

AIP Console - Interactive API documentation

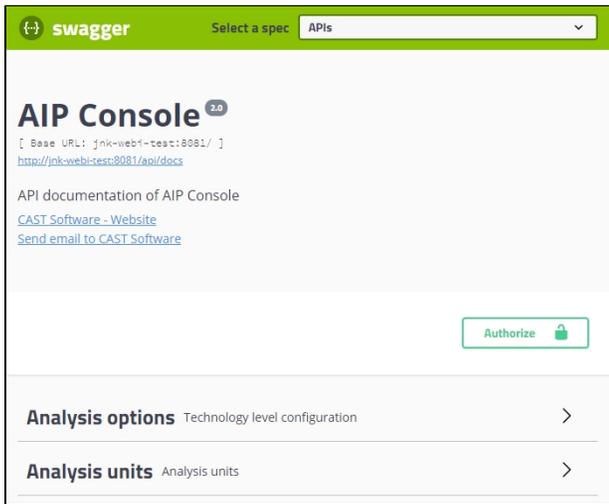
- [Introduction](#)
- [Authorization](#)
- [Using the API interactively](#)
- [Using curl to interact with the API - request headers](#)
- [How to schedule a "job" with the API](#)
 - [List of available "jobs"](#)
 - [Getting the request payload](#)
 - [Use case](#)
 - [Job status](#)
- [Real life implementation of the API](#)

Introduction

AIP Console has API documentation provided by Swagger. This documentation is interactive and you can test requests directly. The API documentation is accessible via the following URL (change the URL to suit your own environment):

```
http://<aip_console>:8081/swagger-ui.html
```

Click to enlarge



 The AIP Console API interface contains API commands for both the AIP Console and AIP Node(s).

Authorization

Before you can interactively use the graphical API, you must be authorized. Authorization requires a **user API key**, which you can generate in your [user profile settings](#):

AIP Console ^{2.0}

[Base URL: jnk-web1-test:8081/]
<http://jnk-web1-test:8081/api/docs>

API documentation of AIP Console
[CAST Software - Website](#)
[Send email to CAST Software](#)

Authorize 

Analysis options Technology level configuration >

Using the API interactively

Show the request type you want to test:

Analysis options Technology level configuration

GET	/api/applications/{guid}/technologies/bo	Get Business objects analysis options	
GET	/api/applications/{guid}/technologies/cpp	Get cpp analysis options	
PUT	/api/applications/{guid}/technologies/cpp/configuration	Update cpp analysis configuration	

Expand the specific request using the blue section in the top left:

Analysis options Technology level configuration

GET	/api/applications/{guid}/technologies/bo	Get Business objects analysis options	
GET	/api/applications/{guid}/technologies/cpp	Get cpp analysis options	
PUT	/api/applications/{guid}/technologies/cpp/configuration	Update cpp analysis configuration	

and click **Try it out**:

Click to enlarge

Analysis options Technology level configuration ▼

GET /api/applications/{guid}/technologies/bo Get Business objects analysis options 🔒

Parameters Try it out

Name	Description
guid * required string (path)	guid

Using curl to interact with the API - request headers

It is possible to interact with the API using [curl](#), in which case, as well as an [API key](#), you will also need to use a [CSRF token](#), for example:

```
curl --location --request GET 'http://localhost:8081/api/applications' \
--header 'x-api-key: qZUfQfX3.G5Yx7ehS2HI9Va5k4h0uTM7jbF0YWT9R' \
--header 'x-xsrf-token: e01b177c-f5d1-4137-a9c3-8ef39bef67e6'
```

To generate a CSRF token, run a GET request to **/api/**:

```
curl --location --request GET "http://host:port/api/"
```

And then retrieve the token from cookies:

```
XSRF-TOKEN          e01b177c-f5d1-
                    4137-a9c3-
                    8ef39bef67e6
```

How to schedule a "job" with the API

The **/job** API is the main API in AIP Console and is used to launch actions such as create application, add version, launch analysis etc. The basic flow of consuming the **/jobs** API is:

- Schedule a job with a job type
- Poll the job details API to get the real-time job status

List of available "jobs"

Job type	Description
analyze	Run analysis for a specific version of an application
reject_version	Reject a version of an application
delete_version	Delete a version of an application
purge_version	Purge a version of an application
delete_snapshot	Delete a snapshot of an application
upload_application	Upload application to measurement service
consolidate_snapshot	Consolidate a snapshot
declare_application	Create an application

delete_application	Delete an application
function_points	Compute function points
deliver_version	Create and deliver a version for an application
BACKUP	backup the triplet and the delivery folder for an application
backup_measurement	backup measurement schema
RESTORE	restore an application to a previous state based on a backup
SHERLOCK_BACKUP	Create a Sherlock backup
upload_deliver_version	Unzip previously uploaded source and deliver version
upload_snapshot_version	Unzip previously uploaded source, create version, analyze and create a snapshot
rescan_application	Clone previous version, launch analysis and create a snapshot
add_version	Add a new version for an application
clone_version	Clone an existing version's structure
edit_version	Edit an existing version
sync_application	Synchronize application
dataflow_security_analyze	Run dataflow security analyzer

Getting the request payload

Depending on the job type, the request payload for each job is different, therefore the easiest way to get the exact payload is to run a test using your browser's dev tools:

- F12 to open the dev tools of Chrome (for example) and launch the action in AIP Console
- In dev tools, switch to the **Network tab**
- Click on the POST request to **/jobs** and then click the **Headers tab**
- Check the request payload

The screenshot shows the Chrome DevTools Network tab with the Headers sub-tab selected. The request is a POST to /jobs. The headers include: Referer: https://demo-eu.castsoftware.com/ui/index.html, Sec-Fetch-Dest: empty, Sec-Fetch-Mode: cors, Sec-Fetch-Site: same-origin, User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome..., X-Requested-With: XMLHttpRequest, and X-XSRF-TOKEN: d57b70e6-da07-4735-9f70-bec028e3b8a3. The Request Payload is expanded to show a JSON object: {jobType: "add_version", ...}, with jobParameters containing details like appGuid, appName, endStep, and versionName.

```

Name
  jobs
  9aab2b40-5e93-4197-ae74-e895...
  jobs
  jobs
  9aab2b40-5e93-4197-ae74-e895...

Headers
  Referer: https://demo-eu.castsoftware.com/ui/index.html
  Sec-Fetch-Dest: empty
  Sec-Fetch-Mode: cors
  Sec-Fetch-Site: same-origin
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome...
  X-Requested-With: XMLHttpRequest
  X-XSRF-TOKEN: d57b70e6-da07-4735-9f70-bec028e3b8a3

Request Payload
  view source
  {jobType: "add_version", ...}
  {
    jobParameters: {
      endStep: "consolidate_snapshot",
      versionName: "Version-2020-04-27T14-29-55",
      ...
    }
  }
  
```

Use case

Example to **add a version** of a given application and generate a snapshot. POST `/jobs` to schedule a job with type `add_version`, using an existing predefined source folder: `source.folder.location: webi/source_code`

```
curl --location --request POST 'http://host:port/api/jobs' \
--header 'x-api-key: 87IUS4TO.9SzUy5om67lkhlcRQJzppf3ltrFhDcUo' \
--header 'x-xsrf-token: 4d1620ac-dbfd-4ffa-ac56-e14fa9bd88da' \
--header 'Content-Type: application/json' \
--data-raw '{
  "jobType": "add_version",
  "jobParameters": {
    "endStep": "consolidate_snapshot",
    "versionName": "Version-2020-04-27T14-29-55",
    "appGuid": "225e7c73-2dbd-486d-a1df-53d66f39ff75",
    "appName": "spag2",
    "deliveryConfigGuid": "",
    "releaseDate": "2020-04-27T12:29:55.987Z",
    "sourcePath": "sources:webi/source_code",
    "fromVersionGuid": "3b5cdb3a-eb64-4d63-b38c-25454282b495",
    "extensionAutoConfigEnabled": true,
    "objectives": "global_risk,functional_points",
    "moduleGenerationType": "one_per techno"
  }
}'
```

In the response of the api, we get a **jobUrl** url which gives realtime status of the job:

```
{
  "appGuid": "225e7c73-2dbd-486d-a1df-53d66f39ff75",
  "jobGuid": "c784880e-377a-4dc3-a6f5-6a63d5866bf2",
  "jobUrl": "/api/jobs/c784880e-377a-4dc3-a6f5-6a63d5866bf2"
}
```

Poll the **jobUrl** to get the realtime status:

```
curl --location --request GET 'http://host:port/api/jobs/c784880e-377a-4dc3-a6f5-6a63d5866bf2' \
--header 'x-api-key: 87IUS4TO.9SzUy5om67lkhlcRQJzppf3ltrFhDcUo' \
--header 'x-xsrf-token: 4d1620ac-dbfd-4ffa-ac56-e14fa9bd88da' \
```

Job status shows in the response with the current step:

```
{
  "progressStep": "code_scanner"
  "state": "started"
  "updated": "2020-04-27T13:08:34.662Z"
  "url": "/api/jobs/0c84c2ad-765e-4865-9598-30112ac81dd3"
}
```

Job status

Status	Description
completed	Job is done successfully
started	Job is in progress
starting	Job is in pending state waiting for execution
stopped	Job was interrupted in the middle of the process
failed	Job completed with failure
cancelled	Job is cancelled by user

Real life implementation of the API

CAST has used the AIP Console API to build various tools that can be used to improve the analysis process (**Jenkins plugin** and **automation**). You can find out more about these tools here: <https://github.com/CAST-Extend/com.castsoftware.uc.aip.console.tools> - explore tools to see how the API can be used in a real life context.



Note:

- If you are a member of the CAST-Extend GitHub organization, you can push merge requests from another branch (master branch is read-only) or you can fork the project and change it to suit your own requirements
- These tools are not supported by CAST