**Web Services**

|  |  |
| --- | --- |
| **Author:** | Manuel Rego Casasnovas |
| **Contact:** | [rego@igalia.com](mailto:rego@igalia.com) |
| **Date:** | 28/11/2012 |
| **Copyright:** | Some rights reserved. This document is distributed under the Creative Commons Attribution-ShareAlike 3.0 licence, available in <http://creativecommons.org/licenses/by-sa/3.0/>. |

Basic documentation about LibrePlan web services usage.

Table of Contents

* [1   Introduction](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#introduction)
* [2   Requirements](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#requirements)
* [3   Default credentials](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#default-credentials)
* [4   Common parameters](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#common-parameters)
* [5   XML Schemas](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#xml-schemas)
* [6   Export scripts](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#export-scripts)
* [7   Import scripts](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#import-scripts)
* [8   Available web services](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#available-web-services)
  + [8.1   Business entities](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#business-entities)
  + [8.2   Other](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#other)
  + [8.3   Bound users](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients#bound-users)

**[1   Introduction](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id1)**

Inside scripts/rest-clients/ folder of LibrePlan source code, you can find several scripts to test LibrePlan web services. There are web services for most import entities in the application. And if needed more could be easily developed.

All *integration entities* (LibrePlan entities that can be exported and/or imported) has a common attribute called code. This field would be the one used to share information between LibrePlan and other systems or other LibrePlan instances.

LibrePlan web services are REST based. But, they are just using two HTTP methods with the following meaning:

* GET: Used to extract information from LibrePlan. For each case they will return a list with all the entities. For example, resources web service will return a list with all application resources.

You can also pass code attribute (since LibrePlan 1.2) in order to return just one entity sending the code as a GET parameter.

* POST: Used to add and update information in LibrePlan. These services will receive a XML file with a list of entities. For each entity there could be 2 different cases:
  + If it already exists: Update entity with new data.
  + If it does not exist: Add the new entity.

These means that delete is not allowed for all the entities in the web services, in that way only new info could be added or updated. This is because of entities are related with others and remove operation could be dangerous. Anyway for some specific entities the delete operation has been implemented.

**[2   Requirements](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id2)**

These scripts are written in bash so you need to be running a bash terminal to use them. And they use cURL to do the HTTP requests, you can install it with the following command in Debian based distributions:

# apt-get install curl

Moreover, it is recommended to have Tidy available in your system. You can install it with the following command in Debian based distributions:

# apt-get install tidy

In openSUSE with:

# zypper install tidy

Or in Fedora with:

# yum install tidy

**[3   Default credentials](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id3)**

In order to test these scripts you will need an user with reader and writer permissions for web services of LibrePlan. Default credentials are:

* Reader permission:
  + User: wsreader
  + Password: wsreader
* Writer permission:
  + User: wswriter
  + Password: wswriter

Note

LibrePlan web services use [HTTP Basic Authentication](http://www.w3.org/Protocols/HTTP/1.0/spec.html#BasicAA)

**[4   Common parameters](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id4)**

There are already defined several parameters to be used in the scripts to define the server to execute the tests:

Demo environment (without parameters) - default

Use server where demo is launched using the following URL: http://demo.libreplan.org/libreplan/

Example:

$ ./export-resources.sh

Production environment (--prod)

Use server deployed with Tomcat in a production deployment in the following URL: http://localhost:8080/libreplan/ (the installation done by Debian package).

Example:

$ ./export-resources.sh --prod

Development environment (--dev)

Use server deployed with Jetty during development in the following URL: http://localhost:8080/libreplan-webapp/

Example:

$ ./export-resources.sh --dev

**[5   XML Schemas](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id5)**

To get XML schema use the following script:

$ get-xml-schema.sh <service-path>

Example:

$ get-xml-schema.sh resources

You can also get it with a browser going to the following URL using a user with read credentials for web services: /ws/rest/<service-path>/?\_wadl&\_type=xml

**[6   Export scripts](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id6)**

There are several scripts to just get information from LibrePlan system. They alway start with export- prefix. These scripts usually don't need any parameter (apart from the one to select the environment). They will return a XML output with the data requested.

Example:

$ ./export-resources.sh

To export just one specific resource by code you can add a new extra parameter. Example:

$ ./export-resources.sh WORKER00011

**[7   Import scripts](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id7)**

As for the previous point, in order to insert data in LibrePlan system thought web services there are several scripts with prefix import-. In this case, these scripts need a special parameter, that would be a XML file with data to be inserted in LibrePlan. There are usually files with -sample suffix as example data. Again, output for these scripts is a XML message with the possible errors trying to insert the data in the system.

Example:

$ ./import-resources.sh resources-sample.xml

**[8   Available web services](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id8)**

**[8.1   Business entities](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id9)**

For each entity there are the following methods:

* Export all:
  + HTTP method: GET
  + No parameters
  + URL: /ws/rest/<service-path>/
* Export one:
  + HTTP method: GET
  + Parameter: entity-code
  + URL: /ws/rest/<service-path>/<entity-code>/
* Import one or more:
  + HTTP method: POST
  + No parameters
  + URL: /ws/rest/<service-path>/
* Remove entity (only available for work reports and order elements):
  + HTTP method: DELETE
  + Parameter: entity-code
  + URL: /ws/rest/<service-path>/<entity-code>/
  + Special URL for work report lines: /ws/rest/workreports/line/<entity-code>/

Supported entities:

* Exception Days:
  + Service path: calendarexceptiontypes
  + DTO: org.libreplan.ws.calendarexceptiontypes.api.CalendarExceptionTypeDTO
  + Business class: org.libreplan.business.calendars.entities.CalendarExceptionType
* Calendars:
  + Service path: calendars
  + DTO: org.libreplan.ws.calendars.api.BaseCalendarDTO
  + Business class: org.libreplan.business.calendars.entities.BaseCalendar
* Cost categories:
  + Service path: costcategories
  + DTO: org.libreplan.ws.costcategories.api.CostCategoryDTO
  + Business class: org.libreplan.business.costcategories.entities.CostCategory
* Criteria:
  + Service path: criteriontypes
  + DTO: org.libreplan.ws.resources.criterion.api.CriterionTypeDTO
  + Business class: org.libreplan.business.resources.entities.CriterionType
* Labels:
  + Service path: labels
  + DTO: org.libreplan.ws.labels.api.LabelTypeDTO
  + Business class: org.libreplan.business.labels.entities.LabelType
* Materials:
  + Service path: materialcategories
  + DTO: org.libreplan.ws.materials.api.MaterialCategoryDTO
  + Business class: org.libreplan.business.materials.entities.MaterialCategory
* Projects:
  + Service path: orderelements
  + DTO: org.libreplan.ws.common.api.OrderDTO
  + Business class: org.libreplan.business.orders.entities.Order
* Resources:
  + Service path: resources
  + DTO: org.libreplan.ws.resources.api.ResourceDTO
  + Business class: org.libreplan.business.resources.entities.Resource
* Work Hours:
  + Service path: typeofworkhours
  + DTO: org.libreplan.ws.typeofworkhours.api.TypeOfWorkHoursDTO
  + Business class: org.libreplan.business.costcategories.entities.TypeOfWorkHours
* Unit Measures:
  + Service path: unittypes
  + DTO: org.libreplan.ws.typeofworkhours.api.TypeOfWorkHoursDTO
  + Business class: org.libreplan.business.materials.entities.UnitType
* Work Reports:
  + Service path: workreports
  + DTO: org.libreplan.ws.workreports.api.WorkReportDTO
  + Business class: org.libreplan.business.workreports.entities.WorkReport
* Expense Sheets:
  + Service path: expenses
  + DTO: org.libreplan.ws.expensesheets.api.ExpenseSheetListDTO
  + Business class: org.libreplan.business.expensesheet.entities.ExpenseSheet

**[8.2   Other](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id10)**

* Resource hours:
  + Methods:
    - Export all resource hours between two dates:
      * HTTP method: GET
      * Parameters: <start-date> and <end-date>
      * URL: /ws/rest/resourceshours/<start-date>/<end-date>/
    - Export all resource hours between two dates for a specified resource:
      * HTTP method: GET
      * Parameters: <start-date>, <end-date> and <resource-code>
      * URL: /ws/rest/resourceshours/<resource-code>/<start-date>/<end-date>/
  + DTO: org.libreplan.ws.resources.api.ResourceWorkedHoursListDTO

**[8.3   Bound users](https://github.com/LibrePlan/libreplan/tree/master/scripts/rest-clients" \l "id11)**

Special services intended to be used by bound resources. The user and password for the service correspond to the bound user.

There are 3 services:

* My tasks:
  + Export assigned tasks to the bound user:
    - HTTP method: GET
    - No parameters
    - URL: /ws/rest/bounduser/mytasks/
  + DTO: org.libreplan.ws.boundusers.api.TaskListDTO
* Timesheets by task:
  + Export the personal timesheets data of the bound user for a task:
    - HTTP method: GET
    - Parameters: <task-code>
    - URL: /ws/rest/bounduser/timesheets/<task-code>
  + DTO: org.libreplan.ws.boundusers.api.PersonalTimesheetEntryListDTO
* Import personal timesheets:
  + Import personal timesheets of the bound user:
    - HTTP method: POST
    - No parameters
    - URL: /ws/rest/bounduser/timesheets/
  + DTO: org.libreplan.ws.boundusers.api.PersonalTimesheetEntryListDTO