Server Services - 2.4

On this page:

Server

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server	Information about REST API internal state

JSON Representation

Properties	Description	type	Occurs
href	Auto reference	URI	1
name	"Server"	String	1
startDate	Starting date of the server	Date	1
memory	Statistics about memory usage (mega-bytes)	Structure	1
memory.totalInitialMemory	Total memory according to the JVM in mega-bytes, before initializing memory cache	Integer	1
memory.totalMemory	Total memory according to the JVM in mega-bytes	Integer	1

memory.freeMemory	Free memory according to the JVM in mega-bytes	Integer	1
memory.usedMemory	Total memory - Free memory	Integer	1
requests	Statistics about requests	Structure	1
requests.totalCounter	Total number of requests	Integer	1
requests.errorsCounter	Total number of requests in errors	Integer	1
requests.totalTime	Total elapsed time of requests milliseconds	Integer	1
requests.averageTime	Total elapsed time of requests in milliseconds	Integer	1
requests.maxTime	Longest elapsed time of all requests in milliseconds	Integer	1
clients	Statistics about concurrent clients	Structure	1
currentConcurrentClients	The current number of concurrent clients in read time. Note that the current request to fetch this information is not included in the count.	Integer	1
maxConcurrentClients	The maximum number of concurrent clients	Integer	1
status	Server status either "LOADING" or "READY"	String	1
loadDate	Date of Memory cache update. This date is set at start time of the server or when a reload is requested.	Date	1
abortedDomains	Array of aborted domains. An aborted domain is a domain for which loading has failed.	Array	1
abortedDomains[]	An aborted domain	Structure	0..1
abortedDomains[].name	Domain name	String	1
abortedDomains[].loadingDate	Date of loading start when the loading has been failed	String	1

license.status	License status regarding access to Central Bases	String	1

Use Case	Status	Description
No License	NO_LICENSE_KEY	License key was not found
	INVALID_LICENSE_KEY	License key is not valid
	CANNOT_ACCESS_LICENSE_KEY	License key file is not readable
	INVALID_LICENSE_FILE	Cannot find license key file
Restricted License	RESTRICTED_LICENSE	License is a restricted license
	GLOBAL_ACCESS_TOKENS_EXCEEDED	License is a restricted license, and quota of global access tokens is exceeded
	UNIT_ACCESS_TOKENS_EXCEEDED	License is a restricted license, and quota of unit access tokens is exceeded
Unrestricted License	UNRESTRICTED_LICENSE	License is an unrestricted license

domainsLocations	Get data source name and schema name for each domain. The ADMINISTRATOR role is required.	Structure	0..1
recommendedDbVersion	The preferred version of AIP (for compliancy with database schema)		
securityMode	This is the configuration value of property **security.mode** from the **application.properties** file: - "default": authentication based of configuration files - "ldap": authentication based on LDAP protocol - "saml": authentication based on SAML protocol	String	1
samlSingleLogout	This is the configuration value of **security.saml.single.logout** property from **application.properties** file. - "true": logout action is enabled for "saml" security mode - "false": logout action is disabled for "saml" security mode	Boolean	1
languages	Installed translations	Array	1
languages[ ]	An available locale language	String	0..1
reportEnabled	Check whether the configuration variable report. is set report.reportGenerator	Boolean	1

JSON Example

GET DEMO

```
{
        "href": "server",
        "name": "Server",
        "startDate": {
                "time": 1612255923923,
                "isoDate": "2021-02-02"
        },
        "memory": {
                "totalInitialMemory": 109,
                "totalMemory": 125,
                "freeMemory": 44,
                "usedMemory": 81
        },
        "requests": {
                "totalCounter": 20,
                "errorsCounter": 0,
                "totalTime": 18447419617,
                "averageTime": 922370980,
                "maxTime": 929175903
        },
        "clients": {
                "currentConcurrentClients": 0,
                "maxConcurrentClients": 1
        },
        "loadDate": {
                "time": 1612256578092,
                "isoDate": "2021-02-02"
        },
        "abortedDomains": [],
        "status": "READY",
        "version": "X.X.X-XXX",
        "recommendedDbVersion": "8.3.3",
        "license": {
                "status": "NO_LICENSE_KEY"
        },
        "securityMode": "default",
        "samlSingleLogout": false,
        "sessionTimeout": 900,
        "languages": [
                ".gitkeep"
        ],
        "reportEnabled": false,
        "jiraEnabled": true,
        "domainsBinding": {
                "href": "server/domains-bindings",
                "name": "Domains/schemas bindings"
        },
        "authorizations": {
                "href": "server/authorizations",
                "name": "Authorized applications definitions per users"
        }
}
}
```

Server Cache Management

> ⊙ **Warning**
>
> Administrator role is required.

REST server stores portfolio objects, configuration, and snapshots in a memory cache. This memory cache is loaded as soon as the REST Server is started.

An URL allows to reload all domains in memory cache. This action may be required when a new snapshot has been added, and can be performed from a command line with a tool such as "curl":

Use Curl for automation:

```
C:> curl -u admin:cast -H "Accept: application/json" http://localhost:8080/rest/server/reload
```

For each domain, Lucene index files can be created to allow components or violations search.

Lucene index files are created at start time and reload time if these options are enabled:

---

application.properties

```
# Rebuild Lucene components index on start if outdated (true or false)
rebuildComponentsSearchIndexesOnStart=true
# Rebuild Lucene violations index on start if outdated (true or false)
rebuildViolationsSearchIndexesOnStart=false
```

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/reload{?parameters}	Sync clients with new domains. Reload the domains configuration and refresh the server memory cache, Example of use: A new application onboarding requires to create a new ED domain in the domains.properties file. The client may receive an HTTP Status 503 "Service not available" if a server/refresh call is already in progress. In case of intensive traffic, the service awaits for the end of current processing request to start. This web service has the following effects - Reload the configuration of domains (domains.properties file) - Reload the server memory cache - Rebuild Lucene Indexes - Respond a "503 Service unavailable" to any client during the reload - Invalidate the browser cache
GET	application/json	server/refresh{?parameters}	Sync clients with new snapshots. Refresh the server memory cache when a new snapshot has been added. Example of Usage: A new snapshot has been taken for an ED domain. Example of Usage: AAD domains needs to be refreshed when a new application or a new snapshot has been added. This processing is made in the background with no interruption for HD clients, and reduced interruption for ED clients. This web service has the following effects - Reload the server memory cache - Rebuild Lucene Indexes - Respond a "503 Service unavailable to ED clients requesting Lucene index data, whereas this Lucene index is under construction for this specific domain. - Invalidate the browser cache **Note:** If two concurrent requests are sent for 2 different domains, then these domains will be refreshed in parallel. **Note**: onboarding of a new applications must be performed with the "server/reload" or the "server/domains-bindings" web service that will update the domains.properties configuration file and will trigger a "server/reload" service for these domains.
GET	application/json	server/reset	Sync clients after a snapshot reconsolidation, a license key change, or a change of the license.xml file (authorizations in case of restricted license). The web service reloads the license.key file, the license.xml file and invalidates the browser cache.

Parameters

URI Parameter	Description	Values	Default value
domain	Specify a single domain to reload (for example to refresh a domain after a snapshot)	a string	$all

DBMS Warm-up Service

> ⊘ **Warning**
>
> Administrator role is required.

This service fetches data (results, components, violations) for domains hosted in central bases, in order to pre load data in memory afer a DBMS cold restart.

It avoids to penalize first user fetching data.

This service loops on each domain hosted by a central base, and trigger some queries on components, violations and assessment results.

> **Use Curl for automation:**
>
> ```
> C:> curl -u admin:cast  http://localhost:8080/rest/server/warmup
> ```

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/warmup	Warm up DBMS after a cold restart (central base hosts only)

Lucene Index

For each domain, a Lucene index is created to allow the search for components.

Another Lucene index can be created to allow the search for violations.

URI Templates

HTTP Action	Media Type	URI Templates	Description
PUT	application/json	{Domain}/components-index	Create or overwrite the components search index for a domain ⊘ **Warning** Administrator role is required.
GET	application/json	{Domain}/components-index	Get index status for a domain
PUT	application/json	{Domain}/violations-index	Create or overwrite the violations search index for a domain ⊘ **Warning** Administrator role is required.
GET	application/json	{Domain}/violations-index	Get index status for a domain

JSON Representation

Properties	Description	type	Occurs
href	Auto reference	URI	1
name	Name	String	1
status	- upToDate : Index is up to date - N/A : not applicable, no index - toUpdate : last snapshot is more recent than the index date, an index rebuild is required	String	1
date	Index file date	Date	0..1
lastSnapshotDate	Last Snapshot date	Date	0..1
size	Index file size	Integer	0..1

JSON Example

```
{
        "href": "ENDTOEND83/components-index",
        "name": "Components search index for applications of ENDTOEND83",
        "status": "upToDate",
        "date": {
                "time": 1496752452859,
                "isoDate": "2017-06-06"
        },
        "lastSnapshotDate": {
                "time": 1493778823000,
                "isoDate": "2017-05-03"
        },
        "size": 4145870
}
```

Domains Bindings

A Domain binding associates a domain name with a data source name and a schema name.

This resource is based on the use of *domains.properties* file.

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server /domains-bindings	Get all domains bindings ⚠ **Warning** Administrator role is required.

PUT	application/json	server /domains-bindings	Update the domains bindings, or create new domains bindings.

A triplet (domain, data source, central) is replaced for an existing domain or an existing central, otherwise it is added.

This web service triggers the "reload" service for the domains listed in the payload, in order to update the server memory cache and Lucene index files.

This Web service overwrites the *domains.properties* file.

> ⚠ **Warning**
>
> This service accepts the "Authorization" HTTP header to transmit user's credentials, so that a prior call to the login "service" is not required. Thus, we can start the Web Server with an empty list of domains, and bypass the "login" service that prevents connection when no domain is defined.

In case of exception when writing this file, an HTTP Status "403 Forbidden" is returned. Check the permissions of this file.

> ⚠ **Warning**
>
> Administrator role is required.

Example:

Assuming there are two existing domains AED1, AED2, add a new domain:

```
[
        {
                "name": "AED3",
                "dataSource": "DEV_CSS2",
                "schema": "appli1_central"
        }
]
```

Assuming there are three existing domains AED1, AED2, AED3, change schemas for domains AED1, AED2:

```
[
        {
                "name": "AED1",
                "dataSource": "DEV_CSS2",
                "schema": "appliA_central"
        },
        {
                "name": "AED2",
                "dataSource": "DEV_CSS2",
                "schema": "appliB_central"
        }
]
```

DELETE	applicatio n/json	server /domains- bindings	Update the database binding for a list of domains, or create some new domains with their own bindings.

This web service does not need to trigger the "reload".

This Web service overwrites the *domains.properties* file.

> ⊘ **Warning**
>
> This service accepts the "Authorization" HTTP header to transmit user's credentials, so that a prior call to the login "service" is not required. Thus, we can start the Web Server with an empty list of domains, and bypass the "login" service that prevents connection when no domain is defined.

In case of exception when writing this file, an HTTP Status "403 Forbidden" is returned. Check the permissions of this file.

> ⊘ **Warning**
>
> Administrator role is required.

Example:

Assuming there are two existing domains AED1, AED2, remove domain AED2

```
[
        {
                "name": "AED2"
        }
]
``` |

JSON Representation

Properties	Description	type	Occurs
name	Domain name	String	1
dataSource	Data source name	String	1
schema	Schema Name	String	1

JSON Example

```
[
        {
                "name": "AED1",
                "dataSource": "DEV_CSS2",
                "schema": "appli1_central"
        },
        {
                "name": "AED2",
                "dataSource": "DEV_CSS2",
                "schema": "appli2_central"
        }
]
```

Profiles

This resource is based on the use of profiles table.

URI Templates

HTTP Action	Media Type	URI Templates	Description

⊘

GET	application/json	server/profiles	Get all users' profiles
			⚠ **Warning** Administrator role is required.
PUT	application/json	server/profiles	Update users' profiles. Create or replace profiles ⚠ **Warning** Administrator role is required.
DELETE	application/json	server/profiles	Delete users' profiles. ⚠ **Warning** Administrator role is required.

JSON Representation

Properties	Description	type	Occurs
profileName.name	profile name	String	1
authorizations	All authorizations for a user	Array	0..1
authorizations[ ]	An authorization defines which applications can be accessed	Structure	1
authorizations[ ].allApplications	Set applications access to all applications	Boolean	0..1
authorizations[ ].application authorizations[ ].adgDatabase	Allow access to an application matching application name and adgDatabase name;	String	0..1
authorizations[ ].applicationPattern authorizations[ ].adgDatabasePattern	Allow access to all applications matching applicationPattern regular expression and adgDatabasePatern regular expression	String	0...1
authorizations[ ].tag authorizations[ ].category authorizations[ ].restrictions	Allow access to all applications matching a tag of a category	String	0..1
authorizations[ ].technology authorizations[ ].restrictions	Allow access to all applications matching a technology	String	0..1
authorizations[ ].restrictions	All restrictions, applicable for authorizations defined with tag, category and technology attributes	Array	0..1
authorizations[ ].restrictions[ ]	A restriction	Structure	0..1
authorizations[ ].restrictions[ ].application authorizations[ ].restrictions[ ].adgDatabase	Deny access to an application matching application name and adgDatabase name	String	0..1
authorizations[ ].restrictions[ ].applicationPattern authorizations[ ].restrictions[ ].adgDatabasePattern	Deny access to all applications matching applicationPattern regular expression and adgDatabasePatern regular expression	String	0...1
authorizations[ ].restrictions[ ].tag authorizations[ ].restrictions[ ].category	Deny access to all applications matching a tag of a category	String	0..1
authorizations[ ].restrictions[ ].technology	Deny access to all applications matching a technology	String	0..1
roles	All roles for a user	Array	0..1
roles[ ]	Defines all roles available for user	Structure	1
roles[ ].key	Specify which role key	String	1

JSON Example

```
[
    {
            "profileName": {
                    "name": "GroupWith2Apps_group_profile"
            },
            "authorizations": [
                    {
                            "application": "Billing platforms",
                            "adgDatabase": "demo_709_central",
                            "restrictions": []
                    },
                    {
                            "application": "Dream Team",
                            "adgDatabase": "adg_contrex_central",
                            "restrictions": []
                    },
                    {
                            "application": "TransactionNet",
                            "adgDatabase": "ice_800_central",
                            "restrictions": []
                    }
            ],
            "roles": [QUALITY_MANAGER]
    }
]
```

User Profiles

This resource is based on the use of user_profiles table.

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/user-profiles	Get all users' along with profiles assigned to each user ⊙ **Warning** Administrator role is required.
PUT	application/json	server/user-profiles	Update users' profiles. Create or replace profiles to users or groups ⊙ **Warning** Administrator role is required. WARNING: there is not checking of user/group names validity.
DELETE	application/json	server/user-profiles	Delete users' profiles. ⊙ **Warning** Administrator role is required.

JSON Representation

Properties	Description	type	Occurs
user	user name. Excludes the group property.	String	1
group	LDAP group name. Excludes the user property.	String	1

profiles	All profiles for a user	Array	0..1
profiles[]	Defines all profiles available for user	Structure	1
profiles[].name	Specify which profiles name	String	1

JSON Example

```
[
        {
                "profiles": [
                        {
                                "name": "admin_profile"
                        }
                ],
                "user": "admin"
        },
        {
                "profiles": [
                        {
                                "name": "ATLAS_user_profile",
                                "name": "CIO_user_profile"
                        }
                ],
                "user": "QualityManNoRightsOnCastOldCode"
        }
]
```

License Key

This resource returns license key and its status. This resource is disabled in **INTEGRATED** security mode.

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/license-key	Get license key along with its status in dashboard. ⊘ **Warning** Administrator role is required.

JSON Representation

Properties	Description	type	Occurs
key	license key	String	1
status	Server status	String	1

JSON Example

```
{
        "key": "CAST_R&D:1;Unrestricted:Unlimited:XXXXXXXX",
        "status": "UNRESTRICTED_LICENSE"
}
```

Roles definition

This resource returns definition for each roles available in dashboard.

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/roles-definition	Get user roles definition available in dashboard.

JSON Representation

Properties	Description	type	Occurs
key	role key	String	1
name	role name	String	1
description	role description	String	1

JSON Example

```
[
        {
                "key": "ADMIN",
                "name": "Admin ",
                "description": "The Admin has rights to access all applications and he has role provides
permission to execute reload, reset and refresh dashboard memory, create, update and delete categories and
tags."
        },
        {
                "key": "QUALITY_MANAGER",
                "name": "Quality Manager ",
                "description": "The Quality manager role provides permission to add and remove objects from the
Action Plan and to use the Engineering Dashboard - Action Plan Recommendation feature."
        },
        {
                "key": "QUALITY_AUTOMATION_MANAGER",
                "name": "Quality Automation Manager ",
                "description": "The Quality automation manager role provides permission to add and remove
objects from the Education list."
        },
        {
                "key": "EXCLUSION_MANAGER",
                "name": "Exclusion Manager ",
                "description": "The Exclusion manager role provides permission to add and remove objects from
the Exclusion list."
        },
        {
                "key": "CODE_RESTRICTED",
                "name": "Code Restricted ",
                "description": "The Code restricted role prevents users from viewing source code in the
Engineering Dashboard."
        }
]
```

All applications

This resource returns all applications available across all domains from rest API cache.

URI Templates

HTTP Action	Media Type	URI Templates	Description

🛑

GET	application/json	server/all-applications	Get all applications available across all domains .
			⊘ **Warning** Administrator role is required.

For JSON representation and JSON example check here Application Structure Resources - 2.4#Application

All technologies

This resource returns all technologies available across all domains from rest API cache.

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/all-technologies	Get all technologies available across all domains . ⊘ **Warning** Administrator role is required.

JSON Representation

This web service returns simple list from REST API cache.

JSON Example

```
[HTML5, JEE, SQL, JAVA]
```

Users-Groups

This resource returns list of users and groups based on security mode. This resource is not available in **INTEGRATED** security mode.

URI Templates

HTTP Action	Media Type	URI Templates	Description
GET	application/json	server/users-groups?{parameters}	Get users and groups list for the default security mode, and LDAP/SAML when LDAP is configured ⊘ **Warning** Administrator role is required.

Query Parameters

Parameter	Description	Values	Default value
prefix	Specify the starting letters of the user name or group name. For LDAP 3 letters are required.	a text	N/A

JSON Representation

Properties	Description	type	Occurs
name	user or group name	String	1

type	user/group	String	1

JSON Example

```
[
        {
                "name": "QualityAutoManNoRightsOnBigBen",
                "type": "user"
        },
        {
                "name": "StaticFileGroupWithNoApps",
                "type": "group"
        },
        {
                "name": "ExclusionManNoRightsOnDreamTeam",
                "type": "user"
        }
]
```