

Server/Client API

Server API

The key concept in REST is the idea that each resource should have a unique global identifier and that operations can be applied to the resource through a set of well-defined operations (GET,POST,PUT and DELETE). In Elixir Repertoire, this means that each repository file and folder, user, group, report, datasource, target, and even a log, all have distinct identifiers – URLs – that allow direct interaction. For example, creating a new user called **bill** can be done through the Web Interface with your browser, as described previously, but can just as easily be done by sending a PUT request to the URL */user/bill*.

Similarly, accessing a server log is as simple as sending a GET request to */log/file/server.log*. You can integrate that log into a portal if you choose, or maybe write a utility to scan the log for specific tasks that you are monitoring.

To render a report template (CustomerListing) to output PDF

- <http://localhost:8080/report/ElixirSamples/Report/CustomerListing.rml?mime-type=application/pdf>

To render a report template (CustomerListing) to Ms Excel

- <http://localhost:8080/report/ElixirSamples/Report/CustomerListing.rml?mime-type=application/vnd.ms-excel>

To render a report template (CustomerListing) to output PDF as inline (open in the window)

- <http://localhost:8080/report/ElixirSamples/Report/CustomerListing.rml?mime-type=application/pdf&elx.disposition=inline>

To render a report template (CustomerListing) to output PDF as attachment (user will be prompted to open or save the file)

- <http://localhost:8080/report/ElixirSamples/Report/CustomerListing.rml?mime-type=application/pdf&elx.disposition=attachment>

To render a datasource

- <http://localhost:8080/data/ElixirSamples/DataSource/ChartData.ds>

To render a datasource to CSV

- <http://localhost:8080/data/ElixirSamples/DataSource/FruitSales.ds?mime-type=text/csv>

Dashboard Publishing

Publishing dashboards using Repertoire Server is a URL requests to a PML file

- <http://localhost:8086/tool/dashboard/ElixirSamples/Dashboard/Corporate%20Performance%20Management/Performance%20Management%20Dashboard.pml>

Publishing dashboards with parameters

When publishing dashboards with parameter, we need to be careful of ambiguous syntax.

Example:

If a dashboard URL with parameters is directly invoked:

- <http://localhost:8086/tool/dashboard/ElixirSamples/Dashboard/Corporate%20Performance%20Management/Performance%20Management%20Dashboard.pml?para1=A¶2=B>

the server would first authenticate with a correct username and password, and return the URL called, and redirect to: <http://localhost:8086/tool/dashboard/ElixirSamples/Dashboard/Corporate%20Performance%20Management/Performance%20Management%20Dashboard.pml?para1=A>

This is incorrect as the 2nd parameter is lost due to ambiguous syntax.

One should encode

? = %3F

& = %26

so that there is no ambiguous syntax.

Example:

This can be invoked directly and the server would authenticate with a correct username and password and return the URL called with the 2nd parameter still intact:

- <http://localhost:8086/tool/dashboard/ElixirSamples/Dashboard/Corporate%20Performance%20Management/Performance%20Management%20Dashboard.pml?para1=A%26para2=B>

Client API

Elixir Repertoire server provides a set of Java standalone client APIs to connect to the server. This API allows you to trigger all the report-related or ETL functions. Access to the functions is controlled by the Server license. The report-related functions allow listing of reports deployed on the server repository, extracting dynamic parameters in the report and generating reports. The ETL functions allow you to access data generation and DataStore functions.

This client library may be deployed as part of a J2EE solution.

The detailed version of the Java API documentation is shipped with the server.

Version 7.1

- [Repertoire Client for Java 1.4 Environment](#) [Repertoire Client Support for Java 1.4 Environment]

Version 7.0.1

Repertoire Client for Java 5 and above:

The JAR files is located at *Repertoire-Server-7.0.1\RepertoireServer\clients\lib*

- Glint.jar
 - log4j.jar
 - RepertoireClient.jar
-

Known Issues

Version 7.1.0

- ERSCClient does not allow a large JSON parameter to be passed.

Download the latest RepertoireClient support for Java 1.5 and Java 1.4 from the respective Client API sections shown in this page.

Version 7.0.1

- A **renderReport()** API call followed by a **getParameter()** API call will cause the server to remember the previous parameter name. The subsequent getParameter calls will not be able to work correctly. The server remembers the parameters already passed in, and hence it does not ask for the parameters.
- `_IJobInfo.JOB_RECEIVED_` - The correct value is not returned
- `_IJobInfo.JOB_STARTED_` - The correct value is not returned

More documentation on accessing the Server's Resources using JAVA or REST API can be found in the following sections:

1. .Net [Sample on using .Net with a DLL Client Component]
2. Java [Sample on using Java Calls to access Server's Resources]
3. REST API [Sample on using REST Calls to access Server's Resources]