# Adobe Task Queue Manager getting started guide

Introduction

This guide describes how to setup a system to allow access to the cloud API of the Adobe Task Queue Manager and perform a set of basic requests.

## Features

Task Queue Manager allows customers to send jobs from one machine to another while receiving real time updates about progress. These jobs can either be encode jobs (similar to Adobe Media Encoder Remote) or script jobs with arbitrary extension code. The transfer and delivery from the client to the server machine is handled by queues in the cloud. A server/worker can be deployed to any computer running PremierePro or Adobe Media Encoder. The content of e.g. encode jobs is composed of the project file (or team project snapshot) an optional export preset file and meta data.

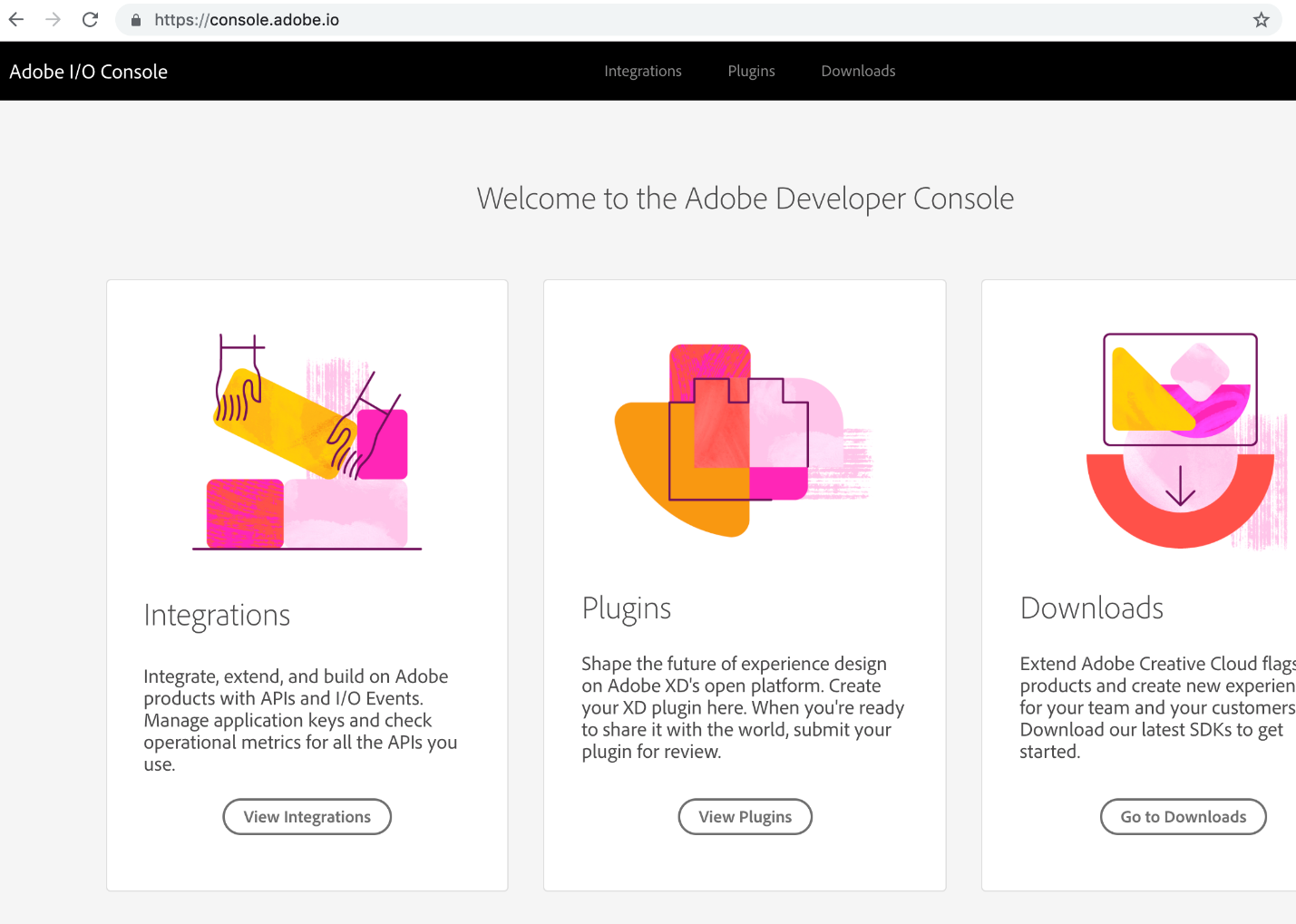
## 

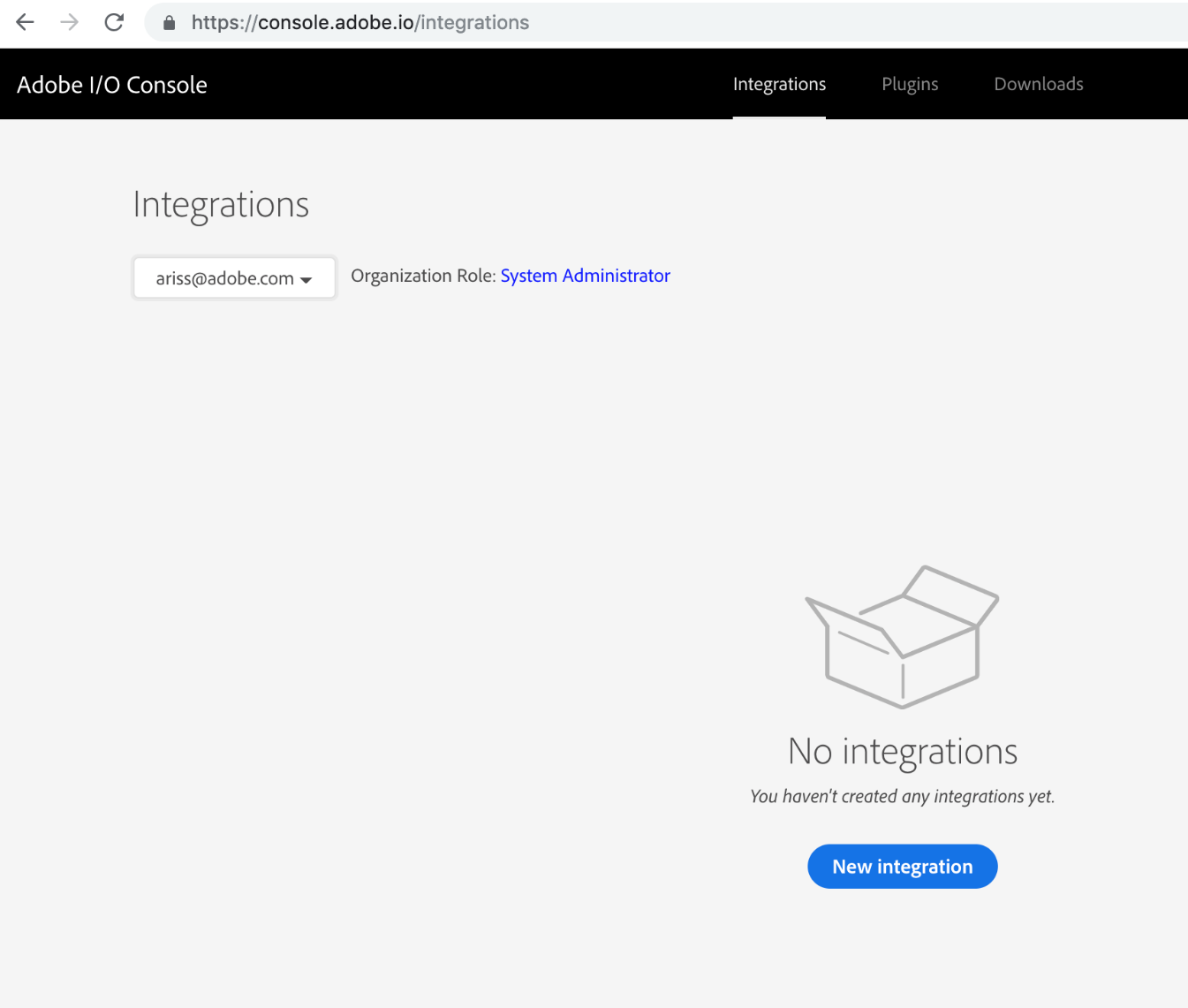
## Installing a worker

1. [Install Java jdk 11](https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/downloads-list.html)
2. Install Premiere Pro 14.0 and Adobe Media Encoder 14.0 (2020)
3. The worker can be installed by extracting the zip file to any folder on a machine. It can also be setup to run as a service or started as a process.
4. Retrieve the account credentials from console.adobe.io and configure the cc-worker.xml file
5. Make sure there is a queue already created to be set on the worker (created via API or from Premiere Pro DOM calls)

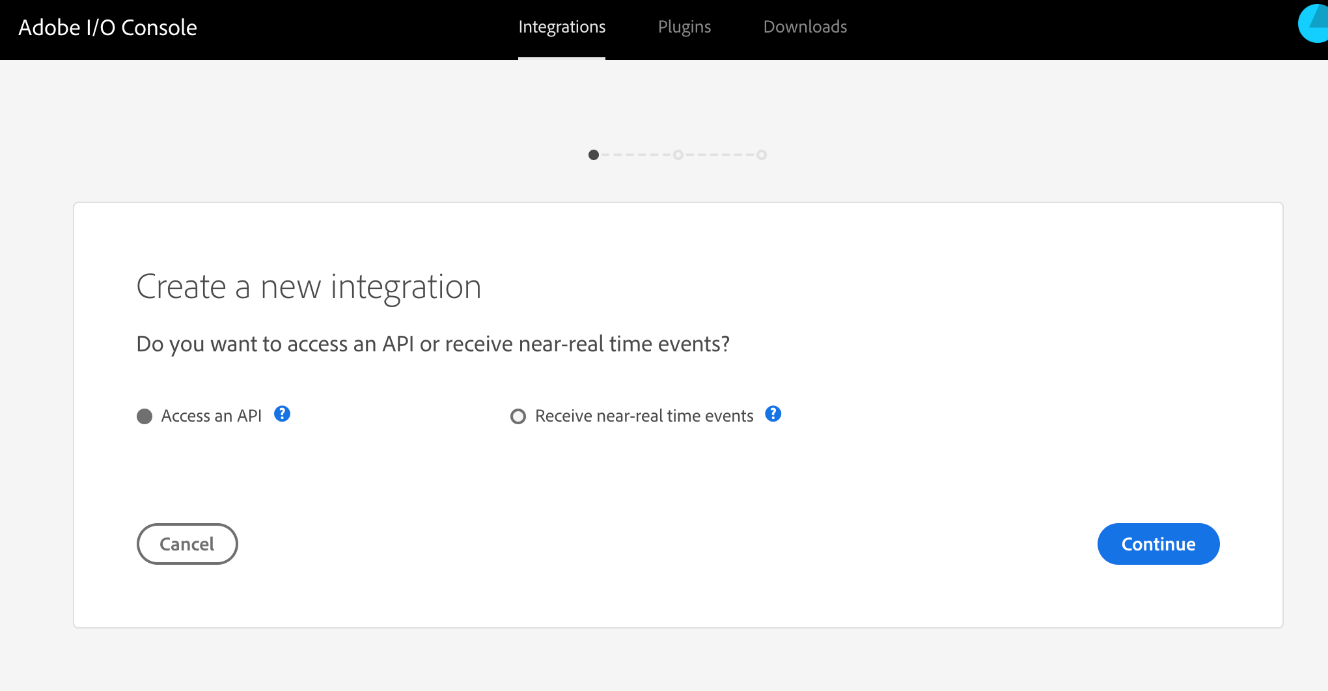
## Account setup

Login using an administrator account of your domain on <https://console.adobe.io/>. And choose to create a new integration.

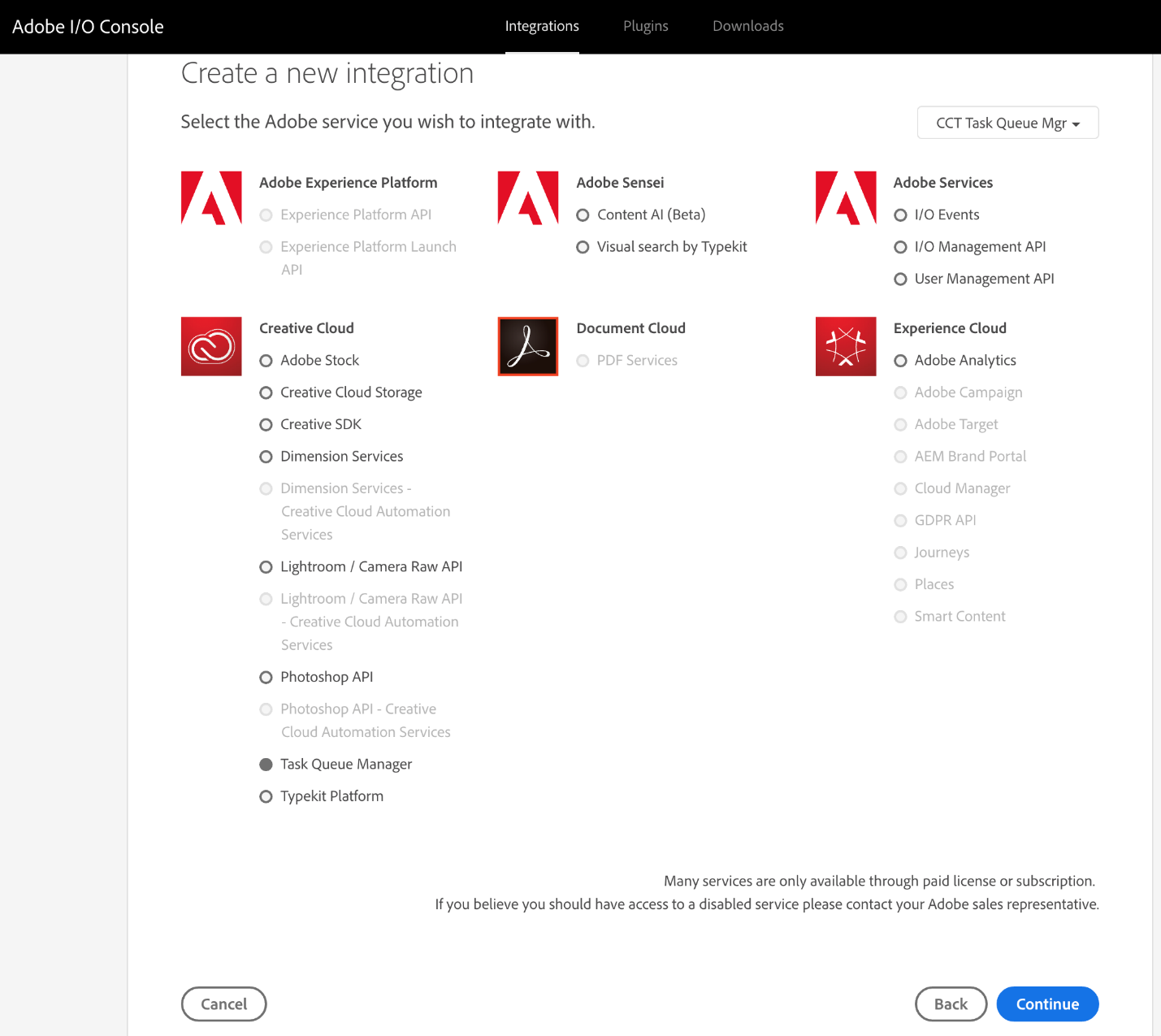




On the following screen choose "Access an API"

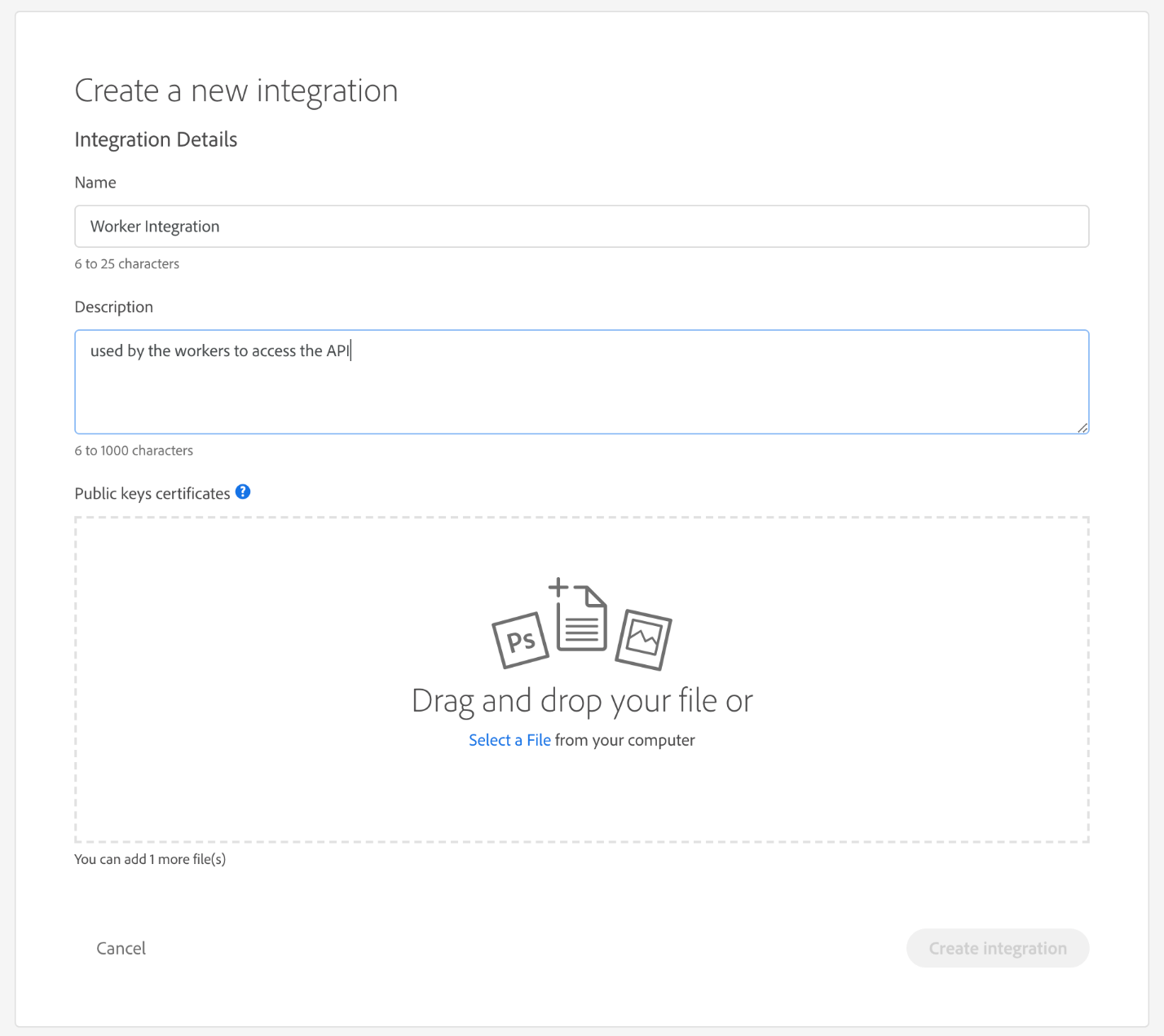


Choose "Task Queue Manager" from the list.

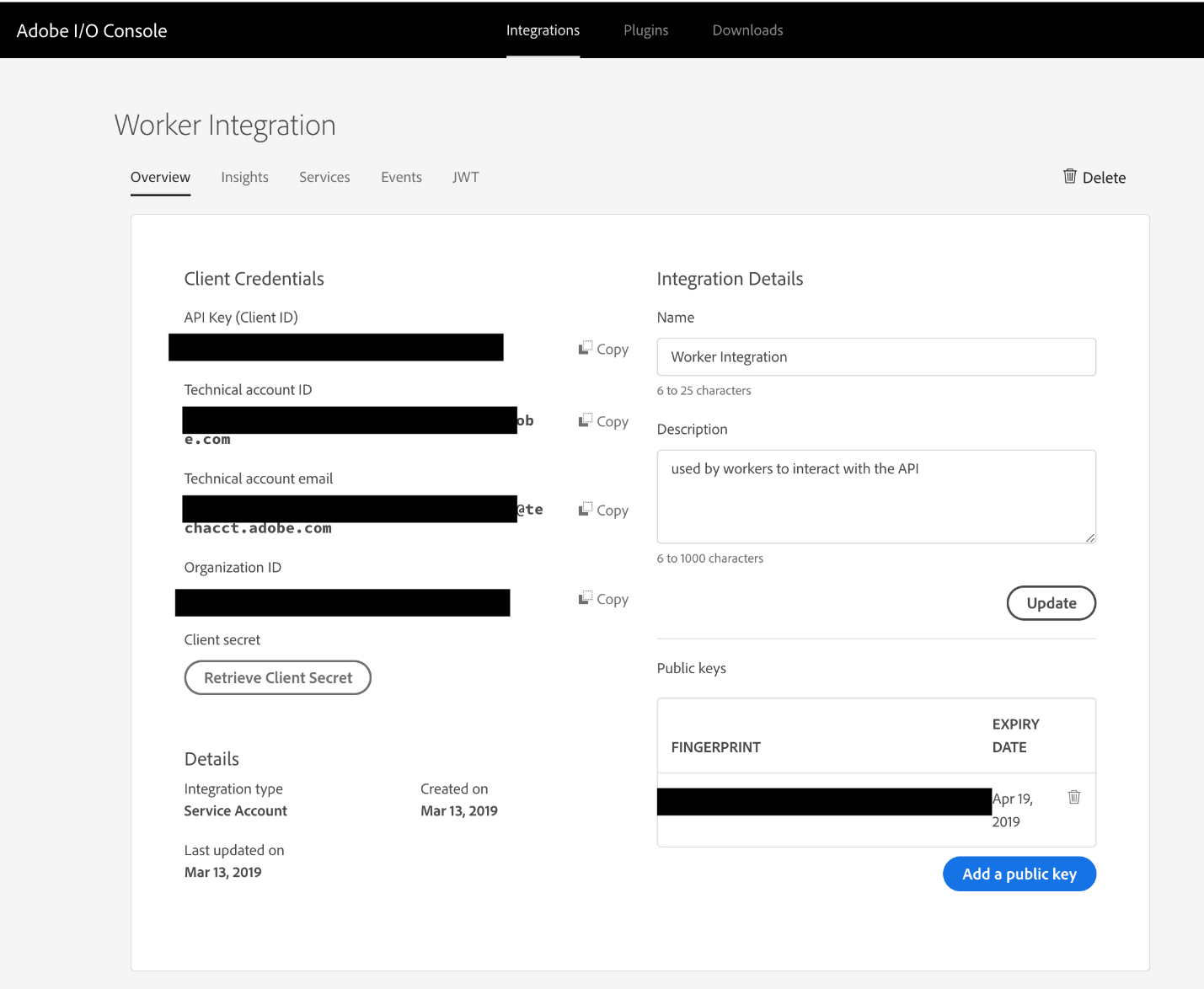


In the following screen you can name the integration and provide a security certificate used for authentication - see the "Creating the worker keystore file and certificate" chapter in the developer guide below or

<https://www.adobe.io/authentication/auth-methods.html#!AdobeDocs/adobeio-auth/master/AuthenticationOverview/ServiceAccountIntegration.md#step-2-create-a-public-key-certificate>. Drag the "certificate\_pub.crt" into the drop box and continue.



The integration is now created and you can use the following page to retrieve all configuration values needed to setup a worker. (API\_ADOBE\_USER\_ID in the config file refers to the Technical account ID on this page)



## Running as a process (for debugging only)

Update the run-as-process.bat file found in the worker directory with the information found on the [adobe.io](http://adobe.io) developer console. Then start the bat file. Make sure java RE >= 11 is installed and available on the PATH.

## Running as a service

The configuration file cc-worker.xml can be found in the service folder. The XML lists all required values which were generated in the previous step for creating a JWT token on [Adobe.io](http://Adobe.io). Make sure to use a local user which has sufficient rights to start both Premiere Pro and Adobe Media Encoder. The service should not be run on the default service account.

<service>

<id>dva-worker</id>

<name>Adobe Task Queue Manager</name>

<description>This service runs the Adobe Task Queue Manager</description>

<env name="WORKER\_HOME" value="%BASE%"/>

<env name="API\_ADOBE\_USER\_ID" value="user-id-from-adobe-io-console"/>

<env name="API\_KEY" value="api-key-from-adobe-io-console"/>

<env name="API\_ADOBE\_ORG\_ID" value="org-id-from-adobe-io-console"/>

<env name="API\_CLIENTSECRET" value="clientsecret-from-adobe-io-console"/>

<env name="QUEUE\_ID" value="the-queue-id-to-process" />

<env name="API\_KEYSTORE\_FILE" value="%BASE%\keystore.ks" />

<env name="API\_KEYSTORE\_PASSWORD" value="changeme" />

<!-- <env name="API\_ENDPOINT" value="cloud-dispatcher-beta.adobe.io" /> -->

<executable>java</executable>

<arguments>-Djava.io.tmpdir=%TEMP% -Xmx1024m -jar "%BASE%\worker.jar"</arguments>

<onfailure action="restart" delay="10 sec"/>

<log mode="roll-by-time">

<pattern>yyyyMMdd</pattern>

</log>

<logpath>%BASE%\logs</logpath>

<serviceaccount>

<domain>YOURDOMAIN</domain>

<user>useraccount</user>

<password>password or %PASSWORD\_FROM\_ENVIRONMENT\_VARIABLE%</password>

<allowservicelogon>true</allowservicelogon>

</serviceaccount>

</service>

The process can be setup to run as a service as soon as the configuration file and keystore file are correctly setup:

This requires Admin rights on the local machine

cc-worker.exe install

cc-worker.exe start

The process will populate log files to the /logs directory of the same folder. Make sure the process is running as a user which is allowed to write to that folder, and has rights to start Adobe Media Encoder. The correct user can be verified when opening the service list, opening the "Adobe Task Queue Manager" service and inspecting the "log on" tag. It should show the service using a specific user account instead of "Local system account".

Please note cc-worker.exe install has to be repeated every time the user password is changed in the configuration.

**Prerequisite:**

Task Queue Manager features are supported only on 'New World' scripting engine which is not the default on Premiere Pro/ AME 14.0.0 builds. Therefore in order to use Task Queue Manager, New World needs to be enabled on all clients and worker machine using one of the following methods.

1. Quit Premiere Pro/ Media Encoder. Copy attached debug files for AME and Premiere Pro and replace them at following location.

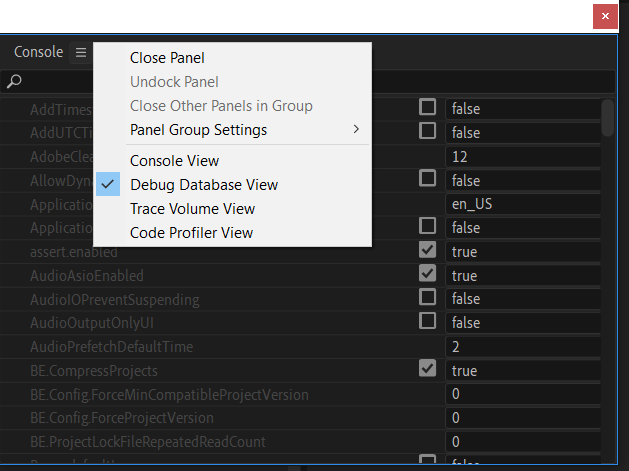
|  |  |  |
| --- | --- | --- |
|  | **Media Encoder** | **Premiere Pro** |
| Windows | C:\Users\<UserName>\Documents\Adobe\  Adobe Media Encoder\14.0\ | C:\Users\Administrator\AppData\Roaming\Adobe\Premiere Pro\14.0\ |
| Mac | /Users/UserName/Documents/Adobe/Adobe\ Media\ Encoder/14.0/ | /Users/<UserName>/Library/Preferences/Adobe/Premiere\ Pro/14.0/ |

The debug files changes following flag with respect to each product:

* Premiere Pro: **ScriptLayerPPro.EnableNewWorld** to **true**
* Media Encoder: **AME**.**ScriptLayer.EnableNewWorld** to **true**

Attached debug database file will only enable 'New World' scripting engine and rest of debug flags default to original values. If other debug flags are also being used, they need to be taken care of also.

1. Launch Premiere Pro. Open any project. Press CTRL+F12(Windows)/Command+Fn+F12(Mac) that will open Console Window. Click on hamburger menu next to Console and select Debug Database View. Search for debug flag **ScriptLayerPPro.EnableNewWorld**and change it to ***true*** by selecting the checkbox. New settings will take place on re launch. Launch Adobe Media Encoder, Press CTRL+F12 that will open Console Window. Click on hamburger menu next to Console and select Debug Database View. Search for debug flag ***AME***.***ScriptLayer.EnableNewWorld***and change it to ***true*** by selecting the checkbox. New settings will take place on re launch.



# Developer guide

## Generating an API token

For testing purpose you can use the Premiere Pro scripting engine to request a token for the currently logged in user. In production this workflow will be replaced with: <https://www.adobe.io/apis/cloudplatform/console/authentication/createjwt.html>. The integration on Adobe.Io can be created via: <https://console.adobe.io/integrations>.

In the ExtendScript Toolkit connect to Premiere Pro and invoke the following command:

taskQueueManagerObj = qe.tqm.logIntoTaskQueueManager();

var authTokenObj = qe.tqm.getAuthToken();  
var bearerToken = authTokenObj.token;

## Accessing the API

The API can be accessed via the URL: https://[cloud-dispatcher-beta.adobe.io](http://cloud-dispatcher-stage.adobe.io)

The API uses the JSON+HAL format (<https://en.wikipedia.org/wiki/Hypertext_Application_Language>).

It requires the "Authorization" header to be set to "Bearer yourBearerTokenFromExtendScriptHere" and the "X-Api-Key" header set to "task-queue-manager-beta-1".

## Using the Task Queue Manager API from within Premiere Pro (DOM API)

This example script will lookup a queue from the list of existing queues (without creating a new one). A typical workflow would involve one central queue being created by an administrator (assigned to the organisation) which all users can share for submitting work to the central pool of workers.

qe.tqm.logIntoTaskQueueManager();

var queueId = "same-queue-id-as-used-in-the-worker";

var queueList = qe.tqm.getQueueListObject();

var queue = queueList.getQueueByID(queueId);

var job = queue.createRemoteAMERenderJob(

"example encode job 1",

"c:\\full\\path\\to\\team\\projects\\snapshot\\tp-snapshot.tp2snap",

"c:\\full\\path\\to\\team\\projects\\snapshot\\ExportPresetH264MatchSequence.epr",

"c:\\worker\\harddrive\\or\\network\\share\\output.mp4"

);

var result = job.loadSync();

## API

All API calls are scoped to the qe dom.

### Root Level Methods

##### qe.tqm.logIntoTaskQueueManager() - returns Boolean (success)

 used to create an authentication token from the currently logged in user to communicate the the TQM API

##### qe.tqm.loginByRegion(regionName) - returns Boolean (success)

used to create an authentication token from the currently logged in user on the region provided as string parameter to communicate the the TQM API

##### qe.tqm.getActiveRegion(); - returns String

returns the active region user is currently logged in.

##### qe.tqm.getQueueListObject() - returns QueueListObject

creates a request object to read the list of queues visible to the logged in user

##### qe.tqm.createQueue(queueName, ownerId) returns  QueueObject

create a request with the given parameters for creating a new queue, owned by the user (if a blank ownerId is given) or the organisation of the ownerId

##### qe.tqm.getAuthToken() - returns AuthenticationObject

returns the Authentication object which can be then used to get current authentication token using  AuthenticationObject.token(only available after calling the logIntoTaskQueueManager method)

##### qe.tqm.setDiscoveryURL(String) - return Boolean (success)

sets the base discovery URL used to communicating with the api (can e.g. be used to switch to a different region or localhost for debugging)

### Queue List Level Methods

##### queueList.getQueues() returns an array of Queue objects

after the request was loaded (async/sync) the list of queues is available in this array

##### queueList.getQueueByID(queueId) returns a Queue Object

directly read a queue object by queueId

##### queueList.loadAsync() returns Boolean (success)

process the request async (non blocking)

##### queueList.loadSync() returns Boolean (success)

process the request sync (blocking)

### Queue Level Methods

##### queue.identifier - returns a String identifier property

the ID property of the queue

##### queue.name - returns a String name of the queue

the name property of the queue

##### queue.createRemoteAMERenderProjectJob(jobName, localProjectFilePath, itemGUID, localExportPresetPath, remoteOutputFilePath) - returns a Job request object

creates a project file based job and attaches both the export preset and the project to the request object.

##### queue.prepareRemoteAMERenderProjectJob(jobName, localProjectFilePath, itemGUID, localExportPresetPath, remoteOutputFilePath) - returns a Job request object

creates project file based job  that will not be automatically be started and attaches both the export preset and the project to the request object.

##### queue.createRemoteAMERenderJob(jobName, localTeamProjectSnapshotPath, localExportPresetPath, remoteOutputFilePath) - returns a Job request object

creates a team projects snapshot based job and attaches both the snapshot and export preset to the request object.

##### queue.prepareRemoteAMERenderJob(jobName, localTeamProjectSnapshotPath, localExportPresetPath, remoteOutputFilePath) - returns a Job request object

creates a team projects snapshot based job that will not be automatically be started and attaches both the snapshot and export preset to the request object.

##### queue.loadSync() - returns queue id(success)

process the request sync (blocking)

##### queue.loadAsync() - returns a Boolean (success)

process the request async (non-blocking)

### Job Level Methods

##### job.****identifier**** - returns a String identifier property

the ID property of the job

##### job.****name**** - returns a String name property

the Name property of the job

##### job.****created**** - returns a String created property

the creation timestamp of the job in the 'YYYY-MM-DDTHH:MM:SS.XXXZ' format.

##### job.estimatedCompletion- returns a String created property

##### the estimated completion time in the 'YYYY-MM-DDTHH:MM:SS.XXXZ' format.

job.**lastModified** - returns a String lastModified property

the last modification timestamp of the job in the 'YYYY-MM-DDTHH:MM:SS.XXXZ' format.

##### job.****estimatedCompletion**** - returns a String estimatedCompletion property

the estimated-completion timestamp of the job in the 'YYYY-MM-DDTHH:MM:SS.XXXZ' format.

##### job.****state**** - returns a String state property

the current state of the job (i.e WAITING, PROGRESSING, COMPLETED, CANCELED)

##### job.****type**** - returns a String type property

the type of the job. (i.e AME.2020.encode, AE.2020.render, PPRO.2020.ingest, etc)

##### job.****lockId**** - returns a String lockId property

the lockId of the job.

##### job.****progress**** - returns a Float progress property

the current progress value (0-100) of the job.

##### job.****progressState**** - returns a String progressState property

the current state of progress of the job. (i.e. Error, Done, None)

##### job.isCompleted - returns Boolean (success)

status completion of job.(i.e. true, false)

##### job.isError - returns Boolean (success)

status completion if job is in error state or not.(i.e. true, false)

##### job.addInput(name, pathToLocalFile) - returns Boolean (success)

adds an input file the the job - (e.g. if the job is created via prepareRemoteAMERenderJob - the job will not be started and may receive additional input files, and can then be started via the job.start() call)

##### job.start() - returns a Boolean (success)

start the job

##### job.cancel() - returns a Boolean (success)

cancel the job

##### job.loadSync() - returns a job id(success)

process the request sync (blocking)

**job.loadSync(job.identifier) -  take job id as string input and** returns a job id(success) and refreshes the property of job passed as input

process the request sync (blocking)

##### job.loadAsync() - returns a Boolean (success)

process the request async (non-blocking)

### Additional Snapshot Methods

##### app.project.rootItem.children[childNumber].saveProjectSnapshot(snapShotPath) - take childNumber as integer, snapShotPath as string and returns a Boolean (success)

##### app.anywhere.openTeamProjectSnapshot(snapShotPath) - take snapShotPath as string and returns a Boolean (success)

##### project.applyProjectSnapshot(snapShotPath,keepSessionProperties,deleteAssetsNotInSnapshot) - take snapShotPath as string, keepSessionProperties as boolean,  deleteAssetsNotInSnapshot as boolean returns a Boolean (success)

The snapshot contents will be merged into the currently-open Team Project. If keepSessionProperties is set to true, the existing properties will not be replaced; otherwise, the project name and description will be replaced by those of the snapshot. If deleteAssetsNotInSnapshot is set to true, any assets not in the snapshot will be deleted; otherwise, the snapshot will be merged into the existing project (snapshot assets will overwrite any existing identical assets).

# Adobe Task Queue Manager API documentation

# Table of Contents

* [Introduction](#_introduction)
  + [Authentication](#_authentication)
    - [Creating the worker keystore file and certificate](#_creating_the_worker_keystore_file_and_)
* [Resources](#resources)
  + [Discovery Links](#resources-discovery-documentation-test-)
    - [Path parameters](#_path_parameters)
    - [Query parameters](#_query_parameters)
    - [Request fields](#_request_fields)
    - [Response fields](#_response_fields)
    - [Example request](#_example_request)
    - [Example response](#_example_response)
  + [List My Jobs](#resources-queue-documentation-test-list)
    - [Path parameters](#_path_parameters_2)
    - [Query parameters](#_query_parameters_2)
    - [Request fields](#_request_fields_2)
    - [Response fields](#_response_fields_2)
    - [Example request](#_example_request_2)
    - [Example response](#_example_response_2)
  + [Create Input](#resources-queue-documentation-test-crea)
    - [Path parameters](#_path_parameters_3)
    - [Query parameters](#_query_parameters_3)
    - [Request fields](#_request_fields_3)
    - [Response fields](#_response_fields_3)
    - [Example request](#_example_request_3)
    - [Example response](#_example_response_3)
  + [List Workers](#resources-worker-documentation-test-lis)
    - [Path parameters](#_path_parameters_4)
    - [Query parameters](#_query_parameters_4)
    - [Request fields](#_request_fields_4)
    - [Response fields](#_response_fields_4)
    - [Example request](#_example_request_4)
    - [Example response](#_example_response_4)
  + [Delete Worker](#resources-worker-documentation-test-del)
    - [Path parameters](#_path_parameters_5)
    - [Query parameters](#_query_parameters_5)
    - [Request fields](#_request_fields_5)
    - [Response fields](#_response_fields_5)
    - [Example request](#_example_request_5)
    - [Example response](#_example_response_5)
  + [List Queues](#resources-queue-documentation-test-queu)
    - [Path parameters](#_path_parameters_6)
    - [Query parameters](#_query_parameters_6)
    - [Request fields](#_request_fields_6)
    - [Response fields](#_response_fields_6)
    - [Example request](#_example_request_6)
    - [Example response](#_example_response_6)
  + [Find Queue](#resources-queue-documentation-test-get-)
    - [Path parameters](#_path_parameters_7)
    - [Query parameters](#_query_parameters_7)
    - [Request fields](#_request_fields_7)
    - [Response fields](#_response_fields_7)
    - [Example request](#_example_request_7)
    - [Example response](#_example_response_7)
  + [Create Queue](#resources-queue-documentation-test-crea)
    - [Path parameters](#_path_parameters_8)
    - [Query parameters](#_query_parameters_8)
    - [Request fields](#_request_fields_8)
    - [Response fields](#_response_fields_8)
    - [Example request](#_example_request_8)
    - [Example response](#_example_response_8)
  + [Delete Queue](#resources-queue-documentation-test-dele)
    - [Path parameters](#_path_parameters_9)
    - [Query parameters](#_query_parameters_9)
    - [Request fields](#_request_fields_9)
    - [Response fields](#_response_fields_9)
    - [Example request](#_example_request_9)
    - [Example response](#_example_response_9)
  + [List Jobs By Queue](#resources-queue-documentation-test-list)
    - [Path parameters](#_path_parameters_10)
    - [Query parameters](#_query_parameters_10)
    - [Request fields](#_request_fields_10)
    - [Response fields](#_response_fields_10)
    - [Example request](#_example_request_10)
    - [Example response](#_example_response_10)
  + [Listen For Queue Job Updates](#resources-queue-documentation-test-list)
    - [Path parameters](#_path_parameters_11)
    - [Query parameters](#_query_parameters_11)
    - [Request fields](#_request_fields_11)
    - [Response fields](#_response_fields_11)
    - [Example request](#_example_request_11)
    - [Example response](#_example_response_11)
  + [Delete Job](#resources-queue-documentation-test-dele)
    - [Path parameters](#_path_parameters_12)
    - [Query parameters](#_query_parameters_12)
    - [Request fields](#_request_fields_12)
    - [Response fields](#_response_fields_12)
    - [Example request](#_example_request_12)
    - [Example response](#_example_response_12)
  + [Create Team Project Encode Job On Queue](#resources-queue-documentation-test-crea)
    - [Path parameters](#_path_parameters_13)
    - [Query parameters](#_query_parameters_13)
    - [Request fields](#_request_fields_13)
    - [Response fields](#_response_fields_13)
    - [Example request](#_example_request_13)
    - [Example response](#_example_response_13)
  + [Create Project File Encode Job On Queue](#resources-queue-documentation-test-crea)
    - [Path parameters](#_path_parameters_14)
    - [Query parameters](#_query_parameters_14)
    - [Request fields](#_request_fields_14)
    - [Response fields](#_response_fields_14)
    - [Example request](#_example_request_14)
    - [Example response](#_example_response_14)
  + [Create Premiere Pro Script Job On Queue](#resources-queue-documentation-test-crea)
    - [Path parameters](#_path_parameters_15)
    - [Query parameters](#_query_parameters_15)
    - [Request fields](#_request_fields_15)
    - [Response fields](#_response_fields_15)
    - [Example request](#_example_request_15)
    - [Example response](#_example_response_15)
  + [Create Team Project Script Job On Queue](#resources-queue-documentation-test-crea)
    - [Path parameters](#_path_parameters_16)
    - [Query parameters](#_query_parameters_16)
    - [Request fields](#_request_fields_16)
    - [Response fields](#_response_fields_16)
    - [Example request](#_example_request_16)
    - [Example response](#_example_response_16)
  + [Create Premiere Pro Ingest Job On Queue](#resources-queue)
    - [Path parameters](#_path_parameters_17)
    - [Query parameters](#_query_parameters_17)
    - [Request fields](#_request_fields_17)
    - [Response fields](#_response_fields_17)
    - [Example request](#_example_request_17)
    - [Example response](#_example_response_17)
  + [Create Premiere Pro OMF Conversion Job On Queue](#resources-queue)
    - [Path parameters](#_path_parameters_18)
    - [Query parameters](#_query_parameters_18)
    - [Request fields](#_request_fields_18)
    - [Response fields](#_response_fields_18)
    - [Example request](#_example_request_18)
    - [Example response](#_example_response_18)
  + [Create Premiere Pro AAF Conversion Job On Queue](#resources-queue)
    - [Path parameters](#_path_parameters_19)
    - [Query parameters](#_query_parameters_19)
    - [Request fields](#_request_fields_19)
    - [Response fields](#_response_fields_19)
    - [Example request](#_example_request_19)
    - [Example response](#_example_response_19)
  + [Patch Job](#resources-queue-documentation-test-canc)
    - [Path parameters](#_path_parameters_20)
    - [Query parameters](#_query_parameters_20)
    - [Request fields](#_request_fields_20)
    - [Response fields](#_response_fields_20)
    - [Example request](#_example_request_20)
    - [Example response](#_example_response_20)

# [Introduction](#_introduction)

# Getting started with the Adobe Task Queue Manager API. The API uses the HAL JSON format: <http://stateless.co/hal_specification.html>. The API can be navigated by following the links starting from the discovery endpoint.

## [Authentication](#_authentication)

The API supports authentication through a bearer token header and an api key. This can be generated via a JWT token workflow using the [adobe.io](http://adobe.io) integration: <https://www.adobe.io/apis/cloudplatform/console/authentication/gettingstarted.html> . Setting both the 'Authorization' and the 'X-Api-Key' header is required for every request.

### [Creating the worker keystore file and certificate](#_creating_the_worker_keystore_file_and_)

openssl req -nodes -text -x509 -newkey rsa:2048 -keyout secret.pem -out certificate.pem -days 356

openssl pkcs8 -topk8 -inform PEM -outform DER -in secret.pem -nocrypt > secret.key

openssl pkcs12 -export -in certificate.pem -inkey secret.pem -out my.p12

keytool -importkeystore \

-deststorepass 123456 -destkeypass 123456 -destkeystore keystore.ks \

-srckeystore my.p12 -srcstoretype PKCS12 -srcstorepass 123456

keytool -changealias -keystore keystore.ks -alias 1 -destalias AdobePrivateKey

# [Resources](#resources)

### [Discovery Links](#resources-discovery-documentation-test-)

GET /

Returns the list of all links available on the root API.

#### [Path parameters](#_path_parameters)

No parameters.

#### [Query parameters](#_query_parameters)

No parameters.

#### [Request fields](#_request_fields)

No request body.

#### [Response fields](#_response_fields)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| activeRegion | String | true |  |

#### [Example request](#_example_request)

$ curl 'https://cloud-dispatcher-beta.adobe.io/' -i -X GET \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token'

#### [Example response](#_example_response)

HTTP/1.1 200 OK

Content-Type: application/hal+json;charset=UTF-8

Content-Length: 797

{ "activeRegion" : "region-unknown", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io" }, "jobs" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/{?page,size}", "templated" : true }, "queues" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/{?page,size}", "templated" : true }, "workers" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/workers/{?page,size}", "templated" : true }, "region-ue1" : { "href" : "https://cloud-dispatcher-beta-ue1.adobe.io" }, "region-ew1" : { "href" : "https://cloud-dispatcher-beta-ew1.adobe.io" }, "region-an1" : { "href" : "https://cloud-dispatcher-beta-an1.adobe.io" } } }

### [List My Jobs](#resources-queue-documentation-test-list)

GET /api/v1/jobs/

#### [Path parameters](#_path_parameters_2)

No parameters.

#### [Query parameters](#_query_parameters_2)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| page | Integer | true | Default value: '0'. |
| size | Integer | true | Default value: '50'. |

#### [Request fields](#_request_fields_2)

No request body.

#### [Response fields](#_response_fields_2)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| content | Array[Object] | true |  |
| content[].lastModified | String | false |  |
| content[].progressState | String | true |  |
| content[].estimatedCompletion | String | true |  |
| content[].created | String | false |  |
| content[].lockId | String | true |  |
| content[].progress | Decimal | true |  |
| content[].name | String | true |  |
| content[].state | String | false |  |
| content[].type | String | false |  |
| content[].id | String | false |  |
| page | Object | true |  |
| page.size | Integer | true |  |
| page.totalElements | Integer | true |  |
| page.totalPages | Integer | true |  |
| page.number | Integer | true |  |

#### [Example request](#_example_request_2)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/' -i -X GET \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token'

#### [Example response](#_example_response_2)

HTTP/1.1 200 OK

Content-Length: 2078

Content-Type: application/hal+json;charset=UTF-8

{ "\_embedded" : { "jobResourceList" : [ { "lastModified" : "2019-09-18T08:37:58.577138Z", "progressState" : null, "estimatedCompletion" : null, "created" : "2019-09-18T08:37:58.577133Z", "lockId" : "workerId", "progress" : 0.0, "name" : "Encode Project to MP4", "state" : "WAITING", "type" : "AME.2020.encode", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573" }, "inputs" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573/inputs" }, "inputs-presign" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573/inputs/presign" }, "results" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573/results" }, "results-presign" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573/results/presign" }, "waitForCancel" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573/waitFor/canceled" }, "job-updates" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2e46f6a6-a933-4192-b609-d3bb40242573/updatedSince/2019-09-18T08:37:58.577138Z" } }, "id" : "2e46f6a6-a933-4192-b609-d3bb40242573" } ] }, "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/?page=0&size=50&sort=lastModified,desc" }, "byId" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/{id}", "templated" : true }, "updates" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/since/2019-09-18T08:37:58.577138Z/" } }, "page" : { "size" : 50, "totalElements" : 1, "totalPages" : 1, "number" : 0 } }

### [Create Input](#resources-queue-documentation-test-crea)

POST /api/v1/jobs/{id}/inputs/presign

#### [Path parameters](#_path_parameters_3)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| id | Object | false |  |

#### [Query parameters](#_query_parameters_3)

No parameters.

#### [Request fields](#_request_fields_3)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| payloads | Array[Object] | true |  |
| payloads[].index | Integer | true |  |
| payloads[].length | Integer | true |  |
| payloads[].name | String | true |  |

#### [Response fields](#_response_fields_3)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| signedPayloadPairs | Array[Object] | true |  |
| signedPayloadPairs[].index | Integer | true |  |
| signedPayloadPairs[].requestIndex | Integer | true |  |
| signedPayloadPairs[].signedUploadURL | String | true |  |

#### [Example request](#_example_request_3)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/dd7d9b8e-f315-45dc-af9d-641f07e4b6e1/inputs/presign' -i -X POST \

-H 'Content-Type: application/json' \

-d '{"payloads":[{"index":1,"length":123,"name":"input.prproj"}]}'

#### [Example response](#_example_response_3)

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

Content-Length: 124

{ "signedPayloadPairs" : [ { "index" : 0, "requestIndex" : 1, "signedUploadURL" : "https://signed-url" } ] }

### [List Workers](#resources-worker-documentation-test-lis)

GET /api/v1/workers/

#### [Path parameters](#_path_parameters_4)

No parameters.

#### [Query parameters](#_query_parameters_4)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| page | Integer | true | Default value: '0'. |
| size | Integer | true | Default value: '50'. |

#### [Request fields](#_request_fields_4)

No request body.

#### [Response fields](#_response_fields_4)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| content | Array[Object] | true |  |
| content[].lastModified | String | true |  |
| content[].totalProcessed | Integer | true |  |
| content[].name | String | true |  |
| content[].state | String | true |  |
| content[].owner | String | true |  |
| content[].id | String | true |  |
| page | Object | true |  |
| page.size | Integer | true |  |
| page.totalElements | Integer | true |  |
| page.totalPages | Integer | true |  |
| page.number | Integer | true |  |

#### [Example request](#_example_request_4)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/workers/' -i -X GET

#### [Example response](#_example_response_4)

HTTP/1.1 200 OK

Content-Type: application/hal+json;charset=UTF-8

Content-Length: 848

{ "\_embedded" : { "workerResourceList" : [ { "lastModified" : "2019-09-18T08:38:02.514370Z", "totalProcessed" : 0, "name" : "Media Encoder Cluster Worker 1", "state" : "IDLE", "owner" : "orgid@AdobeOrg", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/workers/f845dca5-19bf-4939-8e1c-d891080e3ed0" } }, "id" : "f845dca5-19bf-4939-8e1c-d891080e3ed0" } ] }, "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/workers/?page=0&size=50&sort=lastModified,desc" }, "byId" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/workers{/id}", "templated" : true } }, "page" : { "size" : 50, "totalElements" : 1, "totalPages" : 1, "number" : 0 } }

### [Delete Worker](#resources-worker-documentation-test-del)

DELETE /api/v1/workers/{id}

Deletes the worker activity data.

#### [Path parameters](#_path_parameters_5)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| id | Object | false |  |

#### [Query parameters](#_query_parameters_5)

No parameters.

#### [Request fields](#_request_fields_5)

No request body.

#### [Response fields](#_response_fields_5)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| lastModified | String | true |  |
| totalProcessed | Integer | true |  |
| name | String | true |  |
| state | String | true |  |
| owner | String | true |  |
| id | String | true |  |

#### [Example request](#_example_request_5)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/workers/f895c6d6-4709-45b7-babe-06d93ff573a5' -i -X DELETE \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token' \

-H 'Content-Type: application/json;charset=UTF-8'

#### [Example response](#_example_response_5)

HTTP/1.1 204 No Content

### [List Queues](#resources-queue-documentation-test-queu)

GET /api/v1/queues

#### [Path parameters](#_path_parameters_6)

No parameters.

#### [Query parameters](#_query_parameters_6)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| page | Integer | true | Default value: '0'. |
| size | Integer | true | Default value: '50'. |

#### [Request fields](#_request_fields_6)

No request body.

#### [Response fields](#_response_fields_6)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| content | Array[Object] | true |  |
| content[].name | String | true |  |
| content[].owner | String | false |  |
| content[].id | String | false |  |
| page | Object | true |  |
| page.size | Integer | true |  |
| page.totalElements | Integer | true |  |
| page.totalPages | Integer | true |  |
| page.number | Integer | true |  |

#### [Example request](#_example_request_6)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues' -i -X GET \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token'

#### [Example response](#_example_response_6)

HTTP/1.1 200 OK

Content-Type: application/hal+json;charset=UTF-8

Content-Length: 1289

{ "\_embedded" : { "queueResourceList" : [ { "name" : "Media Encoder Queue Hamburg", "owner" : "example ORG", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/6fd0e4da-dd03-4a05-be99-848b7f71c09a" }, "listen" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/6fd0e4da-dd03-4a05-be99-848b7f71c09a/listen" }, "progressingBy" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/6fd0e4da-dd03-4a05-be99-848b7f71c09a/jobs/progressingBy/{lockId}{?page,size}", "templated" : true }, "jobs" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/6fd0e4da-dd03-4a05-be99-848b7f71c09a/jobs{?page,size}", "templated" : true } }, "id" : "6fd0e4da-dd03-4a05-be99-848b7f71c09a" } ] }, "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues?page=0&size=50&sort=lastModified,desc" }, "byId" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues{/id}", "templated" : true } }, "page" : { "size" : 50, "totalElements" : 1, "totalPages" : 1, "number" : 0 } }

### [Find Queue](#resources-queue-documentation-test-get-)

GET /api/v1/queues/{id}

#### [Path parameters](#_path_parameters_7)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| id | Object | false |  |

#### [Query parameters](#_query_parameters_7)

No parameters.

#### [Request fields](#_request_fields_7)

No request body.

#### [Response fields](#_response_fields_7)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| name | String | true |  |
| owner | String | false |  |
| id | String | false |  |

#### [Example request](#_example_request_7)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/b0ef78d4-bd8e-4979-bd04-2ae76f10cc69' -i -X GET \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token'

#### [Example response](#_example_response_7)

HTTP/1.1 200 OK

Content-Type: application/hal+json;charset=UTF-8

Content-Length: 782

{ "name" : "Media Encoder Queue Hamburg", "owner" : "example ORG", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/b0ef78d4-bd8e-4979-bd04-2ae76f10cc69" }, "listen" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/b0ef78d4-bd8e-4979-bd04-2ae76f10cc69/listen" }, "progressingBy" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/b0ef78d4-bd8e-4979-bd04-2ae76f10cc69/jobs/progressingBy/{lockId}{?page,size}", "templated" : true }, "jobs" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/b0ef78d4-bd8e-4979-bd04-2ae76f10cc69/jobs{?page,size}", "templated" : true } }, "id" : "b0ef78d4-bd8e-4979-bd04-2ae76f10cc69" }

### [Create Queue](#resources-queue-documentation-test-crea)

POST /api/v1/queues

#### [Path parameters](#_path_parameters_8)

No parameters.

#### [Query parameters](#_query_parameters_8)

No parameters.

#### [Request fields](#_request_fields_8)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| name | String | false |  |
| owner | String | true |  |

#### [Response fields](#_response_fields_8)

No response body.

#### [Example request](#_example_request_8)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token' \

-H 'Content-Type: application/json' \

-d '{"name":"Media Encoder Queue Munich","owner":"orgId@AdobeOrg"}'

#### [Example response](#_example_response_8)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/queues/7f269d3e-2f21-4e4f-bb43-7beecb8694e3

### [Delete Queue](#resources-queue-documentation-test-dele)

DELETE /api/v1/queues/{id}

#### [Path parameters](#_path_parameters_9)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| id | Object | false |  |

#### [Query parameters](#_query_parameters_9)

No parameters.

#### [Request fields](#_request_fields_9)

No request body.

#### [Response fields](#_response_fields_9)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| name | String | true |  |
| owner | String | false |  |
| id | String | false |  |

#### [Example request](#_example_request_9)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/36b9e5bd-5883-435d-af54-549b45b52368' -i -X DELETE \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token' \

-H 'Content-Type: application/json' \

-d '{"name":"Media Encoder Queue Munich","owner":"orgId@AdobeOrg"}'

#### [Example response](#_example_response_9)

HTTP/1.1 204 No Content

### [List Jobs By Queue](#resources-queue-documentation-test-list)

GET /api/v1/queues/{queueId}/jobs/

#### [Path parameters](#_path_parameters_10)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_10)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| page | Integer | true | Default value: '0'. |
| size | Integer | true | Default value: '50'. |

#### [Request fields](#_request_fields_10)

No request body.

#### [Response fields](#_response_fields_10)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| content | Array[Object] | true |  |
| content[].lastModified | String | false |  |
| content[].progressState | String | true |  |
| content[].estimatedCompletion | String | true |  |
| content[].created | String | false |  |
| content[].lockId | String | true |  |
| content[].progress | Decimal | true |  |
| content[].name | String | true |  |
| content[].state | String | false |  |
| content[].type | String | false |  |
| content[].id | String | false |  |
| page | Object | true |  |
| page.size | Integer | true |  |
| page.totalElements | Integer | true |  |
| page.totalPages | Integer | true |  |
| page.number | Integer | true |  |

#### [Example request](#_example_request_10)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/8287a220-4669-4735-8a0b-d4a4765c50f0/jobs/' -i -X GET \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token'

#### [Example response](#_example_response_10)

HTTP/1.1 200 OK

Content-Length: 2051

Content-Type: application/hal+json;charset=UTF-8

{ "\_embedded" : { "jobResourceList" : [ { "lastModified" : "2019-09-18T08:37:58.661209Z", "progressState" : null, "estimatedCompletion" : null, "created" : "2019-09-18T08:37:58.661202Z", "lockId" : "workerId", "progress" : 0.0, "name" : "Encode Project to MP4", "state" : "WAITING", "type" : "AME.2020.encode", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798" }, "inputs" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798/inputs" }, "inputs-presign" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798/inputs/presign" }, "results" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798/results" }, "results-presign" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798/results/presign" }, "waitForCancel" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798/waitFor/canceled" }, "job-updates" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/79034e66-8d82-4506-a353-bd98d2e1a798/updatedSince/2019-09-18T08:37:58.661209Z" } }, "id" : "79034e66-8d82-4506-a353-bd98d2e1a798" } ] }, "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/8287a220-4669-4735-8a0b-d4a4765c50f0/jobs/?page=0&size=50&sort=lastModified,desc" }, "queue-updates" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/8287a220-4669-4735-8a0b-d4a4765c50f0/jobs/since/2019-09-18T08:37:58.667906Z/" } }, "page" : { "size" : 50, "totalElements" : 1, "totalPages" : 1, "number" : 0 } }

### [Listen For Queue Job Updates](#resources-queue-documentation-test-list)

GET /api/v1/queues/{queueId}/jobs/since/{since}/

#### [Path parameters](#_path_parameters_11)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |
| since | Object | false |  |

#### [Query parameters](#_query_parameters_11)

No parameters.

#### [Request fields](#_request_fields_11)

No request body.

#### [Response fields](#_response_fields_11)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| result | Object | true |  |
| setOrExpired | Boolean | true |  |

#### [Example request](#_example_request_11)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/fe2e5e87-3dc5-4a7e-a034-8af03be4f22d/jobs/since/2019-09-18T08:37:58.038777Z/' -i -X GET \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token'

#### [Example response](#_example_response_11)

HTTP/1.1 200 OK

Content-Type: application/hal+json;charset=UTF-8

Content-Length: 1779

{ "\_embedded" : { "jobResourceList" : [ { "lastModified" : "2019-09-18T08:37:58.038573Z", "progressState" : null, "estimatedCompletion" : null, "created" : "2019-09-18T08:37:58.038568Z", "lockId" : "workerId", "progress" : 0.0, "name" : "Encode Project to MP4", "state" : "WAITING", "type" : "AME.2020.encode", "\_links" : { "self" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b" }, "inputs" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b/inputs" }, "inputs-presign" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b/inputs/presign" }, "results" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b/results" }, "results-presign" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b/results/presign" }, "waitForCancel" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b/waitFor/canceled" }, "job-updates" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/01d35155-5f34-426e-b243-0187fbf9357b/updatedSince/2019-09-18T08:37:58.038573Z" } }, "id" : "01d35155-5f34-426e-b243-0187fbf9357b" } ] }, "\_links" : { "queue-updates" : { "href" : "https://cloud-dispatcher-beta.adobe.io/api/v1/queues/fe2e5e87-3dc5-4a7e-a034-8af03be4f22d/jobs/since/2019-09-18T08:37:58.038573Z/" } } }

### [Delete Job](#resources-queue-documentation-test-dele)

DELETE /api/v1/jobs/{id}

#### [Path parameters](#_path_parameters_12)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| id | Object | false |  |

#### [Query parameters](#_query_parameters_12)

No parameters.

#### [Request fields](#_request_fields_12)

No request body.

#### [Response fields](#_response_fields_12)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| lastModified | String | false |  |
| progressState | String | true |  |
| estimatedCompletion | String | true |  |
| created | String | false |  |
| lockId | String | true |  |
| progress | Decimal | true |  |
| name | String | true |  |
| state | String | false |  |
| type | String | false |  |
| id | String | false |  |

#### [Example request](#_example_request_12)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/1fe0c0ea-9193-4ea9-a51b-824241e309a1' -i -X DELETE \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token' \

-H 'Content-Type: application/json;charset=UTF-8'

#### [Example response](#_example_response_12)

HTTP/1.1 204 No Content

### [Create Team Project Encode Job On Queue](#resources-queue-documentation-test-crea)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_13)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_13)

No parameters.

#### [Request fields](#_request_fields_13)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| jobParameters.js | The parameters passed to the worker, specific to the job type |
| project.tp2snap | The binary Team Projects snapshot |
| exportPreset.epr | The binary export preset |

#### [Response fields](#_response_fields_13)

No response body.

#### [Example request](#_example_request_13)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/df4e5944-3d3a-45c3-aa27-478cdc42f629/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"name":"Team Projects Export","type":"AME.2020.encode"}' \

-F 'jobParameters.js={"outputFile":"/worker/relative/output/directory/file.mp4","method":"addTeamProjectsItemToBatch"}' \

-F 'project.tp2snap=<<project.tp2snap data>>' \

-F 'exportPreset.epr=<<exportPreset.epr data>>'

#### [Example response](#_example_response_13)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/c455834c-b3af-4e5b-91db-93d8878ce81f

### [Create Project File Encode Job On Queue](#resources-queue-documentation-test-crea)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_14)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_14)

No parameters.

#### [Request fields](#_request_fields_14)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| jobParameters.js | The parameters passed to the worker, specific to the job type |
| project.prproj | The binary Project file |
| exportPreset.epr | The binary export preset |

#### [Response fields](#_response_fields_14)

No response body.

#### [Example request](#_example_request_14)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/90f3b20c-55f6-425b-9ad8-72e76fdc510d/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"name":"Project File Export","type":"AME.2020.encode"}' \

-F 'jobParameters.js={"outputFile":"/worker/relative/output/directory/file.mp4","method":"addDLToBatch","itemGUID":"<optional sequenceId>"}' \

-F 'project.prproj=<<project.prproj data>>' \

-F 'exportPreset.epr=<<exportPreset.epr data>>'

#### [Example response](#_example_response_14)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/4925d5ab-5b06-4e45-8f36-78c842aa046f

### [Create Premiere Pro Script Job On Queue](#resources-queue-documentation-test-crea)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_15)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_15)

No parameters.

#### [Request fields](#_request_fields_15)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| script.jsx | The extendscript code to execute on the attached project. The project is saved as a job result if the project was modified |
| project.prproj | The binary Project file |
| exportPreset.epr | The binary export preset |

#### [Response fields](#_response_fields_15)

No response body.

#### [Example request](#_example_request_15)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/8e99ca89-2a2e-46ad-8e64-4d095f90f188/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"name":"Premiere Pro Script","type":"PPRO.2020.script"}' \

-F 'script.jsx=<<script.jsx data>>' \

-F 'project.prproj=<<project.prproj data>>' \

-F 'exportPreset.epr=<<exportPreset.epr data>>'

#### [Example response](#_example_response_15)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/1fa36ae9-9025-4316-9762-5b5be2c22e5f

### [Create Team Project Script Job On Queue](#resources-queue-documentation-test-crea)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_16)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_16)

No parameters.

#### [Request fields](#_request_fields_16)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| script.jsx | The extendscript code to execute on the attached project. The project is saved as a job result if the project was modified |
| snapshot.tp2snap | The binary Team Projects Snapshot file |

#### [Response fields](#_response_fields_16)

No response body.

#### [Example request](#_example_request_16)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/c1d7e767-11d0-445b-b5ca-4c6f66853a5d/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"name":"Team Project Script","type":"PPRO.2020.script"}' \

-F 'script.jsx=<<script.jsx data>>' \

-F 'snapshot.tp2snap=<<snapshot.tp2snap data>>'

#### [Example response](#_example_response_16)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/8cc9e1d5-c751-4350-b101-cc14221e36b5

### [Create Premiere Pro Ingest Job On Queue](#resources-queue)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_17)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_17)

No parameters.

#### [Request fields](#_request_fields_17)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| jobParameters.js | The parameters passed to the worker, specific to the job type |
| project.prproj | The binary Project file |

#### [Response fields](#_response_fields_17)

No response body.

#### [Example request](#_example_request_17)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/9bf5f2d0-eeaa-4318-a572-42f7b929205f/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"name":"Premiere Pro Ingest","type":"PPRO.2020.ingest"}' \

-F 'jobParameters.js={"files":["/path/to/ingest/file.mp4","/other/path/to/ingest/file.mov"]}' \

-F 'project.prproj=<<project.prproj data>>'

#### [Example response](#_example_response_17)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/2652083f-362a-441f-89b5-7977d42a8c36

### [Create Premiere Pro OMF Conversion Job On Queue](#resources-queue)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_18)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_18)

No parameters.

#### [Request fields](#_request_fields_18)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| script.jsx | The extendscript code to execute on the attached project. The project is saved as a job result if the project was modified |
| project.prproj | The binary Project file |

#### [Response fields](#_response_fields_18)

No response body.

#### [Example request](#_example_request_18)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/39ecda56-974f-4dc8-9e64-9ed8f456ddb0/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"handleFramesOut":"if trim is 1, handle frames from 0 to 1000","outputFile":"\\\\a-network-drive\\folder\\file","bitsPerSample":"16 or 24","encapsulation":true,"itemGUID":"64e9277d-706e-4821-8740-b8c4a5a914b0","trimAudio":true,"name":"Premiere Pro Script","format":"WAV or AIFF","type":"PPRO.2020.omf-conversion","title":"OMF Title","handeFramesIn":"if trim is 1, handle frames from 0 to 1000","sampleRate":"48000 or 96000"}' \

-F 'script.jsx=<<script.jsx data>>' \

-F 'project.prproj=<<project.prproj data>>'

#### [Example response](#_example_response_18)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/1261969c-59f6-4bee-943c-17fa97557c43

### [Create Premiere Pro AAF Conversion Job On Queue](#resources-queue)

POST /api/v1/queues/{queueId}/jobs

#### [Path parameters](#_path_parameters_19)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| queueId | Object | false |  |

#### [Query parameters](#_query_parameters_19)

No parameters.

#### [Request fields](#_request_fields_19)

| **Part** | **Description** |
| --- | --- |
| jobDetails | The job properties |
| script.jsx | The extendscript code to execute on the attached project. The project is saved as a job result if the project was modified |
| project.prproj | The binary Project file |

#### [Response fields](#_response_fields_19)

No response body.

#### [Example request](#_example_request_19)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/queues/41b2a254-29b1-4598-9a9b-46ca9f7d754b/jobs' -i -X POST \

-H 'X-Api-Key: your-api-key' \

-H 'Content-Type: multipart/form-data' \

-H 'Authorization: Bearer the-access-token' \

-F 'jobDetails={"mixDownVideo":true,"format":"WAV or AIFF","embedAudio":true,"trimSources":true,"preset":"optional preset","type":"PPRO.2020.aaf-conversion","sampleRate":"48000 or 96000","explodeToMono":true,"outputFile":"\\\\a-network-drive\\folder\\file","bitsPerSample":"16 or 24","itemGUID":"2fa0cf57-f9ac-4486-9bd8-7f9d34ae6068","handleFrames":"number of 'handle' frames","name":"Premiere Pro AAF Conversion"}' \

-F 'script.jsx=<<script.jsx data>>' \

-F 'project.prproj=<<project.prproj data>>'

#### [Example response](#_example_response_19)

HTTP/1.1 201 Created

Location: https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/f3970f8a-f154-4785-8ec3-53ec6fb68077

### [Patch Job](#resources-queue-documentation-test-canc)

PATCH /api/v1/jobs/{id}

#### [Path parameters](#_path_parameters_20)

| **Parameter** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| id | Object | false |  |

#### [Query parameters](#_query_parameters_20)

No parameters.

#### [Request fields](#_request_fields_20)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| estimatedCompletion | String | true |  |
| progress | Decimal | true |  |
| progressState | String | true |  |
| state | String | true |  |

#### [Response fields](#_response_fields_20)

| **Path** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| lastModified | String | false |  |
| progressState | String | true |  |
| estimatedCompletion | String | true |  |
| created | String | false |  |
| lockId | String | true |  |
| progress | Decimal | true |  |
| name | String | true |  |
| state | String | false |  |
| type | String | false |  |
| id | String | false |  |

#### [Example request](#_example_request_20)

$ curl 'https://cloud-dispatcher-beta.adobe.io/api/v1/jobs/55771b5e-455b-4c60-9164-5b24367887b7' -i -X PATCH \

-H 'X-Api-Key: your-api-key' \

-H 'Authorization: Bearer the-access-token' \

-H 'Content-Type: application/json;charset=UTF-8' \

-d '{"state":"CANCELED"}'

#### [Example response](#_example_response_20)

HTTP/1.1 204 No Content