

MainConcept 2GO REST API 2.3

User Guide

Contents

1. Introduction.....	2
2. REST API Description.....	2
3. REST API General.....	3
4. Keeping Task Information	3
5. Service Status.....	3
6. Jobs and Tasks	4
6.1 Job/Task Status.....	4
6.2 Task Type.....	4
6.3 API Function – Container Information	5
6.4 API Function – Container Shutdown	5
6.5 API function - stop processing	6
6.6 API function - start processing.....	6
6.8 API function - list of jobs	8
6.9 API function - remove job from queue.....	8
6.10 API function - job information.....	9
6.11 API function - job progress	10
6.12 API function - job status	11
6.13 API function - job log	11
6.14 API function - abort running job.....	12
7. Technical Support.....	12
8. Copyright Notice.....	12

1. Introduction

The MainConcept 2GO REST API is introduced with version 2.0 and provides users an interface that is more suitable for integration with their existing environment or tools. This API covers functionality to create jobs and query their status using standard REST HTTP clients.

2. REST API Description

When launched without specific processing details or a configuration file, MainConcept 2GO internally starts a web service providing the REST API. This API is used for several functions:

Function	Description
GET /service	Get container status information.
DELETE /service	Shutdown container.
POST /service/stop	Stop internal queue processing.
POST /service/start	Start internal queue processing.
POST /jobs	Add a new job to the internal queue.
GET /jobs	Get the list of all jobs from the internal queue.
DELETE /jobs/{jobID}	Removes a job with specific ID from the internal queue.
GET /jobs/{jobID}	Get all information about specific job.
GET /jobs/{jobID}/progress	Get the progress of the specific job.
GET /jobs/{jobID}/status	Get a status of the specific job.
GET /jobs/{jobID}/log	Get a log of the specific job.
POST /jobs/{jobID}/abort	Abort the specific job.



3. REST API General

The REST API does not require authentication. The functions use the following syntax:

http://[container-ip-address]:[port]/rest/[api-version]/[function]/[parameters]

The "api-version" part in the URL only contains the major version, and with a leading "v". For MainConcept 2GO v2.0 the API version is "v1".

4. Keeping Task Information

The MainConcept 2GO container stores the job history only for jobs that have been started while the container was active. The functions */jobs/{jobID}/status*, */jobs/{jobID}/progress* and */jobs/{jobID}/Log* return an error if the specified jobID was created by a different or previous container instance.

Each MainConcept 2GO container maintains its own job queue that works in FIFO mode. New jobs are added at the end of the queue. Jobs can be removed from the queue. The oldest job in the list is automatically started for processing.

All job information is lost when the container is shut down.

5. Service Status

A 2GO container service can be in a STOPPED or a STARTED status. It is in STARTED mode by default if not specified otherwise at container startup time. When in STARTED mode, the oldest job in the job queue will be processed, followed by the next in the queue, and so on. When in STOPPED mode, only the currently active job (if any) will finish processing, and no new jobs will be started.

Status value	Description
STOPPED	Service is stopped. No queue processing.
STARTED	Service is started. Queue processing.



6. Jobs and Tasks

Jobs are divided into tasks. A job can have multiple tasks that make up the job. A Typical tasks actions are: source file download, transcoding, and output file transfer.

6.1 Job/Task Status

Jobs and tasks in the queue can have multiple different status values.

Status value	Description
PENDING	Job/Task is not started yet.
DONE	Job/Task was finished successfully.
ABORTED	Job/Task is aborted by user.
PROCESSING	Job/Task is currently processing.
ERROR	Job/Task processing resulted in error.

6.2 Task Type

Tasks in jobs can have following types.

Type value	Description
DOWNLOADER	This task downloads content.
EXECUTOR	This task processes content (transcoding, encoding, converting, etc.).
UPLOADER	This task uploads content.



6.3 API Function – Container Information

Returns container service information.

GET /service

Return

HTTP return code	Response Body	Info
200 (ok)	<pre>{ "name": "<mc2go module name>", "status": "STOPPED STARTED", "version": "<mc2go version>" }</pre>	
500 (Internal Server Error)		An error occurred getting the service info.

6.4 API Function – Container Shutdown

Shuts down the container. If a job is currently active, the container is not shut down and this function is ignored.

DELETE /service

Return

HTTP return code	Response Body	Info
200 (ok)		
405 (Method Not Allowed)		A job is currently active and this request was ignored.



6.5 API function - stop processing

Stop processing the job queue, i.e. no new jobs will be processed. New jobs can still be added to the queue. If a job is currently active, it will be finished normally.

POST `/service/stop`

Return

HTTP return code	Response Body	Info
200 (ok) service stopped		Service stopped
500 (Internal Server Error)		An error occurred stopping the service.

6.6 API function - start processing

Start processing the jobs in the queue. The container is in this mode by default. This function is only needed after `/service/stop` was called.

POST `/service/start`

Return

HTTP return code	Response Body	Info
200 (ok) service started		Service started
500 (Internal Server Error)		An error occurred starting the service.



6.7 API function - add job

Adds a new job to the job queue. There are two ways to add jobs: either by posting a JSON file with a job, or a single job can also be posted using just URL parameters.

POST /jobs

A job description file is posted to the endpoint. It must be in JSON format. This JSON file must contain all of the necessary parameters required for the submitted job.

Jobs description file structure:

```
{
  "INPUT": "ftp://10.144.41.202:2121/test.mp4",
  "OUTPUT": "ftp://10.144.41.202:2121/test/xavc_intra.mxf",
  "PRESETNAME": "XAVC_LongGOP_4K_422",
  "KEEP_CONTENT": "TRUE",
  "VERBOSITY": "DEFAULT"
}
```

POST /jobs?param1=value1¶m2=value2¶m3=value3...

All configuration parameters must be submitted using URL parameters.

Return

HTTP return code	Response body	Info
201 (Created)	{ "jobID": "XXYY", }	Job(s) were submitted successfully.
400 (Bad request)		Error occurred while creating the new job. Possible errors: There are no mandatory fields INPUT/OUTPUT Malformed JSON format

6.8 API function - list of jobs

Get the list of jobs in the container's job queue. The optional parameter "status" is used to filter for specific jobs.

GET /jobs?status=<status>

Return

HTTP return code	Response body	Info
200 (ok)	<pre>{ "jobs": [{ "jobID": "XXYY", }] }</pre>	
400 (Bad Request)		Query string is illegal.
404 (Not Found)		Status parameter exists but query value is unknown.
Parameters	Status (optional)	Get only the jobs matching the specified status. Possible values are: DONE ABORTED PENDING PROCESSING ERROR

6.9 API function - remove job from queue

Removes job from the job queue. However, it does not remove a job in PROCESSING state.

DELETE /jobs/{jobID}

Return

HTTP return code	Response Body	Info
200 (ok)		Job removed successfully.
404 (Not found)		Job not found.
403 (Forbidden)		Job is active and cannot be removed.

6.10 API function - job information

Returns job detail information.

GET /jobs/{jobID}

Return

HTTP code	Response body	Info
200 (ok)	<pre>{ "jobID": "XXYY", "status": "ERROR PENDING PROCESSING DONE ABORTED", "parameters": { "param1": "value1", "param2": "value2" }, ... }, "Task_status": [{ "Status": "status", "Task_type": "DOWNLOADER EXECUTOR UPLOADER" }]</pre>	
404 (Not Found)		Job not found



6.11 API function - job progress

Returns job progress.

GET /jobs/{jobID}/progress

Return

HTTP return code	Response Body	Info
200 (ok)	<pre>{ "jobID": "XXYY", "progress": [{ "task_type": "DOWNLOADER EXECUTOR UPLOADER", "task_status": "ERROR PENDING PROCESSING DONE ABORTED", "progress": "X%" } ...] }</pre>	
204 ()		Job has no progress information yet
404 (Not Found)		Job not found

6.12 API function - job status

Returns job status. This is called more often, so it has its own function.

GET /jobs/{jobID}/status

Return

HTTP return code	Response Body	Info
200 (ok)	<pre>{ "jobID": "XXYY", "status": "DONE ABORTED PENDING PROCESSING ERROR", "Error_task_type": "DOWNLOADER EXECUTOR UPLOADER if error" "Error": "<error text if available>" }</pre>	
404 (Not Found)		Job not found

6.13 API function - job log

Returns the job's current log as plain text (together with all the task's logs). The data can also be retrieved after job is already in DONE status.

GET /jobs/{jobID}/log

Return

HTTP return code	Response Body	Info
200 (ok)	PLAIN/TEXT format <complete log>	
204 (No Content)		Job has no log (yet)
404 (Not Found)		Job not found

6.14 API function - abort running job

Abort the specified running job. The job cannot be resumed again.

POST /jobs/{jobID}/abort

Return

HTTP return code	Response Body	Info
200 (ok)		Job was aborted
404 (Not Found)		Job not found
403 (Forbidden)		Job is not active

7. Technical Support

For feedback and assistance with using the MainConcept 2GO, please contact our Customer Care team at apps.support@mainconcept.com.

8. Copyright Notice

Copyright © 2021 MainConcept GmbH or its affiliates. All rights reserved.

MainConcept® and its logos are registered trademarks of MainConcept GmbH or its affiliates. This software is protected by copyright law and international treaties. Unauthorized reproduction or distribution of any portion is prohibited by law.

This manual, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment or representation by MainConcept GmbH or its affiliates. MainConcept GmbH and its affiliates assumes no responsibility or liability for any errors or inaccuracies that may appear in this book and use is at your sole risk.

Except as permitted by such license, no part of the publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of MainConcept GmbH.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

Copyright 2018 Docker, Inc. All rights reserved.

Adobe and Flash are trademarks or registered trademarks of Adobe Systems Incorporated in the USA and other countries.

DTS, the Symbol, and DTS-HD are registered trademarks of DTS, Inc.



Dolby Digital codec manufactured under license from Dolby Laboratories. Dolby and the double-D symbol are trademarks of Dolby Laboratories. Unpublished work. Copyright 2003-2014 Dolby Laboratories, Inc. and Dolby Laboratories Licensing Corporation. All rights reserved.

AAC's HE-AAC and HE-AAC v2 versions are regarded as today's most efficient general perceptual audio codecs. AAC has been standardized by ISO and IEC as part of the MPEG specifications. It is understood that it may be necessary to execute a patent license with the appropriate AAC licensing entities in order to obtain all rights necessary to create and exploit products utilizing AAC and it is recommended to contact the appropriate licensing entities, e.g. Via Licensing (www.vialicensing.com), and negotiate in good faith the adequate contracts, if any.

*Fraunhofer Institute for Integrated Circuits IIS
Attention: Audio and Multimedia Departments - MC AAC LL
Am Wolfsmantel 33
91058 Erlangen, Germany
www.iis.fraunhofer.de/amm
amm-info@iis.fraunhofer.de*

Microsoft, Microsoft Windows XP, Windows Media Player, and the Microsoft logo are registered trademarks of the Microsoft Corporation, Inc.

All other company or product names are trademarks or registered trademarks of their respective owners.

Edition: August 2021

