

Oracle® Communications

Cloud Native Core, Certificate Management

REST Specification Guide



Release 24.2.3

F96512-04

April 2025

ORACLE®

F96512-04

Copyright © 2023, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction	
1.1	Overview	1
1.2	Reference	1
2	OCCM Rest Specification	
2.1	OCCM Issuers	1
2.1.1	Fetch All Issuers	9
2.1.2	Fetch Issuers by Name	12
2.1.3	Add Issuer Configurations	14
2.1.4	Update Issuer Configurations	18
2.1.5	Delete Issuers Data	23
2.2	OCCM Certificates	25
2.2.1	Fetch all Certificate Configurations	33
2.2.2	Fetch Certificate Configurations by UUID	35
2.2.3	Add Certificate Configurations	37
2.2.4	Delete Certificate Configuration Data	41
2.2.5	Recreate Certificates	43
2.3	OCCM Logging Resource	46
2.3.1	Fetch Logging Configuration for a Service	46
2.3.2	Fetch Logging Configurations for All Services	47
2.3.3	Update Logging Configurations for a Service	47

My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
 - For Technical issues such as creating a new Service Request (SR), select **1**.
 - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Acronym	Definition
3GPP	3rd Generation Partnership Project
API	Application Programming Interface
CCA	Client Credentials Assertions
CMP	Certificate Management Protocol
CNC	Cloud Native Core
CNC Console	Cloud Native Configuration Console
OCCM	Oracle Communications Certificate Management
CA	Certification Authority is a trusted entity that issues Secure Sockets Layer (SSL) certificates. CAs are also called issuer in this document.
DNS	Domain Name Server
EE	End Entity
ECC	Elliptic Curve Cryptography
HSM	Hardware Security Module
IDP	Identity Provider
PKI	Public Key Infrastructure
RA	Registration Authority
RSA	Rivest-Shamir-Adleman
SAN	Subject Alternative Name
URI	Uniform Resource Indicator
URN	Uniform Resource Name
CMP Identity Key	Private Key used by Certificate Management to sign the CMPv2 requests and establish trust between Certificate Management and CA.
CMP Identity Certificate	Certificate that corresponds to and certifies the CMP Identity Key. It is included in the CMPv2 requests for authentication by CA.

What's New in This Guide

This section introduces the documentation updates for Release 24.2.x.

Release 24.2.3 - F96512-04, April 2025

There are no changes in this release.

Release 24.2.2 - F96512-03, January 2025

There are no changes in this release.

Release 24.2.1 - F96512-02, October 2024

There are no changes in this release.

Release 24.2.0 - F96512-01, July 2024

Added the REST specifications for recreating certificates in [Recreate Certificates](#).

1

Introduction

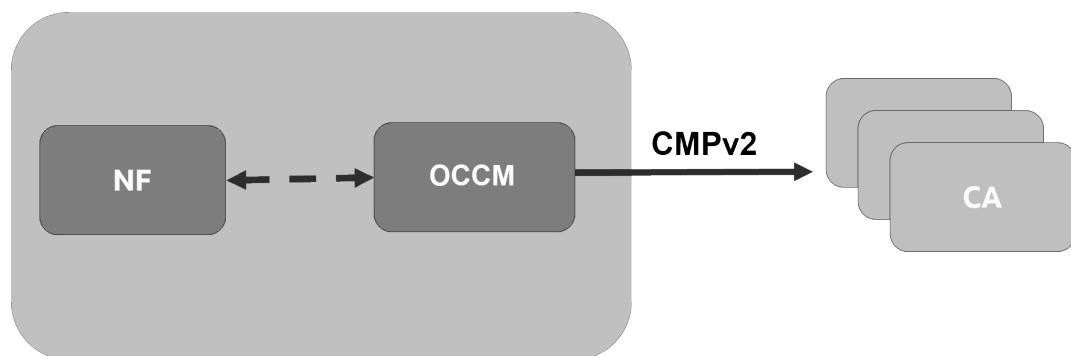
This document provides information about how to configure the services and manageable objects in OCCM using Representational State Transfer Application Program Interfaces (REST APIs).

1.1 Overview

OCCM integrates with the Certificate Authority(s) using Certificate Management Protocol Version 2 (CMPv2) and RFC4210 to facilitate these certificate management operations:

- Operator-initiated certificate creation
- Operator-initiated certificate recreation
- Automatic certificate monitoring and renewal

Figure 1-1 OCCM Integration with CA



OCCM supports transport of CMPv2 messages using HTTP-based protocol.

OCCM provides the following mechanisms to establish initial trust between OCCM and CA(s):

1. Certificate-based message signing
2. Pre-shared key or MAC based authentication

All the subsequent CMPv2 procedures are authenticated using the certificate-based mechanism in compliance with 3GPP TS 33.310.

The keys and X.509 certificates are managed using Kubernetes secrets.

1.2 Reference

Refer to the following documents for more information:

- *Oracle Communications Cloud Native Core, Certificate Management User Guide*
- *Oracle Communications Cloud Native Core, Certificate Management Installation, Upgrade, and Fault Recovery Guide*

- *Oracle Communications Cloud Native Core Security Guide*
- *Oracle Communications Cloud Native Core Solution Upgrade Guide*

2

OCCM Rest Specification

This chapter provides information about REST specifications used in Oracle Communications Cloud Native Core, Certificate Management

OCCM can be configured using Helm configurations, REST APIs, and Cloud Native Configuration Console (CNC Console). REST configurations can also be performed using the Cloud Native Configuration (CNC) Console.

For HELM configurations, see *Oracle Communications Cloud Native Core, Certificate Management Installation, Upgrade, and Fault Recovery Guide*.

For the configurations using CNC Console, see *Oracle Communications Cloud Native Core, Certificate Management User Guide*.

For installing OCCM in an existing NF deployment, see 'Introducing OCCM on an Existing NF Deployment' section in the *Oracle Communications Cloud Native Core, Certificate Management User Guide*.

2.1 OCCM Issuers

OCCM Issuers Data Model

Table 2-1 OCCM Issuers Request Parameters

Field Name	Data Type	Description
name	String	This is a mandatory parameter. Name of CA
server	String	This is a mandatory parameter. Domain URL of CA Note: The user must provide the port where the CA server is running, otherwise application will take it as default port 80.

Table 2-1 (Cont.) OCCM Issuers Request Parameters

Field Name	Data Type	Description
recipientDN	String	This is a mandatory parameter. Distinguished name(DN) of the CMP server(usually the addressed CA). Used in the recipient field of CMP request message headers. The argument must be formatted as /type0=value0/type1=value1/type2=.... Special characters may be escaped by \ (backslash); whitespace is retained. Empty values are permitted, but the corresponding type will not be included. Giving a single / will lead to an empty sequence of RDNs (a NULL-DN). Multi-valued RDNs can be formed by placing a + character instead of a / between the AttributeValueAssertions (AVAs) that specify the members of the set. Example: /DC=org/DC=OpenSSL/DC=users/UID=123456+CN=John Doe
issuerDN	String	This is an optional parameter. X509 issuer Distinguished Name of the CA server to place in the requested certificate template in IR/KUR. The argument must be formatted as /type0=value0/type1=value1/type2=.... Special characters may be escaped by \ (backslash); whitespace is retained. Empty values are permitted, but the corresponding type will not be included. Giving a single / will lead to an empty sequence of RDNs (a NULL-DN). Multi-valued RDNs can be formed by placing a + character instead of a / between the AttributeValueAssertions (AVAs) that specify the members of the set. Example: /DC=org/DC=OpenSSL/DC=users/UID=123456+CN=John Doe

Table 2-1 (Cont.) OCCM Issuers Request Parameters

Field Name	Data Type	Description
totalTimeout	String	This is a mandatory parameter. Maximum total number of seconds a CMP transaction may take. Default Value: 720 seconds. Max value: 21600 seconds. Note: totalTimeout should always be greater than messageTimeout.
messageTimeout	String	This is a mandatory parameter. Number of seconds a CMP request-response message round trip is allowed to take before a timeout error is returned. Default Value 120 seconds. Max is 600 seconds. Imp Note: messageTimeout should always be less than totalTimeout.
cmpProtectionOccmCert	Object	This is a mandatory parameter except when OCCM certificate is manually configured. CMP client authentication options for OCCM certificate
cmpProtectionOccmCert.type	Enum	This is a mandatory parameter. MAC or SIGNATURE Possible values: MAC SIGNATURE
cmpProtectionOccmCert.digestAlgorithm	Enum	This is a mandatory parameter except when cmpProtectionOccmCert.type selected is MAC. Supported digest to use. Default Value: SHA256 Possible values: SHA256, SHA384, SHA512
cmpProtectionOccmCert.macAlgorithm	Enum	This is a mandatory parameter except when cmpProtectionOccmCert.type selected is SIGNATURE) MAC algorithm to use. Possible values: HMACSHA256 HMACSHA384 HMACSHA512

Table 2-1 (Cont.) OCCM Issuers Request Parameters

Field Name	Data Type	Description
cmpProtectionOccmCert.mack8s SecretIn	Object	This is a mandatory parameter except when cmpProtectionOccmCert.type selected is SIGNATURE. Kubernetes secret input details for MAC based authentication of OCCM cert.
cmpProtectionOccmCert.mack8s SecretIn.namespace	String	This is a mandatory parameter. Kubernetes secret namespace where MAC secret is present.
cmpProtectionOccmCert.mack8s SecretIn.namespace	String	This is a mandatory parameter. Name of Kubernetes secret holding MAC secret (pre-shared key) and reference information.
cmpProtectionOccmCert.mack8s SecretIn.passKey	String	This is a mandatory parameter. Kubernetes secret data key against which MAC secret is provided.
cmpProtectionOccmCert.mack8s SecretIn.refKey	String	This is an optional parameter. Kubernetes secret data key against which reference string is provided.
cmpProtectionOccmCert.signK8s SecretIn	Object	This is a mandatory parameter except when cmpProtectionOccmCert.type selected is MAC. Kubernetes secret input details for Signature based authentication of OCCM cert.
cmpProtectionOccmCert.signK8s SecretIn.name	String	This is a mandatory parameter. Kubernetes secret namespace where OCCM Sign secret is present.
cmpProtectionOccmCert.signK8s SecretIn.name	String	This is a mandatory parameter. Name of Kubernetes secret holding pre-configured private key and certificate.
cmpProtectionOccmCert.signK8s SecretIn.key	String	This is a mandatory parameter. Kubernetes secret data key against which the pre-configured private key file (private key file for the client's current CMP signer certificate) is provided.
cmpProtectionOccmCert.signK8s SecretIn.cert	String	This is a mandatory parameter. Kubernetes secret data key against which the pre-configured certificate (client's current CMP signer certificate) is provided.

Table 2-1 (Cont.) OCCM Issuers Request Parameters

Field Name	Data Type	Description
cmpProtectionOcmCert.signK8s SecretIn.extraCerts	Object	This is a mandatory parameter. List of Kubernetes secret data keys against which the certificates to append in the extraCerts field can be provided. They can be used as the default CMP signer certificate chain to include.
cmpProtectionOtherCert	Object	This is a mandatory parameter. CMP client authentication options for Other(NF) certificate
cmpProtectionOtherCert.type	Enum	This is a mandatory parameter. Possible Value: SIGNATURE
cmpProtectionOtherCert.digestAl gorithm	Enum	This is a mandatory parameter. Supported digest to use. Default Value: SHA256 Possible values: SHA256, SHA384, SHA512
cmpProtectionOtherCert.signK8s SecretIn	Object	This is a mandatory parameter. Kubernetes secret input details for Signature based authentication of Other (NF) cert.
cmpProtectionOtherCert.signK8s SecretIn.namespace	String	This is a mandatory parameter. Kubernetes secret namespace where NF Sign secret is present.
cmpProtectionOtherCert.signK8s SecretIn.name	String	This is a mandatory parameter. Name of Kubernetes secret holding OCCM key and cert information.
cmpProtectionOtherCert.signK8s SecretIn.key	String	This is a mandatory parameter. Kubernetes secret data key against which OCCM key is provided/created based on whether OCCM cert is created in manual or automatic mode.
cmpProtectionOtherCert.signK8s SecretIn.cert	String	This is a mandatory parameter. Kubernetes secret data key against which OCCM certificate is provided/created based on whether OCCM cert is created in manual or automatic mode.
cmpProtectionOtherCert.signK8s SecretIn.extraCerts	Array	This is a mandatory parameter. List of Kubernetes secret data keys against which the certificates to append in the extraCerts field can be provided or will be created (if received from CA) along with the OCCM cert, based on whether OCCM cert is created in manual or automatic mode.

Table 2-1 (Cont.) OCCM Issuers Request Parameters

Field Name	Data Type	Description
occmTrustStoreK8sSecretIn	Object	This is a mandatory parameter. Kubernetes secret input which holds OCCM trust store information(CA certificates). Used to validate CMP response messages.
occmTrustStoreK8sSecretIn.namespace	String	This is a mandatory parameter. Kubernetes secret namespace where OCCM trust store secret is present.
occmTrustStoreK8sSecretIn.name	String	This is a mandatory parameter. Name of Kubernetes OCCM trust store secret.
occmTrustStoreK8sSecretIn.rootCACerts	List<String>	This is an optional parameter except if occmTrustStoreK8sSecretIn.serverCert is provided. The certificate(s), typically of root CAs, the client shall use as trust anchors when validating the certificate issued by CA. Note: If server cert is present this is ignored.
occmTrustStoreK8sSecretIn.intermediateCACerts	List<String>	This is an optional parameter. Any non-trusted intermediate CA certificate(s) to use when validating newly enrolled certificates.
occmTrustStoreK8sSecretIn.serverCert	String	This is a mandatory parameter except if occmTrustStoreK8sSecretIn.rootCACerts is provided. CMP/CA server's certificate to expect and directly trust when validating the certificate issued by CA. Note: If this is present root CA certs will be ignored.
uuid	String	Unique ID for logging and tracking purpose
tlsConfig.enableTLS	boolean	This is an optional parameter. This field when set true "-tls_used" will be included in openssl cmp cmd for TLS communication with CA. Server URL should include https scheme Possible values: true, false

Table 2-1 (Cont.) OCCM Issuers Request Parameters

Field Name	Data Type	Description
tlsConfig.tlsTrustStoreK8sSecretNamespace	String	This parameter is mandatory when enableTLS is set to true and optional when enableTLS is set to false. Kubernetes secret namespace where TLS trust store secret is present.
tlsConfig.tlsTrustStoreK8sSecretName	String	This parameter is mandatory when enableTLS is set to true and optional when enableTLS is set to false. Name of Kubernetes TLS trust store secret.
tlsConfig.tlsTrustStoreK8sSecretCertificates	List<String>	This parameter is mandatory when enableTLS is set to true and optional when enableTLS is set to false. Certificate(s) used for validating the certificate presented by CA(s) during TLS handshake.

Table 2-2 OCCM Issuers Response Codes

Response codes	Data type	Cardinality	Description
202 Accepted	Object (Issuers)	1	This is a mandatory parameter Object (Issuers)
200 OK	Object (Issuers) Or List(Issuers)	1	This is a mandatory parameter Object (Issuers) Or List(Issuers) matching criteria
400 Bad request	Problem Details	1	This is a mandatory parameter Input does not match to process request
500 Internal Server Error	Problem Details	1	This is a mandatory parameter Something went wrong
409 Conflict	Problem Details	1	This is a mandatory parameter Record already exists

Note

OCCM Issuers response body data model varies based on REST operation status.

OCCM Issuers JSON Payload

```

{
  "name": "",
  "server": "",
  "recipientDN": "",
  "issuerDN": "",
  "totalTimeout": "",
  "messageTimeout": "",
  "cmpProtectionOccmCert": {
    "type": null,
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "macK8sSecretIn": {
      "namespace": "",
      "name": "",
      "passKey": "",
      "refKey": ""
    },
    "signK8sSecretIn": {
      "namespace": "",
      "name": "",
      "key": "",
      "cert": "",
      "extraCerts": []
    }
  },
  "cmpProtectionOtherCert": {
    "type": "",
    "digestAlgorithm": "",
    "signK8sSecretIn": {
      "namespace": "",
      "name": "",
      "key": "",
      "cert": "",
      "extraCerts": []
    }
  },
  "occmTrustStoreK8sSecretIn": {
    "namespace": "",
    "name": "",
    "rootCACerts": [],
    "intCACerts": [],
    "serverCert": ""
  },
  "tlsConfig": {
    "enableTLS": true,
    "tlsTrustStoreK8sSecretItem": {
      "namespace": "",
      "name": "",
      "tlsTrustedCerts": []
    }
  }
}

```


2.1.1 Fetch All Issuers

OCCM Uses the GET operation to fetch all issuer details.

Resource URI: /occm-config/v1/issuers

Data structures supported by the GET Response Body on this resource

Response codes	Data type	Cardinality	Description
200 OK	Object (Issuers)	1	This is a mandatory parameter. Object (Issuers) matching criteria
500 Internal Server Error	Problem Details	1	This is a mandatory parameter. Something went wrong

Sample GET Request:

```
$ curl --location --request GET 'http://{host}:{port}/occm-config/v1/issuers' \
\
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1Ni'
```

Sample GET Response:

```
200 OK Response Body: '[{
  "uuid": "d692b217-00ca-433b-819d-cb038dba256b",
  "name": "CA-thrust-05",
  "server": "https://thrust5:8445/ejbca/publicweb/cmp/occmalias",
  "recipientDN": "/CN=ManagementCA",
  "issuerDN": "",
  "totalTimeout": "120",
  "messageTimeout": "30",
  "cmpProtectionOcmCert": {
    "type": "MAC",
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "macK8sSecretIn": {
      "namespace": "occncc-thrust5-01",
      "name": "macsecret",
      "passKey": "pwd",
      "refKey": "ref"
    },
  },
  "signK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": "",
    "cert": "",
    "extraCerts": []
  }
},
],
```

```

    "cmpProtectionOtherCert": {
      "type": "SIGNATURE",
      "digestAlgorithm": "SHA384",
      "signK8sSecretIn": {
        "namespace": "occncc-thrust5-01",
        "name": "ca-occm-key-cert-secret-thrust-5102",
        "key": "occmkey.pem",
        "cert": "occm.cer",
        "extraCerts": []
      }
    },
    "occmTrustStoreK8sSecretIn": {
      "namespace": "occncc-thrust5-01",
      "name": "ca-occm-tls-trust-store-secret-thrust-5",
      "rootCACerts": [
        "ManagementCA-chain.pem"
      ],
      "intCACerts": [],
      "serverCert": ""
    },
    "tlsConfig": {
      "enableTLS": true,
      "tlsTrustStoreK8sSecretItem": {
        "namespace": "occncc-thrust5-01",
        "name": "ca-occm-tls-trust-store-secret-thrust-5",
        "tlsTrustedCerts": [
          "ManagementCA-chain.pem",
          "RootCA.pem"
        ]
      }
    }
  },
  {
    "uuid": "897d6073-0bea-4525-b814-0fbda8b85563",
    "name": "CA7",
    "server": "https://thrust5:8445/ejbca/publicweb/cmp/occmalias",
    "recipientDN": "/CN=ManagementCA",
    "issuerDN": "",
    "totalTimeout": "120",
    "messageTimeout": "30",
    "cmpProtectionOcmCert": {
      "type": "MAC",
      "digestAlgorithm": null,
      "macAlgorithm": null,
      "macK8sSecretIn": {
        "namespace": "occncc-thrust5-01",
        "name": "macsecret",
        "passKey": "pwd",
        "refKey": "ref"
      }
    },
    "signK8sSecretIn": {
      "namespace": "",
      "name": "",
      "key": "",
      "cert": "",
      "extraCerts": []
    }
  }
}

```

```

    }
  },
  "cmpProtectionOtherCert": {
    "type": "SIGNATURE",
    "digestAlgorithm": "SHA384",
    "signK8sSecretIn": {
      "namespace": "occncc-thrust5-01",
      "name": "ca-occm-key-cert-secret-thrust-5102",
      "key": "occmkey.pem",
      "cert": "occm.cer",
      "extraCerts": []
    }
  },
  "occmTrustStoreK8sSecretIn": {
    "namespace": "occncc-thrust5-01",
    "name": "ca-occm-tls-trust-store-secret-thrust-5",
    "rootCACerts": [
      "ManagementCA-chain.pem"
    ],
    "intCACerts": [],
    "serverCert": ""
  },
  "tlsConfig": {
    "enableTLS": true,
    "tlsTrustStoreK8sSecretItem": {
      "namespace": "occncc-thrust5-01",
      "name": "ca6-occm-tls-trust-store-secret",
      "tlsTrustedCerts": [
        "ManagementCA-chain.pem"
      ]
    }
  }
},
{
  "uuid": "9c845672-ebec-4ce6-ad98-ce2e6e3fe1d5",
  "name": "CA5",
  "server": "http://thrust5:8445/ejbca/publicweb/cmp/occmalias",
  "recipientDN": "/CN=ManagementCA",
  "issuerDN": "",
  "totalTimeout": "120",
  "messageTimeout": "30",
  "cmpProtectionOccmCert": {
    "type": "MAC",
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "mack8sSecretIn": {
      "namespace": "occncc-thrust5-01",
      "name": "macsecret",
      "passKey": "pwd",
      "refKey": "ref"
    }
  },
  "signK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": "",
    "cert": "",

```

```

        "extraCerts": []
      }
    },
    "cmpProtectionOtherCert": {
      "type": "SIGNATURE",
      "digestAlgorithm": "SHA384",
      "signK8sSecretIn": {
        "namespace": "occncc-thrust5-01",
        "name": "ca-occm-key-cert-secret-thrust-5102",
        "key": "occmkey.pem",
        "cert": "occm.cer",
        "extraCerts": []
      }
    },
    "occmTrustStoreK8sSecretIn": {
      "namespace": "occncc-thrust5-01",
      "name": "ca-occm-tls-trust-store-secret-thrust-5",
      "rootCACerts": [
        "ManagementCA-chain.pem"
      ],
      "intCACerts": [],
      "serverCert": ""
    },
    "tlsConfig": {
      "enableTLS": false,
      "tlsTrustStoreK8sSecretItem": {
        "namespace": "",
        "name": "",
        "tlsTrustedCerts": [
          ""
        ]
      }
    }
  }
]'
```

2.1.2 Fetch Issuers by Name

OCCM uses the GET operation to fetch issuers by name.

Resource URI: /occm-config/v1/issuers

Data structures supported by the GET Response Body on this resource

Response codes	Data type	Cardinality	Description
200 OK	Object (Issuers)	1	This is a mandatory parameter. Object (Issuers) matching criteria
500 Internal Server Error	Problem Details	1	This is a mandatory parameter. This indicates that something went wrong.

Sample GET Request:

```
$ curl --location --request GET 'http://{host}:{port}/occm-config/v1/issuers/
4c5b4025-6c63-438c-bcd7-27b5bf8da4fd' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIs'
```

Sample GET Response:

```
200 OK Response Body: '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
  "name": "CA1",
  "server": "http://ca1-openssl-mock.ns1.svc.thrust5:8080",
  "recipientDN": "/CN=svc.thrust5",
  "issuerDN": "/CN=svc.thrust5",
  "totalTimeout": "720",
  "messageTimeout": "120",
  "cmpProtectionOccmCert": {
    "type": null,
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "macK8sSecretIn": {
      "namespace": "",
      "name": "",
      "passKey": "",
      "refKey": ""
    },
    "signK8sSecretIn": {
      "namespace": "",
      "name": "",
      "key": "",
      "cert": "",
      "extraCerts": []
    }
  },
  "cmpProtectionOtherCert": {
    "type": "SIGNATURE",
    "digestAlgorithm": "SHA256",
    "signK8sSecretIn": {
      "namespace": "ns1",
      "name": "ca1-occm-key-cert-secret",
      "key": "occmkey.pem",
      "cert": "occm.cer",
      "extraCerts": []
    }
  },
  "occmTrustStoreK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-trust-store-secret",
    "rootCACerts": [
      "caroot.cer"
    ],
    "intCACerts": [
```

```

        "intca.cer"
      ],
      "serverCert": "server.cer"
    },
    "tlsConfig": {
      "enableTLS": false,
      "tlsTrustStoreK8sSecretItem": {
        "namespace": "",
        "name": "",
        "tlsTrustedCerts": [""]
      }
    }
  }
}'

```

2.1.3 Add Issuer Configurations

OCCM Uses the POST operation to add the issuers configurations using the request body.

Resource URI: /occm-config/v1/issuers

Data structures supported by the POST Response Body on this resource

Response codes	Data type	Cardinality	Description
202 ACCEPTED	Issuers	1	This is a mandatory parameter. ssuers configuration data
400 BAD REQUEST	Problem Details	1	This is a mandatory parameter. Returns Problem Details structure
409 CONFLICT	Problem Details	1	This is a mandatory parameter. Record already exists

Sample POST Request when Enable TLS is False:

```

$ curl --location --request POST 'http://{host}:{port}/occm-config/v1/issuers' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer LxuLeX9dihXDUcoFwDw' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name": "CA1",
  "server": "http://ca1-openssl-mock.ns1.svc.thrust5:8080",
  "recipientDN": "/CN=svc.thrust5",
  "issuerDN": "/CN=svc.thrust5",
  "totalTimeout": "60",
  "messageTimeout": "30",
  "cmpProtectionOccmCert": {
    "type": null,
    "digestAlgorithm": null,
    "macAlgorithm": null,

```

```

        "macK8sSecretIn": {
            "namespace": "",
            "name": "",
            "passKey": "",
            "refKey": ""
        },
        "signK8sSecretIn": {
            "namespace": "",
            "name": "",
            "key": "",
            "cert": "",
            "extraCerts": []
        }
    },
    "cmpProtectionOtherCert": {
        "type": "SIGNATURE",
        "digestAlgorithm": "SHA256",
        "signK8sSecretIn": {
            "namespace": "ns1",
            "name": "ca1-occm-key-cert-secret",
            "key": "occmkey.pem",
            "cert": "occm.cer",
            "extraCerts": []
        }
    },
    "occmTrustStoreK8sSecretIn": {
        "namespace": "ns1",
        "name": "ca1-occm-trust-store-secret",
        "rootCACerts": ["caroot.cer"],
        "intCACerts": ["intca.cer"],
        "serverCert": ""
    },
    "tlsConfig": {
        "enableTLS": false,
        "tlsTrustStoreK8sSecretItem": {
            "namespace": "",
            "name": "",
            "tlsTrustedCerts": [""]
        }
    }
}

```

Sample POST Response:

```

200 Success Response Body: '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
  "name": "CA1",
  "server": "http://ca1-openssl-mock.ns1.svc.thrust5:8080",
  "recipientDN": "/CN=svc.thrust5",
  "issuerDN": "/CN=svc.thrust5",
  "totalTimeout": "60",
  "messageTimeout": "30",
  "cmpProtectionOccmCert": {
    "type": null,
    "digestAlgorithm": null,

```

```

        "macAlgorithm": null,
        "macK8sSecretIn": {
            "namespace": "",
            "name": "",
            "passKey": "",
            "refKey": ""
        },
        "signK8sSecretIn": {
            "namespace": "",
            "name": "",
            "key": "",
            "cert": "",
            "extraCerts": []
        }
    },
    "cmpProtectionOtherCert": {
        "type": "SIGNATURE",
        "digestAlgorithm": "SHA256",
        "signK8sSecretIn": {
            "namespace": "ns1",
            "name": "ca1-occm-key-cert-secret",
            "key": "occmkey.pem",
            "cert": "occm.cer",
            "extraCerts": []
        }
    },
    "occmTrustStoreK8sSecretIn": {
        "namespace": "ns1",
        "name": "ca1-occm-trust-store-secret",
        "rootCACerts": ["caroot.cer"],
        "intCACerts": ["intca.cer"],
        "serverCert": ""
    },
    "tlsConfig": {
        "enableTLS": false,
        "tlsTrustStoreK8sSecretItem": {
            "namespace": "",
            "name": "",
            "tlsTrustedCerts": [""]
        }
    }
}

```

Sample POST request when Enable TLS is True (HTTPS):

```

$ curl --location --request POST 'http://{host}:{port}/occm-config/v1/
issuers' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer LxuLex9dihXDUcoFwDw' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "CA-TLS-1",
    "server": "https://thrust5:8445/ejbca/publicweb/cmp/occmalias",
    "recipientDN": "/CN=svc.thrust5",

```



```

"issuerDN": "/CN=ManagementCA",
"totalTimeout": "60",
"messageTimeout": "30",
"cmpProtectionOccmCert": {
  "type": null,
  "digestAlgorithm": null,
  "macAlgorithm": null,
  "macK8sSecretIn": {
    "namespace": "",
    "name": "",
    "passKey": "",
    "refKey": ""
  },
  "signK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": "",
    "cert": "",
    "extraCerts": []
  }
},
"cmpProtectionOtherCert": {
  "type": "SIGNATURE",
  "digestAlgorithm": "SHA256",
  "signK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-key-cert-secret",
    "key": "occmkey.pem",
    "cert": "occm.cer",
    "extraCerts": []
  }
},
"occmTrustStoreK8sSecretIn": {
  "namespace": "ns1",
  "name": "ca1-occm-trust-store-secret",
  "rootCACerts": ["caroot.cer"],
  "intCACerts": ["intca.cer"],
  "serverCert": ""
},
"tlsConfig": {
  "enableTLS": true,
  "tlsTrustStoreK8sSecretItem": {
    "namespace": "ns1",
    "name": "occm-tls-trust-store-secret",
    "tlsTrustedCerts": ["ManagementCA-chain.pem"]
  }
}
}'

```

Sample POST Response:

```

200 Success Response Body: '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
  "name": "CA-TLS-1",
  "server": "https://thrust5:8445/ejbca/publicweb/cmp/occmalias",

```

```

"recipientDN": "/CN=svc.thrust5",
"issuerDN": "/CN=ManagementCA",
"totalTimeout": "60",
"messageTimeout": "30",
"cmpProtectionOccmCert": {
  "type": null,
  "digestAlgorithm": null,
  "macAlgorithm": null,
  "macK8sSecretIn": {
    "namespace": "",
    "name": "",
    "passKey": "",
    "refKey": ""
  },
  "signK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": "",
    "cert": "",
    "extraCerts": []
  }
},
"cmpProtectionOtherCert": {
  "type": "SIGNATURE",
  "digestAlgorithm": "SHA256",
  "signK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-key-cert-secret",
    "key": "occmkey.pem",
    "cert": "occm.cer",
    "extraCerts": []
  }
},
"occmTrustStoreK8sSecretIn": {
  "namespace": "ns1",
  "name": "ca1-occm-trust-store-secret",
  "rootCACerts": ["caroot.cer"],
  "intCACerts": ["intca.cer"],
  "serverCert": ""
},
"tlsConfig": {
  "enableTLS": true,
  "tlsTrustStoreK8sSecretItem": {
    "namespace": "ns1",
    "name": "occm-tls-trust-store-secret",
    "tlsTrustedCerts": ["ManagementCA-chain.pem"]
  }
}
}'

```

2.1.4 Update Issuer Configurations

OCCM uses the PUT operation to update the issuers configuration using the issuer uuid (in path variable) and request body.

Resource URI: /occm-config/v1/issuers/{uuid}

Update Issuer functionality depends the following:

- No certificate is attached to an issuer: If no certificate is attached to an issuer then updating all fields are allowed.
- Certificate is attached to an issuer: If any certificate is attached to an issuer then updating following fields are allowed:
 - server - Domain URL of CA
 - tlsConfig - TLS Configuration:
 - * enableTLS - When set to true, HTTPS connection to CA is made. Ensure selecting scheme as HTTPS in server URL if this is set to true.
 - * tlsTrustStoreK8sSecretItem - TLS trust store secret details such as namespace - K8s namespace where TLS trust store secret is present
name - Kubernetes secret which holds TLS trust store information (CA certificates)
tlsTrustedCerts - Trusted certificate(s) to use for validating the TLS server certificate.

Data structures supported by the PUT Response Body on this resource

Response codes	Data type	Cardinality	Description
200 OK	Issuers	1	This is a mandatory parameter. Issuers configuration data
400 BAD REQUEST	Problem Details	1	This is a mandatory parameter. Invalid input is passed to process request.

Sample PUT Request when Enable TLS is False:

```
$ curl --location --request PUT 'http://{host}:{port}/occm-config/v1/issuers/4c5b4025-6c63-438c-bcd7-27b5bf8da4fd' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer LxuLeX9dihXDUcoFwDw' \
--header 'Content-Type: application/json' \
--data-raw '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
  "name": "CA1",
  "server": "http://ca1-openssl-mock.ns1.svc.thrust5:8080",
  "recipientDN": "/CN=svc.thrust5",
  "issuerDN": "/CN=svc.thrust5",
  "totalTimeout": "60",
  "messageTimeout": "30",
  "cmpProtectionOccmCert": {
    "type": null,
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "macK8sSecretIn": {
      "namespace": "",
```

```

        "name": "",
        "passKey": "",
        "refKey": ""
    },
    "signK8sSecretIn": {
        "namespace": "",
        "name": "",
        "key": "",
        "cert": "",
        "extraCerts": []
    }
},
"cmpProtectionOtherCert": {
    "type": "SIGNATURE",
    "digestAlgorithm": "SHA256",
    "signK8sSecretIn": {
        "namespace": "ns1",
        "name": "ca1-occm-key-cert-secret",
        "key": "occmkey.pem",
        "cert": "occm.cer",
        "extraCerts": []
    }
},
"occmTrustStoreK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-trust-store-secret",
    "rootCACerts": ["caroot.cer"],
    "intCACerts": ["intca.cer"],
    "serverCert": ""
},
"tlsConfig": {
    "enableTLS": false,
    "tlsTrustStoreK8sSecretItem": {
        "namespace": "",
        "name": "",
        "tlsTrustedCerts": [""]
    }
}
}'

```

Sample PUT Response:

```

200 Success Response Body: '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
  "name": "CA1",
  "server": "http://ca1-openssl-mock.ns1.svc.thrust5:8080",
  "recipientDN": "/CN=svc.thrust5",
  "issuerDN": "/CN=svc.thrust5",
  "totalTimeout": "60",
  "messageTimeout": "30",
  "cmpProtectionOccmCert": {
    "type": null,
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "macK8sSecretIn": {

```

```

        "namespace": "",
        "name": "",
        "passKey": "",
        "refKey": ""
    },
    "signK8sSecretIn": {
        "namespace": "",
        "name": "",
        "key": "",
        "cert": "",
        "extraCerts": []
    }
},
"cmpProtectionOtherCert": {
    "type": "SIGNATURE",
    "digestAlgorithm": "SHA256",
    "signK8sSecretIn": {
        "namespace": "ns1",
        "name": "ca1-occm-key-cert-secret",
        "key": "occmkey.pem",
        "cert": "occm.cer",
        "extraCerts": []
    }
},
"occmTrustStoreK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-trust-store-secret",
    "rootCACerts": ["caroot.cer"],
    "intCACerts": ["intca.cer"],
    "serverCert": ""
},
"tlsConfig": {
    "enableTLS": false,
    "tlsTrustStoreK8sSecretItem": {
        "namespace": "",
        "name": "",
        "tlsTrustedCerts": [""]
    }
}
}'

```

Sample PUR REquest when Enable TLS is True"

```

curl --location --request PUT 'http://{host}:{port}/occm-config/v1/issuers/
4c5b4025-6c63-438c-bcd7-27b5bf8da4fd' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer LxuLeX9dihXDUcoFwDw' \
--header 'Content-Type: application/json' \
--data-raw '{
    "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
    "name": "CA-TLS-1",
    "server": "https://thrust5:8445/ejbca/publicweb/cmp/occmalias",
    "recipientDN": "/CN=svc.thrust5",
    "issuerDN": "/CN=ManagementCA",

```

```

"totalTimeout": "60",
"messageTimeout": "30",
"cmpProtectionOccmCert": {
  "type": null,
  "digestAlgorithm": null,
  "macAlgorithm": null,
  "macK8sSecretIn": {
    "namespace": "",
    "name": "",
    "passKey": "",
    "refKey": ""
  },
  "signK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": "",
    "cert": "",
    "extraCerts": []
  }
},
"cmpProtectionOtherCert": {
  "type": "SIGNATURE",
  "digestAlgorithm": "SHA256",
  "signK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-key-cert-secret",
    "key": "occmkey.pem",
    "cert": "occm.cer",
    "extraCerts": []
  }
},
"occmTrustStoreK8sSecretIn": {
  "namespace": "ns1",
  "name": "ca1-occm-trust-store-secret",
  "rootCACerts": ["caroot.cer"],
  "intCACerts": ["intca.cer"],
  "serverCert": ""
},
"tlsConfig": {
  "enableTLS": true,
  "tlsTrustStoreK8sSecretItem": {
    "namespace": "ns1",
    "name": "occm-tls-trust-store-secret",
    "tlsTrustedCerts": ["ManagementCA-chain.pem"]
  }
}
}'

```

Sample PUT Response:

```

200 Success Response Body: '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",
  "name": "CA-TLS-1",
  "server": "https://thrust5:8445/ejbca/publicweb/cmp/occmalias",
  "recipientDN": "/CN=svc.thrust5",

```

```

"issuerDN": "/CN=ManagementCA",
"totalTimeout": "60",
"messageTimeout": "30",
"cmpProtectionOccmCert": {
  "type": null,
  "digestAlgorithm": null,
  "macAlgorithm": null,
  "macK8sSecretIn": {
    "namespace": "",
    "name": "",
    "passKey": "",
    "refKey": ""
  },
  "signK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": "",
    "cert": "",
    "extraCerts": []
  }
},
"cmpProtectionOtherCert": {
  "type": "SIGNATURE",
  "digestAlgorithm": "SHA256",
  "signK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-key-cert-secret",
    "key": "occmkey.pem",
    "cert": "occm.cer",
    "extraCerts": []
  }
},
"occmTrustStoreK8sSecretIn": {
  "namespace": "ns1",
  "name": "ca1-occm-trust-store-secret",
  "rootCACerts": ["caroot.cer"],
  "intCACerts": ["intca.cer"],
  "serverCert": ""
},
"tlsConfig": {
  "enableTLS": true,
  "tlsTrustStoreK8sSecretItem": {
    "namespace": "ns1",
    "name": "occm-tls-trust-store-secret",
    "tlsTrustedCerts": ["ManagementCA-chain.pem"]
  }
}
}'

```

2.1.5 Delete Issuers Data

OCCM uses the DELETE operation to delete the Issuers data based on query parameters.

Note

An issuer can only be deleted if there are no certificates referring to this issuer entry.

Resource URI: /occm-config/v1/issuers/{uuid}

Data structures supported by the DELETE Response Body on this resource

Response codes	Data type	Cardinality	Description
200 OK	Issuers	1	This is a mandatory parameter. Issuers configuration data
400 Bad Request	Problem Details	1	This is a mandatory parameter. Input does not match to process request

Sample DELETE Request:

```
$ curl --location --request DELETE 'http://{host}:{port}/occm-config/v1/issuers/4c5b4025-6c63-438c-bcd7-27b5bf8da4fd' \
--header 'oc-cncc-id: Cluster1' \
--header 'oc-cncc-instance-id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIs.....'
```

Sample DELETE Response:

```
200 OK Response Body: '{
  "uuid": "4c5b4025-6c63-438c-bcd7-27b5bf8da4fd",      {
  "name": "CA1",
  "server": "http://ca1-openssl-mock.ns1.svc.thrust5:8080",
  "recipientDN": "/CN=svc.thrust5",
  "issuerDN": "/CN=svc.thrust5",
  "totalTimeout": "60",
  "messageTimeout": "30",
  "cmpProtectionOcmCert": {
    "type": null,
    "digestAlgorithm": null,
    "macAlgorithm": null,
    "macK8sSecretIn": {
      "namespace": "",
      "name": "",
      "passKey": "",
      "refKey": ""
    },
    "signK8sSecretIn": {
      "namespace": "",
      "name": "",
      "key": "",
      "cert": "",
      "extraCerts": []
    }
  }
}
```



```

    }
  },
  "cmpProtectionOtherCert": {
    "type": "SIGNATURE",
    "digestAlgorithm": "SHA256",
    "signK8sSecretIn": {
      "namespace": "ns1",
      "name": "ca1-occm-key-cert-secret",
      "key": "occmkey.pem",
      "cert": "occm.cer",
      "extraCerts": []
    }
  },
  "occmTrustStoreK8sSecretIn": {
    "namespace": "ns1",
    "name": "ca1-occm-trust-store-secret",
    "rootCACerts": ["caroot.cer"],
    "intCACerts": ["intca.cer"],
    "serverCert": ""
  },
  "tlsConfig": {
    "enableTLS": false,
    "tlsTrustStoreK8sSecretItem": {
      "namespace": "",
      "name": "",
      "tlsTrustedCerts": [""]
    }
  }
}

```

2.2 OCCM Certificates

OCCM Certificates Data Model

Table 2-3 OCCM Certifiactes Request Parameters

Field Name	Data Type	Description
name	String	This is a mandatory parameter. Name of the certificate
lcmType	Enum	This is a mandatory parameter. Possible Values: AUTOMATIC, MANUAL
certType	Enum	This is a mandatory parameter. Possible Values: OCCM, OTHER
renewBefore	String	This is an optional parameter. Number of days before the certificate expiry, when the certificate will be renewed. Default Value 14 Days Min: 1 days Max: [(validity i.e csr.days)-1] days

Table 2-3 (Cont.) OCCM Certificates Request Parameters

Field Name	DataType	Description
certPurpose	String	This is an optional parameter. Purpose of certificate creation
issuer	String	This is a mandatory parameter. Name of CA
privateKey	Object	This is a mandatory parameter. Private key details like algorithm, key size and key encoding
privateKey.keyAlgo	Enum	This is a mandatory parameter. Private key algorithm to be used. Supported values: RSA and EC Possible values: RSA, EC
privateKey.keySize	Enum	This is an optional parameter. The number of bits in the generated key. Need to select a bit length of at least 2048 when using RSA and 256 when using ECDSA. These are the smallest key sizes allowed for SSL certificates. Possible values: KEYSIZE_2048, KEYSIZE_4096 Default Value for RSA Key: 2048 bits
privateKey.keyEncoding	Enum	This is an optional parameter. The output format of a private key input source. Default Value: PEM Possible values: PEM, DER
privateKey.ecCurve	Enum	This is an optional parameter. The EC curve to use if the key algorithm selected is EC. Default Value: SECP384r1 Possible values: SECP256r1, SECP384r1
privateKey.keyFormat	String	This is a mandatory parameter. The output format of a private key input source. Default Value: PEM
privateKey.privateKeyK8sSecretOutput	Object	This is a mandatory parameter except in case of OCCM certificates, this field is optional since it is auto-populated from issuer Private key output location
privateKey.privateKeyK8sSecretOutput.namespace	String	This is a mandatory parameter except in case of OCCM certificates, this field is optional since it is auto-populated from issuer. Kubernetes namespace

Table 2-3 (Cont.) OCCM Certificates Request Parameters

Field Name	Data Type	Description
privateKey.privateKeyK8sSecretOut.name	String	This is a mandatory parameter except in case of OCCM certificates, this field is optional since it is auto-populated from issuer. Kubernetes secret name
privateKey.privateKeyK8sSecretOut.key	String	This is a mandatory parameter except in case of OCCM certificates, this field is optional since it is auto-populated from issuer. Kubernetes secret key against which the key-pair will be stored.
csr	Object	This is a mandatory parameter. Certificate Signing Request data
csr.extendedKeyUsage	Object	This is a mandatory parameter. A multi-valued certificate extension containing a list of values indicating purposes for which the certificate public key can be used
csr.extendedKeyUsage.critical	Boolean	This is an optional parameter. When set to true, extended key usage extension will be marked as critical. Default Value: false Possible values: true false
csr.extendedKeyUsage.extendedKeyUsageValues	List<Enum>	This is a mandatory parameter. List of extendedKeyUsage values Default Value(s): CLIENT_AUTH, SERVER_AUTH Possible values: CLIENT_AUTH, SERVER_AUTH
csr.keyUsage	Object	This is a mandatory parameter. A multi-valued certificate extension containing a list of names of the permitted key usages.
csr.keyUsage.critical	Boolean	This is an optional parameter. When set to true, key usage extension will be marked as critical. Possible values: true false

Table 2-3 (Cont.) OCCM Certificates Request Parameters

Field Name	Data Type	Description
csr.keyUsage.keyUsageValues	List<Enum>	This is a mandatory parameter. List of keyUsage values Default Value: DIGITAL_SIGNATURE Possible values: DIGITAL_SIGNATURE, KEY_ENCIPHERMENT, NON_REPUDIATION
csr.basicConstraints	Object	This is an optional parameter. This is a multi-valued extension which indicates whether a certificate is a CA certificate. The first value is CA followed by TRUE or FALSE
csr.basicConstraints.critical	Boolean	This is an optional parameter. When set to true, basicConstraints extension will be marked as critical. Possible values: true false
csr.basicConstraints.basicConstraintsValue	Enum	This is an optional parameter. BasicConstraints value Default Value: END_ENTITY Possible values: END_ENTITY
csr.subject	Object	This is an optional parameter except if csr.subjectAltName is not provided. Information about company
csr.subject.country	String	This is an optional parameter. Country code where company is legally located.
csr.subject.state	String	This is an optional parameter. State where company is legally located.
csr.subject.location	String	This is an optional parameter. The city or town where company is legally located.
csr.subject.organization	String	This is an optional parameter. Your company's legally registered name.
csr.subject.organizationUnit	String	This is an optional parameter. Name of your department within the organization.

Table 2-3 (Cont.) OCCM Certificates Request Parameters

Field Name	Data Type	Description
csr.subject.commonName	String	This is an optional parameter. The Common Name (AKA CN) represents the server name to be protected by the SSL certificate. The certificate is valid only if the request hostname matches the certificate common name.
csr.days.	String	This is a mandatory parameter. Requested validity for the certificate i.e. Number of days requested for which the certificate will be valid. Default Value :365 Days Max Value: 1065 Min Value: 2 Days
csr.subjectAltName	Object	This is an optional parameter except if csr.subject is not provided. A multi-valued extension indicating all of the domain names, IP addresses, URIs etc that are secured by the certificate.
csr.subjectAltName.critical	Boolean	This is an optional parameter. When set to true, subjectAltName extension will be marked as critical. Default Value: true Possible values: true false
csr.subjectAltName.ipAddress	List<String>	This is an optional parameter. List of IP addresses.
csr.subjectAltName.dns	List<String>	This is an optional parameter. List of domain names
csr.subjectAltName.uriIdUrn	List<String>	This is an optional parameter. List of URI ID (URN of the NFInstanceId)
csr.subjectAltName.uriIdApiRoot	List<String>	This is an optional parameter. List of uniform resource locator IDs
csr.certK8sSecretOut	Object	This is a mandatory parameter except in case of OCCM certificate, this field is optional since it is auto-populated from issuer. Certificate output location.

Table 2-3 (Cont.) OCCM Certificates Request Parameters

Field Name	Data Type	Description
csr.certK8sSecretOut.namespace	String	This is a mandatory parameter except in case of OCCM certificate, this field is optional since it is auto-populated from issuer.
csr.certK8sSecretOut.name	String	Kubernetes secret namespace This is a mandatory parameter except in case of OCCM certificate, this field is optional since it is auto-populated from issuer..
csr.certK8sSecretOut.key	String	Kubernetes secret name This is a mandatory parameter except in case of OCCM certificate, this field is optional since it is auto-populated from issuer..
csr.certChainK8sSecretOut	Object	Kubernetes secret key against which the certificate will be stored. This is a mandatory parameter.
csr.certChainK8sSecretOut.namespace	String	Certificate Chain output location. This is an optional parameter.
csr.certChainK8sSecretOut.name	String	Kubernetes secret namespace This is a mandatory parameter.
csr.certChainK8sSecretOut.key	String	Kubernetes secret name This is an optional parameter.
nf	String	Kubernetes secret key against which the certificate chain will be stored. This is a mandatory parameter.
uuid	String	NF name
overrideSecret	boolean	Unique id for logging and tracking purpose This is an optional parameter.
		This flag is used to override the Kubernetes secret with new certificate.
		Default Value: false
		Possible values:
		true
		false
caBundleK8sSecretIn	Object	This is an optional parameter.
		CA bundle secret input details.
		Used to trust peer entities.
caBundleK8sSecretIn.namespace	String	This is an optional parameter.
		Kubernetes secret namespace
caBundleK8sSecretIn.name	String	This is an optional parameter.
		Kubernetes secret name

Table 2-3 (Cont.) OCCM Certificates Request Parameters

Field Name	Data Type	Description
caBundleK8sSecretIn.key	String	This is an optional parameter. Kubernetes secret key against which CA bundle certificate(s) will be stored.

Table 2-4 OCCM Certificate Response Codes

Response codes	Data type	Cardinality	Description
200 OK	Object (Certs) Or List(Certs)	1	This is a mandatory parameter. Object Certs Or List (CertConfig) matching criteria
201 CREATED	Object (Certs)	1	This is a mandatory parameter. Object Certs
202 Accepted	String	1	This is a mandatory parameter. Return uuid
400 Bad request	Problem Details	1	This is a mandatory parameter. Input does not match to process request
500 Internal Server Error	Problem Details	1	This is a mandatory parameter. Something went wrong
409 Conflict	Problem Details	1	This is a mandatory parameter. Record already exists

Note

OCCM Certificates response body data model varies based on REST operation status.

OCCM Certificate JSON payload

```
{
  "name": "",
  "lcmType": "",
  "certType": "",
  "renewBefore": "",
  "certPurpose": "",
  "issuer": "",
  "privateKey": {
    "keyAlgo": "",
    "keySize": "",
```

```

        "keyEncoding": "",
        "ecCurve": "",
        "privateKeyK8sSecretOut": {
            "namespace": "",
            "name": "",
            "key": ""
        }
    },
    "csr": {
        "extendedKeyUsage": {
            "critical": "",
            "extendedKeyUsageValues": []
        },
        "keyUsage": {
            "critical": "",
            "keyUsageValues": []
        }
    },
    "basicConstraints": {
        "critical": true,
        "basicConstraintsValue": ""
    },
    "subject": {
        "country": "",
        "state": "",
        "location": "",
        "organization": "",
        "organizationUnit": "",
        "commonName": ""
    },
    "days": "",
    "subjectAltName": {
        "critical": "",
        "ipAddress": [],
        "dns": [],
        "uriIdUrn": [],
        "uriIdApiRoot": []
    },
    "certK8sSecretOut": {
        "namespace": "",
        "name": "",
        "key": ""
    },
    "certChainK8sSecretOut": {
        "namespace": "",
        "name": "",
        "key": ""
    }
},

"caBundleK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": ""
},

```



```

    "nf": "",
    "overrideSecret": false
  }

```

2.2.1 Fetch all Certificate Configurations

OCCM Uses the GET operation to fetch all the certificate configurations.

Resource URI: /occm-config/v1/certs

Data structures supported by the GET Response Body on this resource

Response codes	Data type	Cardinality	Description
200 OK	Object (CertConfig) Or List(CertConfig)	1	This is a mandatory parameter. List (CertConfig) matching criteria
500 Internal Server Error	Problem Details	1	This is a mandatory parameter. Something went wrong

Sample GET Request:

```

\ $ curl --location --request GET 'http://{host}:{port}/occm-config/v1/certs' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSU.....'

```

Sample GET Response:

```

200 OK Response Body: '[
  {
    "uuid": "b4d896ac-689d-4e12-a76c-54c8de4ffe52",
    "name": "NRFTLS12",
    "lcmType": "AUTOMATIC",
    "certType": "OTHER",
    "renewBefore": "7",
    "certPurpose": "NRF SBI",
    "issuer": "CA21",
    "privateKey": {
      "keyAlgo": "RSA",
      "keySize": "KEYSIZE_2048",
      "keyEncoding": "PEM",
      "ecCurve": "",
      "privateKeyK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrf.pem"
      }
    },
    "csr": {
      "extendedKeyUsage": {
        "critical": false,

```

```

        "extendedKeyUsageValues" : [
            "CLIENT_AUTH",
            "SERVER_AUTH"
        ],
    },
    "keyUsage": {
        "critical" : true,
        "keyUsageValues" : [
            "DIGITAL_SIGNATURE"
        ]
    },
    "basicConstraints": {
        "critical" : true,
        "basicConstraintsValue" : "END_ENTITY"
    },
    "subject": {
        "country": "IN",
        "state": "Karnataka",
        "location": "Bengaluru",
        "organization": "Oracle",
        "organizationUnit": "OracleBU",
        "commonName": "some.example.com"
    },
    "days": "365",
    "subjectAltName": {
        "critical" : null,
        "ipAddress": [
            "10.10.10.13",
            "10.10.10.14"
        ],
        "dns": [
            "centos8-2.example.com",
            "centos8-3.example.com"
        ],
        "uriIdUrn": [
            "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
        ],
        "uriIdApiRoot": [
        ]
    },
    "certK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrf.cer"
    },
    "certChainK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrfcertchain.cer"
    },
    "caBundleK8sSecretIn": {
        "namespace": "occm",

```

```

        "name": "nrf-cabu",
        "key": "cabundle.cer"
    },

    "nf": "NRF",
    "overrideSecret": false
} ]'

```

2.2.2 Fetch Certificate Configurations by UUID

OCCM Uses the GET operation to fetch the certificate configurations details by uuid.

Resource URI: /occm-config/v1/certs/{uuid}

Data structures supported by the GET Response Body on this resource

Response codes	Data type	Cardinality	Description
404 Not Found	Problem Details	1	This is a mandatory parameter. Input does not match to process request
200 OK	Object (Certs)	1	This is a mandatory parameter. Object (CertConfig) matching criteria
500 Internal Server Error	Problem Details	1	This is a mandatory parameter. Something went wrong
400 Bad Request	Problem Details	1	This is a mandatory parameter. wrong Input

Sample Get Request:

```

$ curl --location --request GET 'http://{host}:{port}/occm-config/v1/certs/b4d896ac-689d-4e12-a76c-54c8de4ffe52' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSU...'

```

Sample Get Response:

```

200 OK Response Body: '{
  "uuid": "b4d896ac-689d-4e12-a76c-54c8de4ffe52",
  "name": "NRFTLS12",
  "lcmType": "AUTOMATIC",
  "certType": "OTHER",
  "renewBefore": "7",
  "certPurpose": "NRF SBI",
  "issuer": "CA21",
  "privateKey": {
    "keyAlgo": "RSA",
    "keySize": "KEYSIZE_2048",

```

```

        "keyEncoding": "PEM",
        "ecCurve": "",
        "privateKeyK8sSecretOut": {
            "namespace": "occm",
            "name": "nrf-tls-secret-54",
            "key": "nrf.pem"
        }
    },
    "csr": {
        "extendedKeyUsage": {
            "critical": false,
            "extendedKeyUsageValues": [
                "CLIENT_AUTH",
                "SERVER_AUTH"
            ]
        },
        "keyUsage": {
            "critical": true,
            "keyUsageValues": [
                "DIGITAL_SIGNATURE"
            ]
        }
    },
    "basicConstraints": {
        "critical": true,
        "basicConstraintsValue": "END_ENTITY"
    },
    "subject": {
        "country": "IN",
        "state": "Karnataka",
        "location": "Bengaluru",
        "organization": "Oracle",
        "organizationUnit": "OracleBU",
        "commonName": "some.example.com"
    },
    "days": "365",
    "subjectAltName": {
        "critical": null,
        "ipAddress": [
            "10.10.10.13",
            "10.10.10.14"
        ],
        "dns": [
            "centos8-2.example.com",
            "centos8-3.example.com"
        ],
        "uriIdUrn": [
            "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
        ],
        "uriIdApiRoot": [
        ]
    },
    "certK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
    }
}

```

```

        "key": "nrf.cer"
    },
    "certChainK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrfcertchain.cer"
    }
},

    "caBundleK8sSecretIn": {
        "namespace": "occm",
        "name": "nrf-cabu",
        "key": "cabundle.cer"
    },

    "nf": "NRF",
    "overrideSecret": false
}'

```

2.2.3 Add Certificate Configurations

OCCM uses the POST operation to add the certificate configuration using the Request Body.

Resource URI: /occm-config/v1/certs

Data structures supported by the POST Response Body on this resource

Response codes	Data type	Cardinality	Description
202 Accepted	String	1	This is a mandatory parameter. Unique identification of certs
400 BAD REQUEST	Problem Details	1	This is a mandatory parameter. Returns Problem Details structure as defined in 3GPP TS 29.571 section 5.2.4.1
409 CONFLICT	Problem Details	1	This is a mandatory parameter. Record already exists

Sample POST request:

```

$ curl --location --request POST 'http://{host}:{port}/occm-config/v1/certs' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1Ni.....' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "NRFTLS12",
    "lcmType": "AUTOMATIC",
    "certType": "OTHER",
    "renewBefore": "7",

```

```

    "certPurpose": "NRF SBI",
    "issuer": "CA21",
    "privateKey": {
      "keyAlgo": "RSA",
      "keySize": "KEYSIZE_2048",
      "keyEncoding": "PEM",
      "ecCurve": "",
      "privateKeyK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrf.pem"
      }
    },
    "csr": {
      "extendedKeyUsage": {
        "critical" : false,
        "extendedKeyUsageValues" : [
          "CLIENT_AUTH",
          "SERVER_AUTH"
        ]
      },
      "keyUsage": {
        "critical" : true,
        "keyUsageValues" : [
          "DIGITAL_SIGNATURE"
        ]
      }
    },
    "basicConstraints": {
      "critical" : true,
      "basicConstraintsValue" : "END_ENTITY"
    },
    "subject": {
      "country": "IN",
      "state": "Karnataka",
      "location": "Bengaluru",
      "organization": "Oracle",
      "organizationUnit": "OracleBU",
      "commonName": "some.example.com"
    },
    "days": "365",
    "subjectAltName": {
      "critical" : null,
      "ipAddress": [
        "10.10.10.13",
        "10.10.10.14"
      ],
      "dns": [
        "centos8-2.example.com",
        "centos8-3.example.com"
      ],
      "uriIdUrn": [
        "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
      ],
      "uriIdApiRoot": [
      ]
    }
  }

```

```

    },
    "certK8sSecretOut": {
      "namespace": "occm",
      "name": "nrf-tls-secret-54",
      "key": "nrf.cer"
    },
    "certChainK8sSecretOut": {
      "namespace": "occm",
      "name": "nrf-tls-secret-54",
      "key": "nrfcertchain.cer"
    }
  },

  "caBundleK8sSecretIn": {
    "namespace": "occm",
    "name": "nrf-cabu",
    "key": "cabundle.cer"
  },

  "nf": "NRF",
  "overrideSecret": false
}'

```

Sample POST Response

```

$ 202 Accepted Response Body: '{
  "uuid": "b4d896ac-689d-4e12-a76c-54c8de4ffe52",
  "name": "NRFTLS12",
  "lcmType": "AUTOMATIC",
  "certType": "OTHER",
  "renewBefore": "7",
  "certPurpose": "NRF SBI",
  "issuer": "CA21",
  "privateKey": {
    "keyAlgo": "RSA",
    "keySize": "KEYSIZE_2048",
    "keyEncoding": "PEM",
    "ecCurve": "",
    "privateKeyK8sSecretOut": {
      "namespace": "occm",
      "name": "nrf-tls-secret-54",
      "key": "nrf.pem"
    }
  },
  "csr": {
    "extendedKeyUsage": {
      "critical": false,
      "extendedKeyUsageValues": [
        "CLIENT_AUTH",
        "SERVER_AUTH"
      ]
    },
    "keyUsage": {
      "critical": true,
      "keyUsageValues": [

```

```

        "DIGITAL_SIGNATURE"
    ],
    },
    "basicConstraints": {
        "critical" : true,
        "basicConstraintsValue" : "END_ENTITY"
    },
    "subject": {
        "country": "IN",
        "state": "Karnataka",
        "location": "Bengaluru",
        "organization": "Oracle",
        "organizationUnit": "OracleBU",
        "commonName": "some.example.com"
    },
    "days": "365",
    "subjectAltName": {
        "critical" : null,
        "ipAddress": [
            "10.10.10.13",
            "10.10.10.14"
        ],
        "dns": [
            "centos8-2.example.com",
            "centos8-3.example.com"
        ],
        "uriIdUrn": [
            "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
        ],
        "uriIdApiRoot": [
        ]
    },
    "certK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrf.cer"
    },
    "certChainK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrfcertchain.cer"
    }
},

"caBundleK8sSecretIn": {
    "namespace": "occm",
    "name": "nrf-cabu",
    "key": "cabundle.cer"
},

"nf": "NRF",
"overrideSecret": false
}'

```


2.2.4 Delete Certificate Configuration Data

OCCM uses the DELETE operation to delete the certificate configuration data based on query parameters.

Resource URI: /occm-config/v1/certs/{uuid}

Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
Cert	M	1	200 OK	This is a mandatory parameter. Certificate configuration data
Problem Details	M	1	400 Bad Request	This is a mandatory parameter. Input does not match to process request

Sample DELETE Request:

```
$ curl --location --request DELETE 'http://{host}:{port}/occm-config/v1/certs/
b4d896ac-689d-4e12-a76c-54c8de4ffe52' \
--header 'Oc-Cncc-Id: Cluster1' \
--header 'Oc-Cncc-Instance-Id: Cluster1-OCCM-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1Ni...'
```

Sample Response:

```
200 OK Response Body: '{
  "uuid": "b4d896ac-689d-4e12-a76c-54c8de4ffe52",
  "name": "NRFTLS12",
  "lcmType": "AUTOMATIC",
  "certType": "OTHER",
  "renewBefore": "7",
  "certPurpose": "NRF SBI",
  "issuer": "CA21",
  "privateKey": {
    "keyAlgo": "RSA",
    "keySize": "KEYSIZE_2048",
    "keyEncoding": "PEM",
    "ecCurve": "",
    "privateKeyK8sSecretOut": {
      "namespace": "occm",
      "name": "nrf-tls-secret-54",
      "key": "nrf.pem"
    }
  },
  "csr": {
    "extendedKeyUsage": {
      "critical": false,
      "extendedKeyUsageValues": [
```

```

        "CLIENT_AUTH",
        "SERVER_AUTH"
    ],
    "keyUsage": {
        "critical" : true,
        "keyUsageValues" : [
            "DIGITAL_SIGNATURE"
        ]
    },
    "basicConstraints": {
        "critical" : true,
        "basicConstraintsValue" : "END_ENTITY"
    },
    "subject": {
        "country": "IN",
        "state": "Karnataka",
        "location": "Bengaluru",
        "organization": "Oracle",
        "organizationUnit": "OracleBU",
        "commonName": "some.example.com"
    },
    "days": "365",
    "subjectAltName": {
        "critical" : null,
        "ipAddress": [
            "10.10.10.13",
            "10.10.10.14"
        ],
        "dns": [
            "centos8-2.example.com",
            "centos8-3.example.com"
        ],
        "uriIdUrn": [
            "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
        ],
        "uriIdApiRoot": [
        ]
    },
    "certK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrf.cer"
    },
    "certChainK8sSecretOut": {
        "namespace": "occm",
        "name": "nrf-tls-secret-54",
        "key": "nrfcertchain.cer"
    },
    "caBundleK8sSecretIn": {
        "namespace": "occm",
        "name": "nrf-cabu",
        "key": "cabundle.cer"
    }
}

```

```

    },
    "nf": "NRF",
    "overrideSecret": false
  }'

```

2.2.5 Recreate Certificates

OCCM uses the PUT operation to recreate certificates using existing certificate configurations. You can recreate any certificate that is in ready or expired status. This enhances OCCM's usability in managing certificate lifecycle operations. For example, if a certificate has been deleted, revoked or has expired, the operator can recreate it using existing configurations. The certificate configuration must exist in OCCM while triggering recreate request.

Resource URI: /occm-config/v1/certs/{uuid}

Table 2-5 Path Variable

Name	Data type	P	Cardinality	Description
uuid	String	M	1	Certificate uuid

Data structures supported by the PUT Response Body on this resource

Response codes	Data type	Cardinality	Description
202 Accepted	String	1	This is a mandatory parameter. Unique Certificate Identity.
400 Bad Request	Problem Details	1	This is a mandatory parameter. Returns Problem Details structure as defined in 3GPP TS 29.571 section 5.2.4.1.

Sample PUT Request:

```

curl --location --request PUT 'http://{host}:{port}/occm-config/v1/certs/
9983d728-c618-41c3-b7f9-530d00fb7ab1' \
--header 'oc-cncc-id: Cluster1' \
--header 'oc-cncc-instance-id: Cluster1-occm-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cGE6YW50IjoiOiB1c2VudC5UfCAQ' \
--header 'Content-Type: application/json' \
--data-raw '{
  "uuid": "9983d728-c618-41c3-b7f9-530d00fb7ab1",
  "name": "NRF-TLS-3",
  "lcmType": "AUTOMATIC",
  "certType": "OTHER",
  "renewBefore": "15",
  "certPurpose": "NRF SBI",
  "issuer": "EJBCA-HTTP-1",
  "privateKey": {
    "keyAlgo": "EC",

```

```

    "keySize": null,
    "keyEncoding": "DER",
    "ecCurve": "SECP256r1",
    "keyFormat": null,
    "privateKeyK8sSecretOut": {
      "namespace": "occncc-thrust5-02",
      "name": "nrf-tls-secret250403",
      "key": "nrfkey.pem"
    }
  },
  "csr": {
    "extendedKeyUsage": {
      "critical": false,
      "extendedKeyUsageValues": [
        "CLIENT_AUTH",
        "SERVER_AUTH"
      ]
    },
    "keyUsage": {
      "critical": false,
      "keyUsageValues": [
        "DIGITAL_SIGNATURE",
        "KEY_ENCIIPHERMENT"
      ]
    },
    "basicConstraints": {
      "critical": false,
      "basicConstraintsValue": "END_ENTITY"
    },
    "subject": {
      "country": "IN",
      "state": "KA",
      "location": "BLR",
      "organization": "Oracle",
      "organizationUnit": "CGBU",
      "commonName": "occm"
    },
    "days": "365",
    "subjectAltName": {
      "critical": false,
      "ipAddress": [
        "10.10.10.13",
        "10.10.10.14"
      ],
      "dns": [
        "centos8-2.example.com",
        "centos8-3.example.com"
      ],
      "uriIdApiRoot": null,
      "uriIdUrn": [
        "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
      ]
    },
    "certK8sSecretOut": {
      "namespace": "occncc-thrust5-02",
      "name": "nrf-tls-secret250403",

```

```

        "key": "nrf.cer"
      },
      "certChainK8sSecretOut": {
        "namespace": "occncc-thrust5-02",
        "name": "nrf-tls-secret250403",
        "key": "nrfcertchain.cer"
      }
    },
    "caBundleK8sSecretIn": {
      "namespace": "",
      "name": "",
      "key": ""
    },
    "nf": "NRF",
    "overrideSecret": true
  }'

```

Sample Response:

```

202 Accepted Response Body: '{
  "uuid": "9983d728-c618-41c3-b7f9-530d00fb7ab1",
  "name": "NRF-TLS-3",
  "lcmType": "AUTOMATIC",
  "certType": "OTHER",
  "renewBefore": "15",
  "certPurpose": "NRF_SBI",
  "issuer": "EJBCA-HTTP-1",
  "privateKey": {
    "keyAlgo": "EC",
    "keySize": null,
    "keyEncoding": "DER",
    "ecCurve": "SECP256r1",
    "keyFormat": null,
    "privateKeyK8sSecretOut": {
      "namespace": "occncc-thrust5-02",
      "name": "nrf-tls-secret250403",
      "key": "nrfkey.pem"
    }
  },
  "csr": {
    "extendedKeyUsage": {
      "critical": false,
      "extendedKeyUsageValues": [
        "CLIENT_AUTH",
        "SERVER_AUTH"
      ]
    },
    "keyUsage": {
      "critical": false,
      "keyUsageValues": [
        "DIGITAL_SIGNATURE",
        "KEY_ENCIPHERMENT"
      ]
    },
    "basicConstraints": {

```

```

        "critical": false,
        "basicConstraintsValue": "END_ENTITY"
    },
    "subject": {
        "country": "IN",
        "state": "KA",
        "location": "BLR",
        "organization": "Oracle",
        "organizationUnit": "CGBU",
        "commonName": "occm"
    },
    "days": "365",
    "subjectAltName": {
        "critical": false,
        "ipAddress": [
            "10.10.10.13",
            "10.10.10.14"
        ],
        "dns": [
            "centos8-2.example.com",
            "centos8-3.example.com"
        ],
        "uriIdApiRoot": null,
        "uriIdUrn": [
            "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
        ]
    },
    "certK8sSecretOut": {
        "namespace": "occncc-thrust5-02",
        "name": "nrf-tls-secret250403",
        "key": "nrf.cer"
    },
    "certChainK8sSecretOut": {
        "namespace": "occncc-thrust5-02",
        "name": "nrf-tls-secret250403",
        "key": "nrfcertchain.cer"
    }
},
"caBundleK8sSecretIn": {
    "namespace": "",
    "name": "",
    "key": ""
},
"nf": "NRF",
"overrideSecret": true
}'

```

2.3 OCCM Logging Resource

2.3.1 Fetch Logging Configuration for a Service

OCCM uses the GET operation to fetch the logging configuration for a service

Resource URI: `/occm-config/v1/occm/logging`

Sample GET request:

```
curl --location --request GET 'http://10.121.30.39:30777/occm-config/v1/occm/
logging' \
--header 'oc-cncc-id: Cluster1' \
--header 'oc-cncc-instance-id: Cluster1-occm-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiI...' \
--data-raw ''
```

Sample Response:

```
{"appLogLevel":"INFO", "packageLogLevel":
[{"packageName":"root", "logLevelForPackage":"ERROR"}]}
```

2.3.2 Fetch Logging Configurations for All Services

OCCM uses the GET operation to fetch logging configurations for all services.

Resource URI:/occm-config/v1/all/logging

Sample GET Request:

```
curl --location --request GET 'http://{host}:{port}/occm-config/v1/all/
logging' \
--header 'oc-cncc-id: Cluster1' \
--header 'oc-cncc-instance-id: Cluster1-occm-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAi...' \
```

Sample Response:

```
[{"occm":{"appLogLevel":"INFO","packageLogLevel":
[{"packageName":"root","logLevelForPackage":"ERROR"}]}}]
```

2.3.3 Update Logging Configurations for a Service

OCCM uses the PUT operation to update logging configurations for a service.

Resource URI:/occm-config/v1/occm/logging

Sample PUT Request:

```
curl --location --request PUT 'http://{host}:{port}/occm-config/v1/occm/
logging' \
--header 'oc-cncc-id: Cluster1' \
--header 'oc-cncc-instance-id: Cluster1-occm-instance1' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAi...' \
--header 'Content-Type: application/json' \
--data-raw '{
  "appLogLevel":"INFO",
  "packageLogLevel":[
    {
      "packageName":"root",
      "logLevelForPackage":"ERROR"
    }
  ]
}'
```

```
]
}'
```

Sample Response:

201 Created