XLReporter REST API

Overview

The XLReporter REST API is provided to get a list of available templates in the project and to generate reports from those templates in different formats. These formats include Excel workbook, PDF file, JSON, Bitmap, Text/CSV, and XML.

This can be used to not only generate complete reports but can also be used to deliver information from reports to other sources.

Configuration

Installation and Licensing

The Rest API requires the distributed edition to be installed and must have at least 1 client license available.

The REST API is only available with a subscription license.

Security

To access the REST API, a client needs to log in. It is recommended that user security is enabled on the XLReporter project. If user security is not enabled, every user must log in using the "Guest" account.

If, after logging in, there is a period of inactivity, the client will be automatically logged out. The amount of expired time is governed by the **Idle Time** setting of the *XLRwebpool* application pool configured in IIS.

Templates

The REST API is designed to generate reports from templates that do not require any parameters in order to produce. The template should have no variables defined and be designed to work off the current time.

In the **Template Library**, a set of **Push Report** templates are provided that can be used out of the box.

REST Methods

Login

Log in with a user account and password. The account should be a XLReporter user.

	Description	
Request model	POST	
Default URL	http://localhost/xlrweb/api/login	
Body	Type	JSON (application / JSON)

	Content	{
		"username": "…",
		"password": "…"
		}

Response	Type	JSON

	Value	{
		"endpoint": "login",
		"success": true,
		"error": 0,
		"message": null
		"data": [
		"token": "xxx-xxx-xxx-xxx"
		}
		]
		}

The API returns a "token" which will be used for other methods. The token will expire due to inactivity.

Note that if *http* is used, the user credentials are transmitted in clear text.

Logout

To end a session, the client should log out.

	Description	
Request model	POST	
Default URL	http://localhost/xlrweb/api/logout	
Body	Type	JSON (application / JSON)

	Content	{
		"token": "xxx-xxx-xxx-xxx"
		}

Response	Type	JSON

	Value	{
		"endpoint": "logout",
		"success": true,
		"error": 0,
		"message": null,
		"data": null
		}

Any login that becomes inactive will be logged out automatically.

Templates

Obtain an enumeration of the templates configured in XLReporter.

	Description	
Request model	POST	
Default URL	http://localhost/xlrweb/api/templates	
Body	Type	JSON (application / JSON)
	Content	{ "token": "xxx-xxx-xxx-xxx" }
Response	Type	JSON
	Value	{ "endpoint": "templates", "success": true, "error": 0, "message": null "data": [{ "name": "Batch Cycle", "description": "…" }, { "name": "Product Routing", "description": "…" } …] }

The API returns each template with the following information:

- **name** – name of the template
- **description** – user defined description of the template

The list of templates returned include any template without parameters (e.g., a template that uses no variables) and any template configured with one or more instances. The name of these appear as *Template*|*Instance* (e.g., *Process Trend Charts*|*Temperatures*).

For more information on instances, see **REPORT\Deploy On-Demand Reports** in the **Document Library**.

Report

Produce a report using a specified template.

	Description	
Request model	POST	
Default URL	http://localhost/xlrweb/api/report	
Body	Type	JSON (application / JSON)
	Content	{ "token": "…", "template": "…", "type": "PDF", "screensize": "…" }
Response	Type	Resource or JSON
	Value if Success	Report PDF as bytes
	Value if Error	{ "endpoint": "report",

		"success": false, "error": 500, "message": "…" "data": null }

- **template** is one of the template names obtained using the method **templates**.
  To generate a report from a template with an instance, the template must be specified as *Template|Instance* (the same syntax as delivered in the **Templates** method).
  When run with an instance, all the variables are assigned with the values stored in the instance except for variables assigned using the **Date** panel.
  For those variables, the values stored for these variables are used to calculate an overall duration. That duration is then subtracted from the current time to determine the start date and start time. The end date and end time are set to the current date and time.
- **type** is the format of the response. The following are supported:
  - PDF format is a file returned in PDF format
  - Excel format is a file returned in Excel workbook (XLSX) format
  - Text format is a file returned in text/csv format
  - XML format is a file returned in XML format
  - JSON format is data returned in JSON format
  - Bitmap format is a file returned in bitmap format
- **screensize** indicates the viewport size of the device.
  Supported values are:
  - S Small display e.g., phone
  - M Medium display e.g., tablet
  - L Large display e.g., laptop or PC monitor. This is the default if not specified.

  Note, this setting only applies to **type** set to *PDF*.

The Response is based on the **HTTPStatus** returned in the header. If **Status** is *500.1* this indicates an error has occurred in generating the report and the Response body is a JSON object with details about the error.

The request type determines the response body:

- A file, the response body is that file in bytes.
- JSON, the response body is json object.

The header of the response contains:

- **Content-Length** set to the number of bytes of the file in the body.
- **Content-Type** set to the **type** in the request
  - *application/PDF*, *application/vnd.openxmlformats-officedocument.spreadsheetml.sheet, application/json* etc.
- **Content-Disposition** set to *filename=template.xxx* if the **type** is a file.

Type

If the **type** is set to *Text*, *XML*, *JSON* or *Bitmap*, the template must have a Data Management connection configured to export a range from the report to that format.

- *Text*

  This format requires a **Data Management Date Export Range to Text** connection to be configured in the template.

  Either **Format** is supported. The **Directory** and **File** settings should be left blank so the defaults can be used. **Overwrite** should be set to *Yes* so this template can be reused.
- *XML*

  This format requires a **Data Management Date Export Range to XML (Schema)** connection to be configured in the template.

The **Directory** and **File** settings should be left blank so the defaults can be used. **Mode** should be set to *Overwrite* so this template can be reused.

- *JSON*
  This format requires a **Data Management Date Export Range to JSON** connection to be configured in the template.

  The **Directory** and **File** settings should be left blank so the defaults can be used. **Mode** should be set to *Overwrite* so this template can be reused.
- *Bitmap*
  This format requires a **Data Management Date Export Range to Bitmap** or **Chart to Bitmap** connection to be configured in the template.

  For either function, the **Directory** and **File** settings should be left blank so the defaults can be used.

TemplateSnippets

Obtain an enumeration of snippets a template configured in XLReporter.

A snippet is an image file (.bmp) produced from the template using the **Data Management Data Export Range to Bitmap** or **Chart to Bitmap** function. For more information on these functions see **DESIGN\Data Management** in the **Document Library**.

	Description	
Request model	POST	
Default URL	http://localhost/xlrweb/api/templatesnippets	
Body	Type	JSON (application / JSON)

	Content	{
		“token”: “xxx-xxx-xxx-xxx”,
		“template”:”…”
		}

Response	Type	JSON

	Value	{
		“endpoint”:
		“templatesnippets”,
		“success”: true,
		“error”: 0,
		“message”: null
		“data”:
		[
		{
		“name”: “RangeSnip1”
		},
		{
		“name”: “ChartSnip1”
		}
		…
		]
		}

- **template** is one of the template names obtained using the method **templates**. If the template has an instance, that should be specified along with the template for this method. If any snippet name uses the *{Instance}* variable, that variable is normalized with the instance name specified and delivered as the name of the snippet in the **Response**.

The API returns each template snippet with the following information:

- **name** – name of the template snippet (the **File** setting in the management function)

Snippet

Obtain a snippet using a specified template. Note that this function returns the contents of the snippet file (.bmp) from the last time the report was generated. Typically, the **Report** method should be called before using this method to ensure that the snippet is up to date.

	Description	
Request model	POST	
Default URL	http://localhost/xlrweb/api/snippet	
Body	Type	JSON (application / JSON)
	Content	{ "token": "…", "template": "…", "snippetName": "…" }
Response	Type	Resource or JSON
	Value if Success	Snippet BMP as bytes
	Value if Error	{ "endpoint": "snippet", "success": false, "error": 500, "message": "…" "data": null }

- **template** is one of the template names obtained using the method **templates**. If the template has an instance, that should be specified along with the template for this method. If any snippet name uses the *{Instance}* variable, that variable is normalized with the instance name specified and delivered as the snippet in the **Response**.
- **snippet** is one of the snippet names obtained using the method templatesnippets.

The Response is based on the **HTTPStatus** returned in the header. If **Status** is *500.1* this indicates an error has occurred in obtaining the snippet and the Response body is a JSON object with details about the error.

The request type determines the response body:

- A file, the response body is that file in bytes.
- JSON, the response body is json object.

The header of the response contains:

- **Content-Length** set to the number of bytes of the file in the body.
- **Content-Type** set to the **type** in the request
  - *image/bmp*
- **Content-Disposition** set to *filename=snippet.bmp*.

Demonstration/Troubleshooting

To demonstrate and/or troubleshoot the XLReporter REST API, in the **_rest** subfolder of the **XLReporter** installation folder is the Postman JSON collection *XLRAPI.postman_collection.json*. This collection provides the syntax for the methods provided.

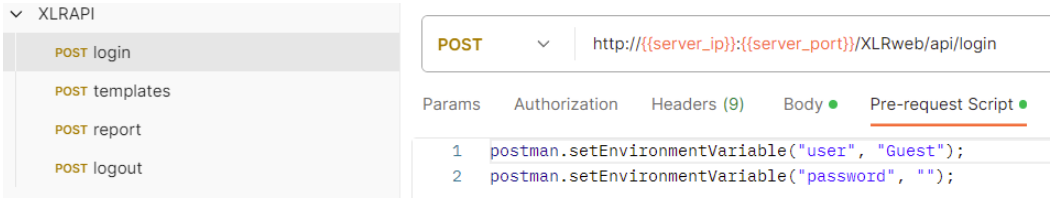
This requires the Postman API platform to be downloaded and installed. If this is being used to troubleshoot, please install Postman on the same machine that you are trying to access the XLReporter REST API from to verify connectivity.

Once installed, import the collection into Postman.

Select the **XLRAPI** collection on the left side. On the right, select the **Pre-request Script** tab.

Update the values for **server_ip** and **server_port** to match what you have set up for XLReporter.

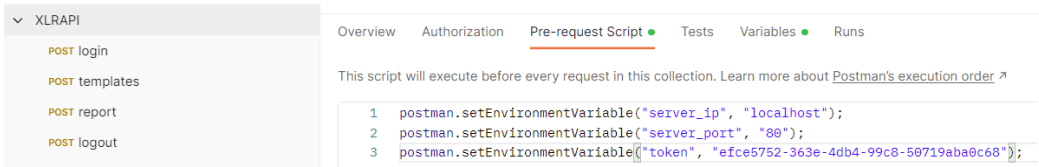
On the left, select **login** and on the right, select the **Pre-request Script** tab.



If user accounts are enabled in the project, update the values for **user** and **guest** for a valid user account.

Click the **Send** button in the upper right to send the login request.

If successful, the **Response** section will have the **token** to submit with all other requests. Highlight and copy the token value.



Return back to the **XLRAPI Pre-request Script** and paste in the **token** value.

On the left, select **templates** and on the right, click the **Submit** button. The **Response** should contain a list of templates from the project that have no parameters required. Highlight and copy the name of a template that you would like to generate a report for.

On the left, select **report** and on the right, select the **Pre-request Script** tab. For **template**, paste in the name you copied from the previous step. If you would like to, change the **type** to any supported type and **size** to any supported size. Click **Send** to generate the report.

Depending on the format and what is set up on your system, the report may appear in the response. Otherwise, you can save the response out and then open the saved file to view the report.

Once you are done demonstrating and/or troubleshooting, select **logout** on the left and click **Send** to log out of the session and free up the client license.