredFileNames": [],
"issuingTemplate": null,
"csrGenerationSource": "appviewx",
"certificateHSMDetails": null,
"deviceDetails": null,
"newConnectors": [],
"csrAvailable": true,
"enhancedSANTypes": null,
"autoRenewDate": "",

```

```

"missingParamsForAutoRenew": "CSR parameters are available for certificate renewal",

"base64ImageContent": null,

"caConnectorName": null,

"caSettingName": null,

"suspendedCertificate": false,

"comments": null,

"mailAddress": "",

"streetAddress": "",

"postalCode": "",

"publicKeyModulus": null,

"requestIds": [

  "R145",

  "R146"

],

"orderId": null,

"publicKey":

"30:81:89:02:81:81:00:8E:EA:40:EF:61:06:00:89:9F:99:70:61:03:C3:D2:28:C1:EF:12:B2:FB:1A:6E:65:1C:85:3D:E1:8D:BD:DF:68:C9:76:A4:23:0D:79:A2:E6:52:F

0:68:FC:AD:87:D2:D7:70:7C:76:C5:3F:A5:96:7B:D3:74:8A:5A:98:70:48:95:37:51:E3:33:1B:A7:32:19:E5:39:54:38:8A:1A:5B:A0:9C:5D:B5:D6:A6:EC:D9:DB:8A:F

B:B8:BD:66:A6:E2:F9:D8:BF:AB:4D:F1:D3:31:E1:CB:6E:84:CF:C3:6F:E8:1F:E3:AF:12:F4:22:11:13:5E:F3:56:B1:8B:4E:18:BC:4D:02:03:01:00:01",

"ellipticCurve": null,

"issuedByRootCertificate": false,

"cumulativeSanCount": 1,

"privatekeyAvailable": true

},

"deviceProfiles": [

{

"profileId": "device_name:@serverssl-insecure-compatible:@Common",

"vendor": "F5",

"deviceName": "device_name",

"vendorCategory": "ADC",

"profileIdentifierForView": "device_name::serverssl-insecure-compatible:Common:server-ssl",

"profileDisplayOrder": "Partition:@Profile Name:@SSL Type",

"connectorType": "PROFILE_CONNECTOR",

"applications": [],

"holisticViewData": {

"Partition": "Common",

"Profile Name": "serverssl-insecure-compatible",

```

```

"SSL Type": "server-ssl"
},
"overviewData": {},
"inventoryData": {
"associatedObjects": "device_name::serverssl-insecure-compatible:Common:server-ssl"
},
"vendorProperties": {
"partition": "Common",
"sslType": "server-ssl",
"name": "serverssl-insecure-compatible",
"partitionNameWithoutPath": "Common",
"iAppProfile": false
}
},
],
"applicationConnectors": [
{
"appconnectorId": "device_name:@serverssl-insecure-compatible:@Common:@9eb94a53963d1ae326dbf4cda6077f55baa8476e",
"certificateUuid": "9eb94a53963d1ae326dbf4cda6077f55baa8476e",
"profileInfo": {
"profileId": "device_name:@serverssl-insecure-compatible:@Common",
"vendor": "F5",
"deviceName": "device_name",
"vendorCategory": "ADC",
"profileIdentifierForView": "device_name::serverssl-insecure-compatible:Common:server-ssl",
"profileDisplayOrder": "Partition:@Profile Name:@SSL Type",
"connectorType": "PROFILE_CONNECTOR",
"applications": [],
"holisticViewData": {
"Partition": "Common",
"Profile Name": "serverssl-insecure-compatible",
"SSL Type": "server-ssl"
},
"overviewData": {},
"inventoryData": {
"associatedObjects": "device_name::serverssl-insecure-compatible:Common:server-ssl"
},

```

```

"vendorProperties": {
  "partition": "Common",
  "sslType": "server-ssl",
  "certificateFileName": "certificate_push_202103291347.crt",
  "rootCertificateFileName": "test_cert_ca.crt",
  "certificatePartition": "Common",
  "name": "serverssl-insecure-compatible",
  "partitionNameWithoutPath": "Common",
  "privateKeyPartition": "Common",
  "keyType": "RSA",
  "privateKeyFileName": "certificate_push_202103291347.key",
  "iAppProfile": false
},
},
"backupInfo": {
  "rollbackEligibility": false,
  "backupDisplayMessage": null,
  "backupCertUuid": null,
  "backupCertCommonName": null,
  "backupCertSerialNumber": null,
  "unbindCertificate": false,
  "backupProperties": null
},
"syncInfo": {
  "syncMessage": {
    "title": "Certificate Push",
    "message": "Certificate pushed to the device successfully",
    "utcTime": "2021-03-29T13:48:30.605",
    "status": "SUCCESS",
    "batchId": null
  },
  "syncStatus": "SYNCRONIZED"
},
"userPreference": {
  "autoPush": false,
  "overwrite": true,
  "securePush": false,

```

```

"userName": "admin",
"scheduledPushDate": null,
"modifiedData": true,
"modifiedTime": "2021-03-29T13:46:34.691",
"fileProperties": {
"pushRootAndIntermediateCertificates": false,
"certificateFileName": "certificate_push.crt",
"rootCertificateFileName": "test_cert_ca.crt",
"privateKeyFileName": "certificate_push.key"
},
"validationType": "DEFAULT",
"customApplications": null
},
"vendorScriptDetails": [],
"workflowDetails": {
"requestId": "146",
"workOrderId": "0",
"workflowStatus": "In Progress",
"workflowType": "VW",
"userName": "admin",
"actionType": "PUSHBIND",
"actionTypeDisplayName": "Push",
"taskId": "avxapi_5",
"taskName": "Post Push Script Execution",
"taskType": "others",
"taskStatus": "In Progress",
"currentAction": null,
"proceedProcess": null,
"breakdownProcess": null,
"reviewComponent": false,
"implemetationTime": 0,
"workOrderLogs": [
"Pre Push Script not available",
"Certificate push successful in device : device_name. Certificate/Key/CA file name already exists in device with different content. Certificate push has been
completed successfully with names: [ Certificate File Name: certificate_push_202103291347.crt, Private Key File Name: certificate_push_202103291347.key,
CA File Name: test_cert_ca.crt ]."
]

```

```
},  
"loggerStatus": null,  
"previousApplicationConnectorId": null  
}  
],  
"success": true,  
"messages": [  
"Certificate push to the profiles completed successfully"  
]  
},  
"message": null,  
"appStatusCode": null,  
"tags": {},  
"headers": null  
}
```

# Chapter 148: Rollback to ADC profile

- [Before you begin](#)
- [Request Structure](#)
- [Response Structure](#)
- [Status Codes](#)
- [Sample Request/Response](#)

## Before you begin

Before attempting to rollback a certificate to a particular ADC device through AppViewX, the user has to ensure the following:

- ADC devices should be added in AppViewX.
- The device should be in Managed state

## Request Structure

URL: /certificate/rollback

Type: POST

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided

Name	Type	Description	Field Type	Constraints
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsource	Query	Source from which the request is triggered (E.g. external)	String	NA
autoApproval	Query	Auto approval needed for the action or not	String	Value should be yes
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

## Payload

Name	Mandatory	Description	Field Type	Constraints
applicationConnectorIds	Yes	Application connector id	List of Strings	

## Response Structure

202 Accepted returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the rollback request	List of response
message	Success message of the action or failure description in case of error	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

## response

Name	Description	Field Type
requestId	Request Id for rollback action for the application connector	String
connectorId	Application connector Id	String

## Status Codes

HTTP Status code	StatusCode	Message	Possible remediation
202 Accepted	NA	App connector rollback action initiated for 1 connector(s).	NA
202 Accepted	NA	Operation cannot be completed for one or more devices as the 'Resource' allocated to you does not have write permission.	User does not have access for the device
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials	Ensure that valid username and password or valid sessionId is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid	Check and ensure that valid value is provided for <<field name>> field in the request.

HTTP Status code	StatusCode	Message	Possible remediation
		- <<field name>>	
417 Expectation failed	FIELD_VALUE_INVALID	Invalid value - <<field name>>	Check and ensure that valid value is provided for the <<field name>> field in the request.
417 Expectation failed	CERT-APP-0012	Application connector ids cannot be empty	Please provide value for the field applicationConnectorIds
417 Expectation failed	ERR_APPLICATION_CONNECTOR_LIST_RETRIVAL	Unable to retrieve connector information.	Connector may not be available. Please provide correct value for the field applicationConnectorIds
417 Expectation failed	ERR_APP_CONNECTORS_NOT_FOUND	Application connector(s) not found	Connector may not be available. Please provide correct value for the field applicationConnectorIds
417 Expectation failed	ERR_INITIALIZE_ROLLBACK_REQUEST	Unable to initialize rollback request.	NA
500 Internal Server Error	avx-common-011	Error while processing	NA

## Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{  
  "applicationConnectorIds": [ "device_name:@clientssl-insecure-compatible:@Common:@c46ec8a04da701721159ce0c3cf772367ade58cb"  
]  
}
```

## Sample Response:

```
{  
  "response": "Success",  
  "message": "App connector rollback action initiated for 1 connector(s).",  
  "appStatusCode": null,  
  "tags": null,  
  "headers": null  
}
```

# Chapter 149: Overview

The API will discover certificates and it's private keys from the firewall device.

# Chapter 150: Before you begin

Before attempting to discover certificates from a particular Firewall device through AppViewX, the user has to ensure the following:

- Firewall devices should be added in AppViewX.
- The device should be in Managed state.

# Chapter 151: Request Structure

URL: `/certificate/discovery/instance`

Type: POST

Name	Param Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login.	String	Required if username and password are not provided.
username	Header	AppViewX login username.	String	Required if sessionId is not provided.
password	Header	AppViewX login password.	String	Required if sessionId is not provided.
Content-Type	Header	Specifies the nature of the data in the payload.	String	Value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
discoveryType	Yes	Type of the discovery.	String	Possible values: ONDEMAND, SCHEDULED
name	Yes	Name of the discovery instance to be created.	String	NA

Name	Mandatory	Description	Field Type	Constraints
description	No	Description of the discovery request.	String	NA
groupName	Yes	Specifies the group under which the discovered certificate needs to be tagged.	String	Not mandatory if the field rbacRuleProcessRequired is <b>true</b> .
source	Yes	Source of the certificates.	String	Value should be <b>FIREWALL</b> .
certStatus	Yes	Specifies the certificate status to be maintained in the inventory after discovering the certificates.	String	Possible values are Managed, Monitored, None. If None certificates will not be moved to certificate inventory.
targetList	Yes	Name of the devices where the certificates need to be discovered.	List of String	Device names which are added already need to be specified.
coolingPeriod	Yes	Whether the discovery is sequential or parallel.	Number	0 - sequential 1< - parallel
associatedRule	No	Name of the rule to be associated with the discovered certificates.	String	Rule name which is added already need to be specified.
rbacRuleProcessRequired	No	Whether Rbac rule process in required or not.	Boolean (true or false)	If true new certificates will be tagged under the group specified in the rule.

# Chapter 152: Response Structure

202 Accepted returns string of type application/json with the following body params.

Name	Description	Field Type
response	Contains the response attributes for the discovery request.	response
message	Success message of the action or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

response

Name	Description	Field Type
discoveryName	Name of the discovery request created.	String
message	Success message of the action or failure description in case of error.	String

## Chapter 153: Status codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Discovery history details added successfully with discovery id 5f5b25dd2ceb1150424ed850, and discovery operation has been triggered.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials.	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>.	Check and ensure that valid value is provided for <<field name>> field in the request.
400 Bad Request	INVALID_REQUEST	Target list not found.	Check and ensure that valid value is provided for the <code>targetList</code> field in the request.
400 Bad Request	MSG_DISC_HISTORY_UNIQUE_NAME	Discovery name already exists. Please specify different Discovery name.	Please provide different value for the field name.

HTTP Status code	appStatusCode	Message	Possible remediation
417 Expectation failed	GROUP_NOT_FOUND	Specified certificate group is not found.	Check and ensure that valid value is provided for the <code>groupName</code> section in the request.
417 Expectation failed	FIELD_VALUE_INVALID	Cooling period should be 0 or greater than 1.	Please provide the correct value in the field <code>coolingPeriod</code> .
417 Expectation failed	FIELD_VALUE_INVALID	Discovery Type should be ONDEMAND or SCHEDULED.	Please provide the correct value in the field <code>discoveryType</code> .
500 Internal Server Error	avx-common-011	Error while processing.	NA

# Chapter 154: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "discoveryType": "ONDEMAND",
  "name": "ServerDiscovery",
  "description": "Discovery from server devices",
  "source": "SERVER",
  "coolingPeriod": "5",
  "associatedRule": "RevokedCertExclusion",
  "certStatus": "MANAGED",
  "rbacRuleProcessRequired": false,
  "groupName": "Default",
  "discoverAllCerts": false,
  "templatesToFilter": [],
  "targetList": [
    "ApacheLinux"
  ]
}
```

## Sample Response:

```
{
  "response": {
    "discoveryName": "ServerDiscovery",
    "message": "Discovery history details added successfully with discovery id 605dc592c0bb5b1924e6851f, and discovery operation has been triggered.",
  },
  "message": "Discovery history details added successfully with discovery id 605dc592c0bb5b1924e6851f, and discovery operation has been triggered.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

# Chapter 155: Overview

The API will discover certificates and it's private keys from the server device.

## Chapter 156: Before you begin

Before attempting to discover certificates from a particular Server device through AppViewX, the user has to ensure the following:

- Server devices should be added in AppViewX.
- The device should be in Managed state.

# Chapter 157: Request Structure

URL: /certificate/discovery/instance

Type: POST

Name	Type	Description	Field Type	Constraints
sessionId	Header	Session Id received after login	String	Required if username and password are not provided
username	Header	AppViewX login username	String	Required if sessionId is not provided
password	Header	AppViewX login password	String	Required if sessionId is not provided
Content-Type	Header	Specifies the nature of the data in the payload.	String	The value of the param should be 'application/json'
gwkey	Query	Tenant Key. This is needed only in case of multi tenant installations and can be ignored in other type of installations.	String	NA
gwsouce	Query	Source from which the request is triggered (E.g. external).	String	NA
Payload	Body	Contains all the params to be sent in the request body for the post request	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
discoveryType	Yes	Type of the discovery	String	<b>Possible values:</b> ONDEMAND, SCHEDULED
name	Yes	Name of the discovery instance to be created.	String	NA

Name	Mandatory	Description	Field Type	Constraints
description	No	Description of the discovery request	String	NA
groupName	Yes	Specifies the group under which the discovered certificate needs to be tagged.	String	Not mandatory if the field <b>rbacRuleProcessRequired</b> is true
source	Yes	Source of the certificates	String	The value should be <b>SERVER</b>
certStatus	Yes	Specifies the certificate status to be maintained in the inventory after discovering the certificates.	String	Possible values are Managed, Monitored, None. If None certificates will not be moved to certificate inventory
targetList	Yes	Name of the devices where the certificates need to be discovered	List of String	Device names that are added already need to be specified
coolingPeriod	Yes	Whether the discovery is sequential or parallel	Number	0 - sequential >1 - parallel
associatedRule	No	Name of the rule to be associated with the discovered certificates	String	Rule name which is added already need to be specified
rbacRuleProcessRequired	No	Whether Rbac rule process is required or not	Boolean (true or false)	If true new certificates will be tagged under the group specified in the rule

# Chapter 158: Response Structure

202 Accepted returns string of type application/json with the following body params

Name	Description	Field Type
response	Contains the response attributes for the discovery request	response
message	Success message of the action or failure description in case of error	String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response	NA

response

Name	Description	Field Type
discoveryName	Name of the discovery request created	String
message	Success message of the action or failure description in case of error	String

## Chapter 159: Status Codes

HTTP Status code	appStatusCode	Message	Possible remediation
202 Accepted	NA	Discovery history details added successfully with discovery id 605dc592c0bb5b1924e6851f, and discovery operation has been triggered.	NA
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials	Ensure that valid <code>username</code> and <code>password</code> or valid <code>sessionId</code> is provided as header param.
400 Bad Request	MANDATORY_FIELD_MISSING	Mandatory field is missing or invalid - <<field name>>	Check and ensure that valid value is provided for <<field name>> field in the request.
400 Bad Request	INVALID_REQUEST	Target list not found	Check and ensure that a valid value is provided for the <code>targetList</code> field in the request.
400 Bad Request	MSG_DISC_HISTORY_UNIQUE_NAME	The Discovery name already exists. Please specify a different Discovery name	Please provide a different value for the field <b>name</b>

HTTP Status code	appStatusCode	Message	Possible remediation
404 Not Found	GROUP_NOT_FOUND	A specified certificate group is not found	Check and ensure that valid value is provided for the <b>groupName</b> section in the request.
417 Expectation failed	FIELD_VALUE_INVALID	cooling period should be 0 or greater than 1	Please provide the correct value in the field <b>coolingPeriod</b> .
417 Expectation failed	FIELD_VALUE_INVALID	Discovery Type should be ONDEMAND or SCHEDULED	Please provide the correct value in the field <b>discoveryType</b> .
500 Internal Server Error	avx-common-011	Error while processing	NA

# Chapter 160: Sample Request/Response

- [Sample Request:](#)
- [Sample Response:](#)

## Sample Request:

```
{
  "discoveryType": "ONDEMAND",
  "name": "ServerDiscovery",
  "description": "Discovery from server devices",
  "source": "SERVER",
  "coolingPeriod": "5",
  "associatedRule": "RevokedCertExclusion",
  "certStatus": "MANAGED",
  "rbacRuleProcessRequired": false,
  "groupName": "Default",
  "discoverAllCerts": false,
  "templatesToFilter": [],
  "targetList": [
    "ApacheLinux"
  ]
}
```

## Sample Response:

```
{
  "response": {
    "discoveryName": "ServerDiscovery",
    "message": "Discovery history details added successfully with discovery id 605dc592c0bb5b1924e6851f, and discovery operation has been triggered.",
  },
  "message": "Discovery history details added successfully with discovery id 605dc592c0bb5b1924e6851f, and discovery operation has been triggered.",
  "appStatusCode": null,
  "tags": {},
  "headers": null
}
```

# Chapter 161: Glossary

This table describes common terms used in this guide.

Terms	Definition
ACME	Automatic Certificate Management Environment (ACME) protocol is a communications protocol for automating the certificate enrollment to the CA and provision the certificate on the requesting entity.
Certificate Authority (CA)	A certificate authority or certification authority is an entity that issues digital certificates. It certifies the ownership of the key pair belongs to the subject within the certificate.
X.509 Digital Certificate	X.509 is a standard defining the format of public key certificates. An X. 509 certificate is using the widely accepted public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.
Identity	The digital certificate can also be called a Digital ID or Identity for the subject to whom it is certified.
PKI	A public key infrastructure (PKI) is a technology containing a set of roles, policies, and procedures needed to create, distribute, store and revoke digital certificates and manage public-key encryption.
KMIP	The Key Management Interoperability Protocol is a communication standard protocol that defines message formats for the management of cryptographic keys on a key management server.
MDM	Mobile Device Management (MDM) is the administration of mobile devices, such as smartphones, tablet computers, and laptops.
EST	The <b>Enrollment</b> over Secure Transport or <b>EST</b> is a cryptographic protocol that describes an X. 509 certificate management protocol targeting public key infrastructure (PKI) clients that need to acquire client certificates and associated certificate authority (CA) certificates. <b>EST</b> is described in RFC 7030
SCEP	Simple Certificate Enrollment Protocol (SCEP) is an IETF RFC. This enables network user to request their digital certificate electronically and as simply as possible. Supported by most of the network devices.
SSL/TLS Certificates	SSL refers to Secure Sockets Layer whereas TLS refers to Transport Layer Security. Both are cryptographic protocols providing secure data communication in a network.