Developer Guide: How to load master libraries in the *HELM Antibody Editor* V2.0

Marco Lanig & Sabrina Hecht
quattro research GmbH, Martinsried
29.09.2016

Table of Contents

1 General

The **HELM Antibody Editor** provides a default SQLite database to load the four needed libraries (domain, mutation,protease, autoconnector config). See “User Guide for the HELM Antibody Editor V2.0.pdf” for more information on this topic. If you are planning to use a company-wide database to store this information, this technical guide may help you to connect your own data sources to the HELM Antibody Editor.

2 Configuration File

2.1 Backend Connectors

The file “ab-application-configuration.properties” allows to configure Java classes used to load a specific library. These classes need to be located in the classpath, so that the editor is able to find them.

Example:

> *backend.connector-class.autoconnector=com.quattroresearch.restclient.AutoconnectorConfigLoader*
> *backend.connector-class.domainlib=com.quattroresearch.restclient.DomainLibraryLoader*
> *backend.connector-class.mutationlib=com.quattroresearch.restclient.MutationLibraryLoader*
> *backend.connector-class.protease-description-loader=com.quattroresearch.restclient.TaPIRRestProteaseDescriptionLoader*

The information about own loader classes needs to be added in four single lines as shown in the above example. The left part of each line contains the type of connector class to add (e.g. backend.connector-class.domainlib). The fully qualified name of the loader class in the classpath has to be entered after the equals sign.

2.2 Target URL

The editor supports three different environment instances that are configurable with their own target URL e.g. pointing to a REST service:

> *uri.backend-dev=http://url.to.dev:80*
> *uri.backend-qa=http://url.to.qa:80*
> *uri.backend-prod=http://url.to.prod:80*

Above paths are loaded according to the command line argument provided when starting the editor. When starting the editor without any argument, the production version is chosen automatically. The following command line arguments are supported:

> Development version: ‘-env dev’ or ‘–env development’
> QA/Test version: ‘-env qa’ or ‘–env test’
> Production version: ‘-env prod’ or ‘–env prd’ or ‘–env production’ or as default when no argument was given

In addition to the backend URL, the path to specific resource can be provided:

path.backend-mutation=/rest/lov/mutations
path.backend-domain-detection=/rest/lov/domains
path.backend-autoconnector-config=/rest/lov/autoconnector
path.backend-protease-descriptions=/registration/rs/lov/protease

2.3 Calling the backend to retrieve the data

The HELM Antibody Editor calls the configured backend connector via reflection with the concatenated target URL when "Use Master …." was selected in the settings.

Example:
Given the example configuration in 2.1 and 2.2, the editor creates an instance of *org.roche.antibody.services.IConfigLoaderDomainLibrary* as follows.

loader = new DomainLibraryLoader("http://url.to.prod:80/rest/lov/domains");

and fetches the data by calling *loader.load()*.

3 Writing a custom backend connector

As already mentioned before, the configured backend connectors need to implement their corresponding interface and provide a constructor which accepts a String as parameter.

3.1 Domain Library

Implements: *org.roche.antibody.services.IConfigLoaderDomainLibrary*

"Load()" should return: List of *org.roche.antibody.model.antibody.DomainLibraryValues*

3.2 Mutation library

Implements: *org.roche.antibody.services.IConfigLoaderMutationLibrary*

"Load()" should return: List of *com.quattroresearch.antibody.Mutation*

3.3 Autoconnector rules

Implements: *org.roche.antibody.services.IConfigLoaderAutoconnectorConfig*

"Load()" should return: List of *String*
(one entry per configuration line, may also contain comment lines starting with '#' or empty lines)

3.4 Protease descriptions

Implements: *org.roche.antibody.services.IProteaseDescriptionLoader*

"getAllProteaseDescriptions()" should return: List of org.roche.antibody.services.ProteaseDescription

3.5 More Help

Please check the provided classes *ConfigLoaderDomainLibrary, ConfigLoaderMutationLibrary, ConfigLoaderAutoconnectorConfig and ConfigLoaderProtease* in the package *org.roche.antibody.services* to see how the default SQLite loaders are implemented. But be careful to use a constructor with only one String parameter in your own implementation. However, you are not forced to really use this parameter to specify a target URL as long as your *load* routine returns valid values.

For information on the libraries and their respective fields, please take a look at the "User Guide for the HELM Antibody Editor V2.0.pdf".

4 Contact

Please contact lanig@quattro-research.com, hecht@quattro-research.com or schirm@quattro-research.com in case of questions and feedback.