REST interface

Using built-in solution webservices

The rest interface has two versions.

- v1, XML-version (default)
- v2, JSON-version

Version 1 (XML)

This article has not yet been fully converted to Wiki format.

Please download the original article: Tempus Serva REST interface.pdf

Netbeans quick start guide

Steps to create a simple interaction

1. Add Webservice to ide (wadl import)
   - URL: [ServerName]/[ApplicationName]/rest/[SolutionSystemName].wadl
   - If import causes trouble: Download the wadl file
2. Create a new project
   1. Add REST Client to project
      - Point to newly created webservice: [SolutionSystemName]
   2. Add JAXB bindings to project (use XSD schema)
      - URL: [ServerName]/[ApplicationName]/rest/[SolutionSystemName].xsd
      - If import causes trouble: Download the xsd file

Sample code for list view (BASIC athentication)

```
//Create session
FirmabilerClient session = new FirmabilerClient();
session.setUsernamePassword("admin", "password1223");

//Set search parameters (first parameter is a dummy)
FirmabilerList result = session.getList(FirmabilerList.class, "", "TITEL=Kasper" );

//Retrieve data and print
List <FirmabilerListItem> list = result.getFirmabilerListItem();
for(int i=0; i<list.size(); i++) {
   //Handle single item
   FirmabilerListItem item = list.get(i);
   System.out.println( item.getDataID() + "\t" + item.getNUMMERPLADE() );
}
```

```
//Close connection
session.close();
```

Sample code for list view (parameter credentials)

```
//Create session
FirmabilerClient session = new FirmabilerClient();

//Login and set search parameters
FirmabilerList result = session.getList(FirmabilerList.class, "admin", "password1223", "TITEL=Kasper" );

//Retrieve data and print
List <FirmabilerListItem> list = result.getFirmabilerListItem();
for(int i=0; i<list.size(); i++) {
    //Handle single item
    FirmabilerListItem item = list.get(i);
    System.out.println( item.getDataID() + "\t" + item.getNUMMERPLADE() );
}

//Close connection
session.close();
```

Version 2 (Json)

A few policies are relevant for this: REST Policies

Policy "restActive" must be true.

Policy "restBasicAuthentication" must be true to allow Basic authentication See wikipedia

If REST calls are made from the browser via a logged-in user and Basic auth is not used, the calls will be made with the credentials of the user.

Download the swagger file from:

```
https://<server>/<application>/rest/v2/swagger.json
```

You can browse and try the api, using the built in SwaggerUI (from version 9147):

```
https://<server>/<application>/rest/v2/swagger
```

GET examples:

Single:

```
https://<server>/<application>/rest/v2/<entity>/<DataID>
```

List:

```
https://<server>/<application>/rest/v2/<entity>
```

URL structure

When logged in the following URLs are available without further authentication

Data list operations: GET, PUT, POST

```
http(s)://<server>/<application>/rest/<version>/<entity>
```

Data item operations: GET, PUT, POST, DELETE

```
http(s)://<server>/<application>/rest/<version>/<entity>/<DataID>
```

File list operations: PUT, POST

```
http(s)://<server>/<application>/rest/<version>/<entity>/<DataID>/<FieldName>/
```

File item operations: GET, PUT, POST, DELETE

```
http(s)://<server>/<application>/rest/<version>/<entity>/<DataID>/<FieldName>/<FileName>
```

Executing codeunit

Operations: GET, PUT, POST, DELETE

```
http(s)://<server>/<application>/rest/<version>/codeunit/<codeunitName>
```

Query parameters

The REST API supports the same filtering and search parameters, as the list-command: Lists.