Web Services API Reference

Platform Configuration API

4/30/2025

Platform Configuration API

Contents

Overview

You can use the Platform Configuration API to get read and write access to low-level data from Configuration Server. See Introduction to the Configuration Layer Objects for a full list of the supported configuration objects.

Accessing the API

The following table provides basic information about accessing the Platform Configuration API:

Root request URL	/api/v2/platform/configuration
HTTP Methods	GET, POST, PUT, DELETE
Required Features	• api-platform-configuration-read — for read operations • api-platform-configuration-write — for write operations

All configuration objects are retrieved directly from Configuration Server by using the following URI naming format: the plural version of the object name in lower case, with each word separated by a hyphen (-). Here are a few examples:

Configuration object	URI name
CfgAccessGroup	access-groups
CfgCampaign	campaigns
CfgAgentLogin	agent-logins
CfgSkill	skills

The API also supports the standard CfgQuery attributes for an object as parameters on the URI. If you filter the returned resources using these parameters, you can enclose the filter values in quotes to force a "string" type, where applicable. If the filter value doesn't have quotes and is numeric, Web Services uses an integer filter type. You should be aware of the filter types Configuration Server expects and use quotes as needed. See List of Configuration Layer Objects for details about configuration objects and their filters.

For example, the CfgApplication object has a `name` attribute (a string) and a `dbid` attribute (an integer).

Here's how you could query by application name, using quotes around the string filter value:

Request

```
GET http://198.51.100.10:8080/api/v2/platform/configuration/applications?name="SomeName"
```

And here's how you could query by dbid, without quotes around the integer filter value:

Request

```
GET http://198.51.100.10:8080/api/v2/platform/configuration/applications?dbid=123
```

Security

All Platform Configuration API operations are performed with the credentials of the currently logged in user. If a user doesn't have permissions to access a given object or to perform a given operation, Web Services returns an error. If you retrieve object lists through a query, Web Services only returns the set of objects that are accessible to the currently logged in user.

GET Operations

You can retrieve lists of all objects in a contact center for a given object type. For example, you can get all the CfgAgentLogin objects:

Request

```
GET http://198.51.100.10:8080/api/v2/platform/configuration/agent-logins
```

Response


```
{
   "statusCode":0,
   "agent-logins":[
      {
         "useOverride":"2",
         "tenantDBID":"1",
         "DBID":"261",
         "switchSpecificType":"1",
         "userProperties":{
            "TServer":{
               "wrap-up-time":"0"
            }
         },
         "state":"1",
         "switchDBID":"101",
         "loginCode":"111"
      },
      {
         "useOverride":"2",
         "tenantDBID":"1",
         "DBID":"263",
         "switchSpecificType":"1",
         "userProperties":{
            "provisioning_flags":{
               "modified_At":"4ac7b1ad-ac79-4f7b-a082-317ea79fc667"
            },
            "TServer":{
               "wrap-up-time":"0"
            }
         },
         "state":"1",
         "switchDBID":"101",
```

```
         "loginCode":"123456789"
      }
   ]
}
```

Or you can get a specific object by a database ID. For example, let's get the CfgAgentLogin with a DBID of 261:

Request

```
GET http://198.51.100.10:8080/api/v2/platform/configuration/agent-logins/261
```

Response

```
{
   "statusCode":0,
   "agent-login":{
      "useOverride":"2",
      "tenantDBID":"1",
      "DBID":"261",
      "switchSpecificType":"1",
      "userProperties":{
         "TServer":{
            "wrap-up-time":"0"
         }
      },
      "state":"1",
      "switchDBID":"101",
      "loginCode":"111"
   }
}
```

You can specify the filters `object_path` and `read_folder_dbid`, which will cause the path of the object and the dbid of the object's parent folder to be returned in the response. Note that these filters can be used with any Configuration Server object. The following example shows the filters being used with agent groups.

Request

```
GET /api/v2/platform/configuration/agent-groups?object_path=1
```

Response

```
{
  "statusCode": "0",
  "agent-groups": [
    {
      "objectPath": "\\Configuration\\Environment\\Agent Groups\\test",
      "CfgGroup": {
        "state": "1",
        "name": "test",
        "tenantDBID": "1",
        "DBID": "1486",
        "capacityTableDBID": "0",
        "siteDBID": "0",
        "capacityRuleDBID": "0",
        "contractDBID": "0",
        "quotaTableDBID": "0"
      }
    },
```

```
    {
      "objectPath": "\\Configuration\\Environment\\Agent Groups",
      "CfgGroup": {
        "state": "1",
        "name": "test1",
        "tenantDBID": "1",
        "DBID": "1487",
        "capacityTableDBID": "0",
        "siteDBID": "0",
        "capacityRuleDBID": "0",
        "contractDBID": "0",
        "quotaTableDBID": "0"
      }
    }
  ]
}
```


Request

```
GET /api/v2/platform/configuration/agent-groups?object_path=1&read_folder_dbid=1
```

Response


```
{
  "statusCode": "0",
  "agent-groups": [
    {
      "objectPath": "\\Configuration\\Environment\\Agent Groups\\test",
      "folderDbid": "4385",
      "CfgGroup": {
        "state": "1",
        "name": "test",
        "tenantDBID": "1",
        "DBID": "1486",
        "capacityTableDBID": "0",
        "siteDBID": "0",
        "capacityRuleDBID": "0",
        "contractDBID": "0",
        "quotaTableDBID": "0"
      }
    },
    {
      "objectPath": "\\Configuration\\Environment\\Agent Groups",
      "folderDbid": "111",
      "CfgGroup": {
        "state": "1",
        "name": "test1",
        "tenantDBID": "1",
        "DBID": "1487",
        "capacityTableDBID": "0",
        "siteDBID": "0",
        "capacityRuleDBID": "0",
        "contractDBID": "0",
        "quotaTableDBID": "0"
      }
    }
  ]
}
```

POST Operations

To add a new object, make sure the `Content-Type` header is set to `application/json`. Your POST must include the full JSON representation of the object as it's required by Configuration Server. Optionally, you can also include **folderDbid** to specify the DBID of a folder where you want the object to be created. For example:

Request

```
POST  http://198.51.100.10:8080/api/v2/platform/configuration/agent-logins
{
   "agent-login":{
      "useOverride":"2",
      "tenantDBID":"1",
      "switchSpecificType":"1",
      "userProperties":{
         "TServer":{
            "wrap-up-time":"0"
         }
      },
      "state":"1",
      "switchDBID":"101",
      "loginCode":"111"
   }
  "folderDbid":123//optional integer you can use to specify the folder DBID where you want
the object to be created
}
```

You can find details about the configuration objects in Introduction to the Configuration Layer Objects.

PUT Operations

To update a configuration object, you need to include the corresponding Configuration Server delta structure. For example, to update a CfgSkill you need to include the CfgDeltaSkill.

> ### Important
> The URI naming format for delta objects is the same as the other configuration objects: the name in lower case, with each word separated by a hyphen (-). For example, CfgDeltaSkill would be "delta-skill".

Because the DBID is part of the delta structure in Configuration Server, all PUT requests should be sent via the **/platform/configuration/<type>** path.

Here's an example of how to update a configuration skill with a DBID of 218 by sending the delta structure:

Request

```
PUT http://198.51.100.10:8080/api/v2/platform/configuration/skills
```

```
{
   "delta-skill":{
      "CfgSkill":{
         "DBID":218,
         "name":"NewName"
      }
   }
}
```

To update an agent group (CfgDeltaAgentGroup), it would be "delta-agent-group":

Request

```
PUT http://198.51.100.10:8080/api/v2/platform/configuration/agent-groups
{
   "delta-agent-group": {
      "CfgDeltaGroup": {
         "CfgGroup": {
            "DBID": "232",
            "userProperties": {
               "NewSection": {"option1": "1234", "option2": "5678"}
            }
         }
      }
   }
}
```

DELETE Operations

DELETE operations work on a URI with a DBID and don't require any additional parameters. For example:

Request

```
DELETE http://198.51.100.10:8080/api/v2/platform/configuration/agent-logins/261
```