Report Service - 2.2

On this page:

- Report
  - Configuration
  - URI Templates
  - Error Messages
  - Example of use
  - Administration Task

Report

This Web Services allows to get a report using the Report Generator in an asynchronous way.

The REST API spawns the Report Generator processes, and the REST API is also called back by the Report Generator itself which runs as a REST API client.

See the **server** resource object and the **reportEnabled** attribute to check whether this web service is enabled

Configuration

The ## REPORT CONFIGURATION section in **application.properties** file allows to configure the interaction between the REST API and the Report Generator:

```
# The location of the Report Generator Command line. If this variable is not set then the Web Services are not
enabled.
# The path is probably something such as:
report.reportGenerator=dotnet c:\\ReportGenerator\\CastReporting.Console.Core.dll

# Set the directory to save reports. This directory should be cleaned up periodically with a "cron" script,
using files dates as a criterion.
report.directory=c:\\temp\\reports

# The URL to call back the REST API. "localhost" can be used as a domain name, as the REST API server and the
Report Generator must share the same host.
# eg. http://localhost:8080/CAST-RESTAPI/rest for a WAR archive
# eg. http://localhost:8080/rest for a ZIP archive
report.webServiceURL=http://localhost:8080/CAST-RESTAPI/rest

#Set the maximum number of concurrent Report Generator processes to limit the CPU and memory consumption
report.maxConcurrentProcesses=4

# Set a delay to determine never ending Report Generator processing.
# After this delay the Report Generator process will be killed
# Example to set a delay of 10 minutes:
report.maxDelayInSeconds=600
```

URI Templates

HTTP Action	Media Type	URI Templates	Description

GET	***/***	`{Domain}/reports?{parameters}`	Get a report as an "open office" document file Return HTTP Status - **201 Accepted:** the report generation is in progress, or is queued because the number of concurrent Report Generator processes exceeds a limit (see configuration variable). - **200 OK:** report generation has succeeded, the report file is downloaded at attachment or opening a new tab (see inline parameter) - **503 Service Unavailable:** report generation is disabled or incorrect settings - **500 Internal Server Error:** report generation has failed for this template and this application. In case of an internal error, the administrator has to: - Check the REST API log to check whether the issue comes from the REST API configuration (ex: Report Generator is not installed) - Check the Report Generator log to check whether the issue comes from the Report Generator (ex: bad template name)

Parameters

URI Parameter	Description	Values	Default value
template	A Report Generator template name	a string	N/A
snapshot-id	A snapshot ID to identify an application snapshot.	an integer	N/A
previous-snapshot-id	A previous snapshot ID if it is required by the template (Note: immediate previous snapshot ID is deduced by the report generator)	an integer	N/A

Error Messages

HTTP Status	Response	Remediation
503 Service Unavailable	`{ "code": 20, "message": { "status"; "FATAL", "cause": "The report generator is not enabled for the REST API" } }`	Install the Report Generator. Set the report.reportGenerator path.
503 Service Unavailable	`{ "code": 20, "message": { "status": "FATAL", "cause": "Cannot run Report Generator. Check report.reportGenerator setting" } }`	The report.reportGenerator property is an incorrect path. Check and fix this path.
503 Service Unavailable	`{ "code": 20, "message": { "status": "FATAL", "cause": "Cannot access reports directory. Check report.directory setting" } }`	The report.directory property is an incorrect path, or the REST API does not have access to this directory. Check this directory, ensure that the service account of the REST API is granted to read and create files in this directory.
503 Service Unavailable	`{ "code": 20, "message": { "status": "FATAL", "cause": "Report Generator error (exit code: 1) - Bad command line" } }`	The Report Generator rejects the command line arguments sent by the REST API. Install the latest version of the REST API. If the problem persists, contact CAST Support with the REST AI log file containing the effective arguments used to launch the Report Generator.
503 Service Unavailable	`{ "code": 20, "message": { "status": "FATAL", "cause": "Report Generator error (exit code: 2) - Cannot access to the REST API. Check report. webServiceURL setting" } }`	The report.webServiceURL property is not correct. Check this URL with a browser or with Curl.
500 Internal Error	`{ "code": 20, "message": { "status": "FAIL", "cause": "Report Generator error (exit code: 3) - Internal error. See the log files" } }`	There is an internal error of the Report Generator. If you have updated the templates files by yourself, check the Report Generator log files; otherwise, contact CAST Support with the REST API log file containing the effective arguments used to launch the Report Generator.
	`{ "code": 20, "message": { "status": "FAIL", "cause": "Report Generator error (exit code: XXX) - Unknown error. See the log files" } }`	There is an internal error of the Report Generator. Contact CAST Support with the REST AI log file containing the effective arguments used to launch the Report Generator.

500 Internal Error	{ "code": 20, "message": { "status": "FAIL", "cause": "Report Generator processing has exceeded 600 seconds. Check report. maxDelayInSeconds setting" }	The Report generation is too long. Check that the Report Generator is ending using the Report Generator UI or command line. If the Report Generator ends successfully, then increase the required time and update the report.maxDelayInSeconds property; otherwise contact CAST support.
500 Internal Error	{ "code": 20, "message": "Unexpected end of generation. The document was not found" }	The document was not found. Maybe the Repor Generator was unable to create the file. Check the Report Generator can access to this directory; otherwise contact CAST Support.

Example of use

```
#################
# Check whether the report generator is enabled
#################
C:\Temp>curl -u admin:cast http://localhost:8080/CAST-RESTAPI/rest/server
{"href":"server","name":"Server",... "reportEnabled":true, ...}

#################
# Request for a report -> 202 Accepted, report is not yet ready
#################
C:\Temp>curl -u admin:cast "http://localhost:8080/CAST-RESTAPI/rest/WEBGOAT/reports/?template=CISQ+-+Top+22+-
+Summary.docx&snapshot-id=1"  -O -J -v
< HTTP/1.1 202 Accepted

#################
# Request for a report -> 200, report is ready and downloaded
#################
C:\Temp>curl -u admin:cast "http://localhost:8080/CAST-RESTAPI/rest/WEBGOAT/reports/?template=CISQ+-+Top+22+-
+Summary.docx&snapshot-id=1"  -O -J -v
< HTTP/1.1 200 OK
< Content-Disposition: attachment; filename="WEBGOAT-1-CISQ - Top 22 - Summary.docx"
< Content-Type: application/vnd.openxmlformats-officedocument.wordprocessingml.document

< Content-Length: 479568
curl: Saved to filename 'WEBGOAT-1-CISQ - Top 22 - Summary.docx'
```

Administration Task

The directory of reports is never clean up.

The site administrator has to ensure that enough disk space is available.