

# API DOCUMENTATION

ELIXIR TECHNOLOGY PTE LTD

URL: <https://www.elixirtech.com>

Email: [info@elixirtech.com](mailto:info@elixirtech.com)

## Table of Contents

<b>Introduction</b>	<b>2</b>
<b>Mime Types</b>	<b>3</b>
<b>/rml-engine/api/v1/render</b>	<b>4</b>
<b>/rml-engine/api/v1/render-to-store</b>	<b>6</b>
<b>/rml-engine/api/v1/render-form</b>	<b>7</b>
<b>/rml-engine/api/v1/queue-report</b>	<b>8</b>
<b>/rml-engine/api/v1/get-report</b>	<b>9</b>
<b>/rml-engine/api/v1/cancel-report</b>	<b>10</b>
<b>/docx-engine/api/v1/render</b>	<b>12</b>
<b>/ds-engine/api/v1/generate</b>	<b>14</b>
<b>/repository/api/v1/can-write</b>	<b>16</b>
<b>/repository/api/v1/read</b>	<b>17</b>
<b>/repository/api/v1/write</b>	<b>18</b>
<b>/repository/api/v1/upload-archive</b>	<b>19</b>
<b>/repository/api/v1/download-archive</b>	<b>20</b>

# Introduction

Ambience has many HTTP endpoints and whilst most are for internal use, there are several that are published for external usage. This API document details the stable endpoints that will not change for the lifetime of the Ambience 202X releases. Where changes are necessary, they will be supplied via an alternate URL, for example `api/v2` thus retaining these `api/v1` endpoints for backwards compatibility..

Calls to HTTP endpoints can be made from any modern programming language. To remain language neutral, and supply examples which can be easily tested with no programming ability, the examples shown in this documentation use Curl (<https://curl.se/docs/tutorial.html>).

In order to call the endpoints, an API token needs to be generated in Ambience using the API Tokens module. A token is a UUID string, for example `cfdffa09-4ee2-4391-8e69-3a0f76ec8fd3`. Here we will use TTTT to represent the token. This UUID should be placed as the Authentication header of the request e.g.

```
curl -X GET -H "Authentication: elx.token TTTT"
"XXXX/rml-engine/api/v1/render"
```

Powershell can also be used to call the endpoints with `Invoke-RestMethod` e.g.

```
Invoke-RestMethod -Method Post -Uri 'XXXX/rml-engine/api/v1/render'
-Headers @{ Authentication = "elx.token TTTT"}
```

---

# Mime Types

When rendering a report, many different types of output types are available. As these are consistent across the rml-engine render endpoints, they have been listed separately here.

- application/postscript - PostScript
- application/vnd.hp-pcl - PCL
- application/x-print - Print
- application/x-html-zip - HTML Zip
- application/x-glint - Glint
- application/vnd.openxmlformats-officedocument.spreadsheetml.sheet - XLSX
- application/x-jpeg-zip - JPG Zip
- application/x-png-zip - PNG Zip
- application/x-bmp-zip - BMP Zip
- application/x-xbmp-zip - XBMP Zip
- application/x-tiff-zip - TIFF Zip
- application/x-svg-zip - SVG Zip
- application/vnd.openxmlformats-officedocument.wordprocessingml.document - DOCX
- image/png - PNG
- text/csv - CSV
- text/xml - XML
- text/html - HTML

# /rml-engine/api/v1/render

Available From: Ambience and Repertoire Release 2024.0

## Description

This renders an RML file back to the client with a specified name in a given format. The default output format is PDF but it can also output XLSX, DOCX, HTML, images, CSV or XML with configuration.

## Method

POST

## Example

```
curl -X POST -H "Content-Type: application/json" -H "Authentication: elx.token TTTT" -d '{"rml": "/ElixirSamples/Report/RML/PivotTable.rml", "Staff Name": "Alan Koh"}' -o output-rml.pdf "http://localhost:1740/rml-engine/api/v1/render"
```

## Output

- The bytes of the rendered report are returned as the body of the response, response headers contain additional information.

## Request Headers

- Content-Type
- Authentication

## JSON Content

- rml - String - the path to the RML e.g. "rml":"/ElixirSamples/Report/RML/Master-Detail Report.rml"
- mime-type - String (optional) - the mimetype of the rendered RML. Defaults to application/pdf when unspecified.

- Any other parameters required by the RML e.g. Staff Name: Alan Koh

## Result

- 200 - The rendered RML is returned to the caller

## Response Headers

- Content-Type - The type of the rendered report
- X-elxjob.PageCount - The number of pages in the rendered report
- X-elxjob.RecordCount - The number of records in the rendered report

# /rml-engine/api/v1/render-to-store

Available From: Ambience and Repertoire Release 2024.0

## Description

Renders an RML file and stores it on the server, providing a hyperlink to the rendered file. The default output format is PDF but it can also output XLSX, DOCX, HTML, images, CSV, XML with configuration.

## Method

POST

## Example

```
curl -X POST -H "Content-Type: application/json" -H "Authentication: elx.token TTTT" -d '{"rml": "/ElixirSamples/Report/RML/Master-Detail Report.rml"}' "http://localhost:1740/rml-engine/api/v1/render-to-store"
```

## Request Headers

- Content-Type
- Authentication

## JSON Content

- rml - String - The path to the RML e.g. "rml": "/ElixirSamples/Report/RML/Master-Detail Report.rml"
- mime-type - String (optional) - The mimetype of the rendered RML. Defaults to application/pdf when unspecified.
- Any other parameters required by the RML

## Result

- 200 - The rendered file is successfully stored and a string of the url of the rendered file is returned.

# /rml-engine/api/v1/render-form

Available From: Ambience and Repertoire Release 2024.0

## Description

Renders an RML to the webstore, taking the RML as a path parameter rather than as JSON content. In addition to this, the content of the request must either be application/x-www-form-urlencoded or multipart/form-data

## Method

POST

## Example

```
curl -X POST -H "Authentication: elx.token TTTT" -H "Content-Type: application/x-www-form-urlencoded" "http://localhost:1740/rml-engine/api/v1/render-form?rml=/ElixirSamples/Report/RML/Master-Detail Report.rml"
```

## Request Headers

- Authentication
- Content-Type

## Form Content

- Any other parameters required by the RML

## Parameters

- rml - The path to the RML

## Result

- 200 - The rendered file is successfully stored and a string of the url of the rendered file is returned.



# /rml-engine/api/v1/queue-report

Available From: Ambience and Repertoire Release 2024.0

## Description

Part of the asynchronous reporting API. Sends a request to the rml-engine to queue a report for rendering and returns an id for use with `get-report` to retrieve the report once it has been rendered. The report is only stored for a limited time which is configurable.

## Method

POST

## Example

```
curl -X POST -H "Content-Type: application/json" -H "Authentication: elx.token TTTT" -d '{"rml": "/ElixirSamples/Report/RML/Master-Detail Report.rml", "mimeType": "application/pdf"}' "http://localhost:1740/rml-engine/api/v1/queue-report"
```

## Request Headers

- Content-Type
- Authentication

## JSON Content

- rml - String - The path to the RML e.g. "rml":"/ElixirSamples/Report/RML/Master-Detail Report.rml"
- mimeType - String - The mimetype of the rendered RML.
- Any other parameters required by the RML

## Result

- 200 - The report is queued for rendering. The response body contains json with an id that can be used for subsequent retrieval, for example

```
{"id" : "2cf59fa2-1e34-4ff0-b165-0573196dadbb"}
```

# /rml-engine/api/v1/get-report

Available From: Ambience and Repertoire Release 2024.0

## Description

Part of the asynchronous reporting API. When sent with an id from the `queue-report` endpoint, it returns the rendered report which can be written to disk.

Reports are held in the database for a limited time that is configurable.

If the report is not found, either due to a bad id or the report has been removed from the database because the time it is held has elapsed a 404 is returned.

## Method

GET

## Example

```
curl -X GET -OJ -H "Authentication: elx.token TTTT"  
"http://localhost:1740/rml-engine/api/v1/get-report/2cf59fa2-1e34-4ff  
0-b165-0573196dadbb"
```

## Request Headers

- Authentication

## JSON Content

None

## Result

- 200 - The previously queued report is returned to the caller
- 404 - The report with the id is not found in the database

# /rml-engine/api/v1/cancel-report

Available From: Ambience and Repertoire Release 2024.0

## Description

Cancel a report that has been queued by `queue-report`. Uses the id provided by the aforementioned endpoint. Canceling a report will not remove it from the store if it has already been rendered.

## Method

POST

## Example

```
curl -X POST -H "Authentication: elx.token TTTT"  
"http://localhost:1740/rml-engine/api/v1/cancel-report/2cf59fa2-1e34-  
4ff0-b165-0573196dadbb"
```

## Request Headers

- Authentication

## JSON Content

None

## Result

- 200 - The queued rendering is canceled

## /rml-engine/api/v1/render-to-portal

Available From: Ambience and Repertoire Release 2024.0

### Description

Renders an RML to the report-portal under "Render Results" in a given format.

### Method

POST

### Example

```
curl -X POST -H "Content-Type: application/json" -H "Authentication:
elx.token TTTT" -d '{"rml": "/ElixirSamples/Report/RML/Master-Detail
Report.rml", "mimeType": "application/json"}'
"http://localhost:1740/rml-engine/api/v1/render-to-portal"
```

### Request Headers

- Content-Type
- Authentication

### JSON Content

- rml - String - The path to the RML e.g. "rml": "/ElixirSamples/Report/RML/Master-Detail Report.rml"
- mime-type - String - The mimetype of the rendered RML.
- renderDetails - String (optional) - Any other parameters required by the RML

### Result

- 200 - The report is rendered to report-portal

# /docx-engine/api/v1/render

Available From: Ambience and Repertoire Release 2024.0

## Description

Renders a DOCX file to the current directory with a specified name and a given format. The default output format is DOCX but it can output PDF with an Aspose extension and license

## Method

POST

## Example

```
curl -X POST -H "Content-Type: application/json" -H "Authentication: elx.token TTTT" -d '{"docx": "/ElixirSamples/Report/DocX/Customer Profile by Location.docx", "city": "Burnaby"}' -o output.docx "http://localhost:1740/docx-engine/api/v1/render"
```

## Output

- The name of the output file.

## Request Headers

- Content-Type
- Authentication

## JSON Content

- docx - String - the path to the DOCX e.g. "docx":"/ElixirSamples/Report/DocX/Customer Profile by Location.docx"
- mime-type - String (optional) - the mimetype of the rendered DOCX. Defaults to application/vnd.openxmlformats-officedocument.wordprocessingml.document (docx) when unspecified
- Any other parameters required by the DOCX report

## Result

- 200 - The rendered DOCX is returned to the caller as the response body

# /ds-engine/api/v1/generate

Available From: Ambience and Repertoire Release 2024.0

## Description

Generates a ds file with a specified name and a given format. The default output format is JSON and it can also output CSV and XLSX.

## Method

POST

## Example

```
curl -X POST -H "Content-Type: application/json" -H "Authentication: elx.token TTTT" -d '{"ds": "/ElixirSamples/DataSource/FruitSales.ds"}' -o fruit.json "http://localhost:1740/ds-engine/api/v1/generate"
```

## Output

- The name of the output file.

## Request Headers

- Content-Type
- Authentication

## JSON Content

- ds - String - The path to the ds file e.g. "ds":"/ElixirSamples/DataSource/FruitSales.ds"
- mime-type - String (optional) - The mimetype of the rendered ds. Defaults to application/json when unspecified.
- Any other parameters required by the datasource

## Result

- 200 - The generated ds is returned to the caller as the response body

## Response Headers

- X-elxjob.RecordCount - The number of records in the generated ds file



# /repository/api/v1/can-write

Available From: Ambience and Repertoire Release 2024.0

## Description

Checks whether the user can write to a given repository and returns a JSON containing a Boolean.

## Method

GET

## Example

```
curl -X GET -H "Authentication: elx.token TTTT"  
"http://localhost:1740/repository/api/v1/can-write?path=/ElixirSamples"
```

## Request Headers

- Authentication

## Parameters

- path - The path to the file/folder in the repository we are checking

## Result

- 200 - Json is returned with a result of the query {"canWrite" : true}

# /repository/api/v1/read

Available From: Ambience and Repertoire Release 2024.0

## Description

Returns the contents of a file and the metadata given a path

## Method

GET

## Example

```
curl -X GET -O "Fruit100.ds" -H "Authentication: elx.token TTTT"  
"http://localhost:1740/repository/api/v1/read?path=/ElixirSamples/Data  
aSource/Fruit100.ds"
```

## Output

- The name of the output file.

## Request Headers

- Authentication

## Parameters

- path - The path to the file/folder in the repository

## Result

- 200 - The file contents are returned to the caller

# /repository/api/v1/write

Available From: Ambience and Repertoire Release 2024.0

## Description

Writes a set of bytes in the request body to the repository at the specified path. This path needs to include the name of the file as it will appear in the repository.

## Method

POST

## Example

```
curl -X POST -H "Content-Type:application/octet-stream" -H  
"Authentication: elx.token TTTT" -d  
@/ElixirSamples/DataSource/Fruit100.ds  
"http://localhost:1740/repository/api/v1/write?path=/ElixirSamples/Da  
taSource/Fruit100Upload.ds"
```

## Request Headers

- Content-Type
- Authentication

## Parameters

- path - The path to the destination folder in the repository

## Result

- 200 - The file is successfully written to the repository

# /repository/api/v1/upload-archive

Available From: Ambience and Repertoire Release 2024.0

## Description

Uploads a ZIP archive to the repository. The folder the archive is uploaded to must already exist in the repository and will return a 400 if it does not.

## Method

POST

## Example

```
curl -X POST -H "Authentication: elx.token TTTT" \
  --form file=@example.zip \
  --form path="/User/admin/test-upload" \
  "http://localhost:1740/repository/api/v1/upload-archive"
```

## Request Headers

- Authentication

## JSON Content

- file - String - The ZIP file containing the archive to be uploaded
- path - String - The destination folder in the repository. This folder must already exist in the repository.

## Result

- 200 - The zip is uploaded and unzipped into the repository

# /repository/api/v1/download-archive

Available From: Ambience and Repertoire Release 2024.0

## Description

Downloads a zip file of the given repository path. If a folder is not specified as the path, nothing will be downloaded and a 404 will be returned.

## Method

POST

## Example

```
curl -X POST -O -H "Authentication: elx.token TTTT"  
"http://localhost:1740/repository/api/v1/download-archive?path=/ElixirSamples/DataSource"
```

## Output

- The name of the output archive. If unspecified, the name will be derived from the archive.

## Request Headers

- Authentication

## Parameters

- path - The path to the file/folder in the repository

## Results

- 200 - The zip is returned to the caller as the response body
- 404 - The specified folder could not be found