GROBID manual

Authors: Damien Ridereau, Patrice Lopez

1. Overview

The project grobid-service is a RESTful service implementation for accessing the grobid system. grobid-service is an open source project under the Apache License 2.0. It comes as a war file for deploying on a web container e.g. tomcat. The project also contains the libraries of grobid-core, doing the extraction work.

2. Build and set up environment for local deployment

To build grobid for local deployment, you just have to go to the root of the project and run the following command:

```
mvn clean install
```

Then deploy the generated war to the server. The artifact is in:

```
grobid-service/target/grobid-service-<version>.war
```

3. Build and set up environment for remote deployment

3.1. Logs

Grobid uses Apache log4j as logging library. By default, the log are written in a file grobid.log in the current directory where the application is launched. This is of course not adapted to a deployment in production. In order to set the path and filename for logging, edit the file under grobid/grobid-core/src/resources/log4j.xml and change the following line according to your production logging policy:

```
        <param name="file" value="./grobid.log" />
```

you can indicate the wished log path, for instance for Tomcat:

```
        <param name="file" value="${catalina.base}/logs/grobid.log" />
```

Be sure that the Tomcat or JBoss has the write authorization in the indicated log path.

3.2. Parameters set up

In grobid-service-<version>.war, the file web.xml has 3 parameters to set before starting the server:

org.grobid.property: path to grobid.property

org.grobid.property.service: path to grobid_service.properties

org.grobid.home: path to grobid_home

These properties are filled by the following variables: _GROBID_PROPERTY, _GROBID_SERVICE_PROPERTY, and _GROBID_HOME so that it is possible to fill these values with a script given the environment. It is also possible to set manually these variables before building the war artefact.

3.3. Build

To build grobid for remote deployment, you have to go to the root of the project and run the following command:

```
mvn clean install -PgenericBuild
```

It will generate 2 artifacts, 1 in grobid-home, 1 in grobid-service:

```
grobid-home/target/grobid-home-<version>.zip
grobid-service/target/grobid-service-<version>.war
```

Copy these 2 artifacts to your remote server.

grobid-home-<version>.zip contains the needed native libraries, the models, lexicons and a config directory that contains 2 properties files grobid.properties and grobid_service.properties.

You have to unzip grobid-home wherever you want on your server.

```
unzip grobid-home-<version>.zip
```

4. Use of grobid-service console

Welcome page is available at http://<server instance name>/<root context name> (i.e: for local tomcat http://localhost:8080/<name of the war deploy in webapp>). From there you can access to about grobid (Fig 4.1), process some conversion from the interface "Test Rest Interface" (Fig 4.2) and access the administration parameters contained in grobid.properties and grobid_service.properties (Fig 4.3):

Fig 4.1: About

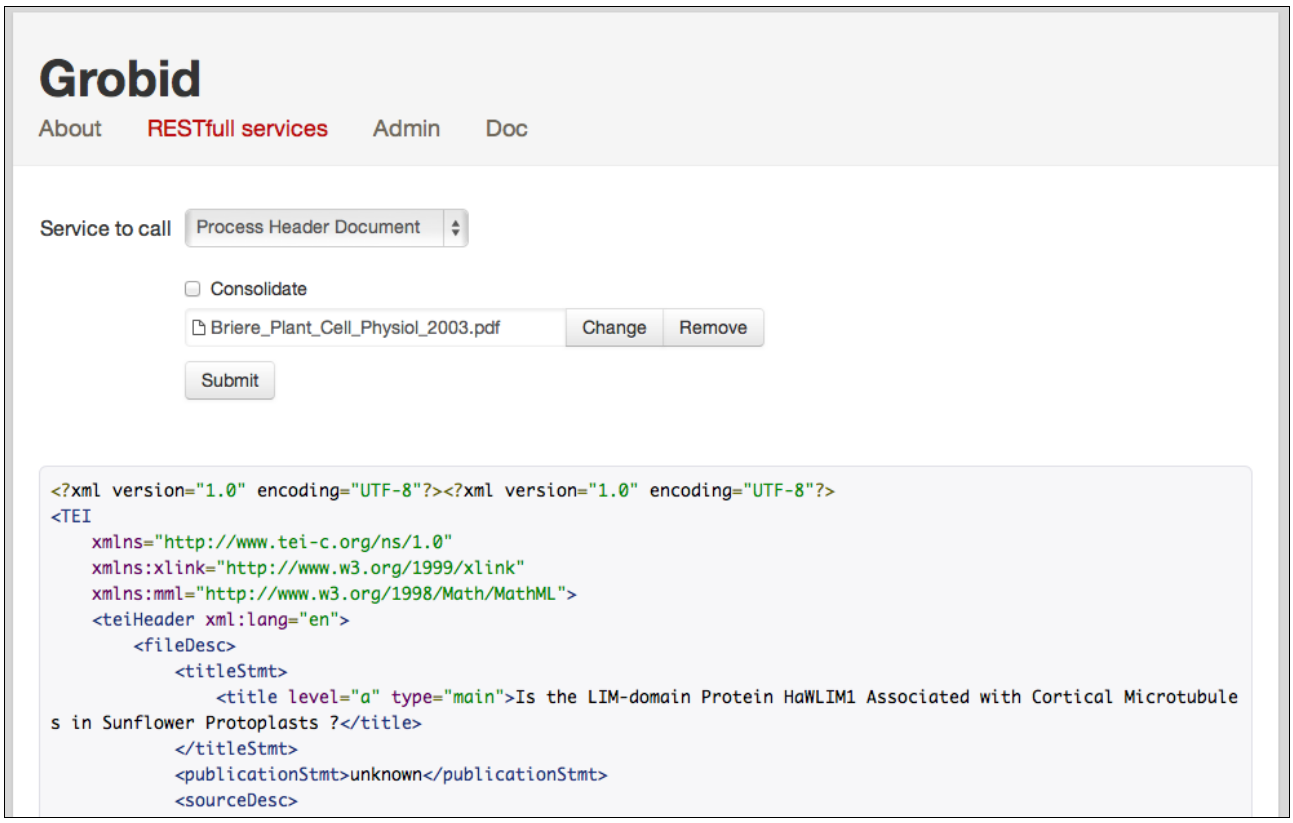

Fig 4.2: Test Rest Interface

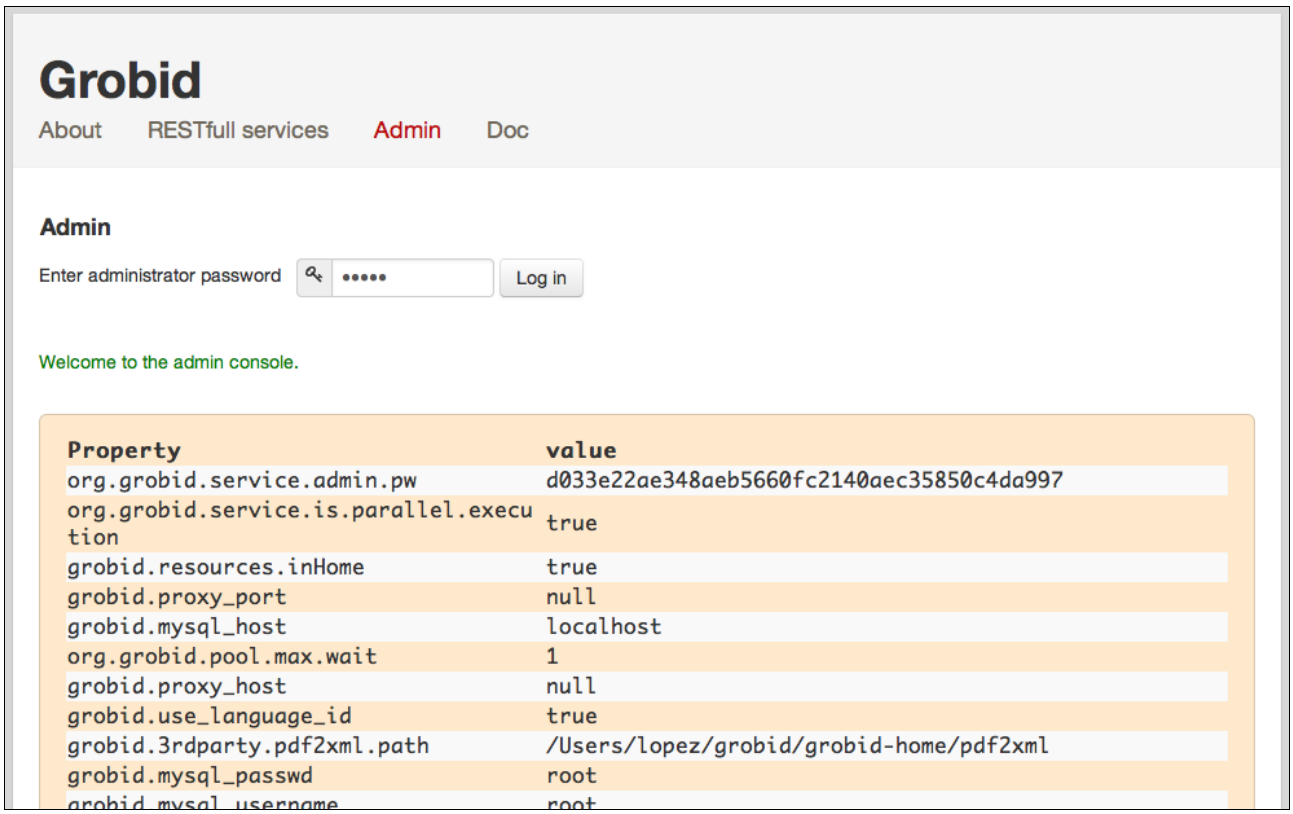

Fig 4.3: Service administration

The web page "Test Rest Interface" (Fig. 4.2) allows you to test the different REST requests quickly and easily. For technical look in the code, GrobidRestService class is the entry point for each rest service of Grobid.

5. grobid-service REST API

The table below shows the provided resources corresponding to the HTTP verbs, to use the grobid-service. All url described bellow are relative path, the root url is *http://<server instance name>/<root context>*.

Type of request	URL	Parameter name	Requesting type	MIME Type		Description
				Request input type	Response output type	
Administration	/admin	sha1	POST	application/x-www-form-urlencoded	text/html	Request to get parameters of grobid.properties and grobid_service.properties formatted in html table.
	/admin?sha1=<pwd>		GET	String		
	/sha1	sha1	POST	application/x-www-form-urlencoded	text/html	Request to get an input string hashed using sha1.
	/sha1?sha1=<input string>		GET	String		
	/allProperties	sha1	POST	application/x-www-form-urlencoded	text/xml	Request to get all properties key/value/type as xml. Sent xml follow the following schema: <properties> <property> <key>key</key> <value>value</value> <type>type</type> </property> <property>...</property> </properties>
	/allProperties?sha1=<password>		GET	String		
	/changePropertyValue	xml	POST	application/x-www-form-urlencoded	text/xml	Change the property value from the property key passed in the xml input. Xml input has to follow the following schema: <changeProperty> <password>pwd</password> <property> <key>key</key> <value>value</value> <type>type</type> </property> </changeProperty>
	/changePropertyValue?xml=<some xml>		GET	String		

General	/grobid	N/A	GET	N/A	text/html	Gives a very brief description about grobid.
Pdf to tei.xml conversion	/processHeaderDocument	input consolidate	POST, PUT	multipart/form-data	application/xml	Extract the header of the input PDF document, normalize it and convert it into a TEI format. Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).
	/processFulltextDocument	input consolidate	POST, PUT	multipart/form-data	application/xml	Convert the complete input document into tei.xml format (header, body and bibliographical section). Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).
Parse/normalize data	/processDate	date	POST, PUT	application/x-www-form-urlencoded	application/xml	Parse a raw date and return the corresponding normalized date in ISO 8601 embedded in a TEI fragment.
	/processHeaderNames	names	POST, PUT	application/x-www-form-urlencoded	application/xml	Parse a raw sequence of names from a header section and return the corresponding normalized authors in TEI format.
	/processCitationNames	names	POST, PUT	application/x-www-form-urlencoded	application/xml	Parse a raw sequence of names from a header section and return the corresponding normalized authors in TEI format..
	/processAffiliations	affiliations	POST, PUT	application/x-www-form-urlencoded	application/xml	Parse a raw sequence of affiliations and return the corresponding normalized affiliations with address in TEI format..
	/processCitations	citations consolidate	POST, PUT	application/x-www-form-urlencoded	application/xml	Parse a raw citation and return the corresponding normalized citations in TEI format. Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).

Citation extraction and normalization from patents	/processCitationPatentTEI	input consolidate	POST, PUT	multipart/form-data	application/xml	Extract and parse the patent and non patent citations in the description of a patent encoded in TEI. Results are added to the original document as TEI stand-off annotations. Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).
	/processCitationPatentST36	input consolidate	POST, PUT	multipart/form-data	application/xml	Extract and parse the patent and non patent citations in the description of a patent encoded in ST.36. Results are returned as a lits of TEI citations. Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).
	/processCitationPatentTXT	text consolidate	POST, PUT	application/x-www-form-urlencoded	application/xml	Extract and parse the patent and non patent citations in the description of a patent sent as UTF-8 text. Results are returned as a lits of TEI citations. Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).
	/processCitationPatentPDF	input consolidate	POST, PUT	multipart/form-data	application/xml	Extract and parse the patent and non patent citations in the description of a patent sent as PDF. Results are returned as a lits of TEI citations. Consolidate is a string of value 0 (no consolidation) or 1 (consolidate).