



DownloadService

OMiLAB Technical Documentation

Software version: v0.2.4

Software version (special): v0.2.3

Contents

1	Service Details	4
2	User Manual	5
2.1	Original Version	5
2.2	Special Version	6
3	DevOps Manual	8
3.1	Installation	8
3.2	Configuration	9
3.2.1	Original Version	9
3.2.2	Special Version	9
4	Contact	11
4.1	Service Developer	11

Revision History

Revision	Date	Author(s)	Description
1.0	16.06.16	S. Doppler	Created Document

1 Service Details

Prerequisites for the user

- None

Prerequisites for administrative user

- Basic HTML and CSS knowledge is beneficial

Prerequisites for the service operator

- HTML, CSS, JavaScript
- SQL
- Java
- Tomcat server administration skills
- Familiarity with the concepts of the OMiLAB Microservice Infrastructure

Dependencies

- FileManager
- MySQL

Frameworks

- Spring Boot 1.2.5.RELEASE and related

Summary

The DownloadService provides the ability to upload and download files.

2 User Manual

The main use case of the DownloadService consists in providing files for downloading. There are two versions of the DownloadService: the originally version to support downloads of ADOxx Libraries and Modelling Toolkits or more general to support downloads of single files and the special version to support multiple downloads that are listed.

2.1 Original Version

At first use you will see a button “Add new download!”. If you click on this drop-down menu you will see a list and can select your wanted type (these types are predefined by the service operator).



Figure 2.1: First view of the DownloadService

If you select a type you will get a new download item.

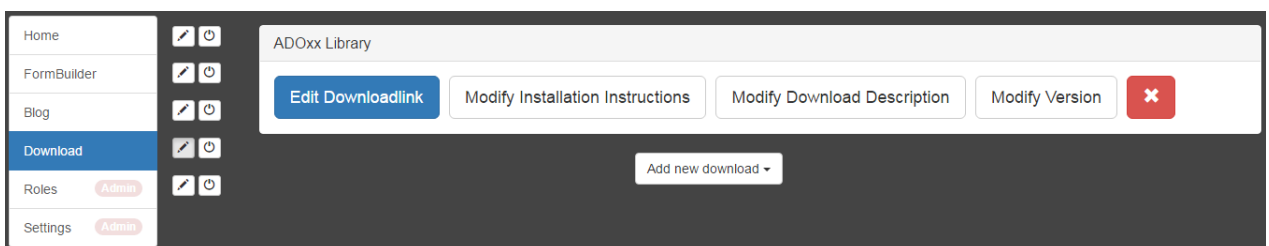


Figure 2.2: Configuration of download item

For the download item you have several options:

- Edit Downloadlink

- Modify Installation Instructions
- Modify Download Description
- Modify Version
- Delete Downloadlink

If you press “*Edit Downloadlink*” you can upload and/or select the file for the download.

In the “*Modify Installation Instructions*” menu you can write an installation instruction for the end-user. The user will see the instructions if he clicks on the “*Instructions*” button.

The download description is shown to the user above the “*Download*” and “*Instructions*” buttons.

When “*Modify Version*” is pressed you can add or edit a version to the download. The end-user then knows if there is a new version available.

If you press the red “*x*” button you can delete the downloadlink.

In the Figure 2.3 you can see the how the user sees the download items. The red frames and texts were added to get a better understanding of the configuration possibilities.

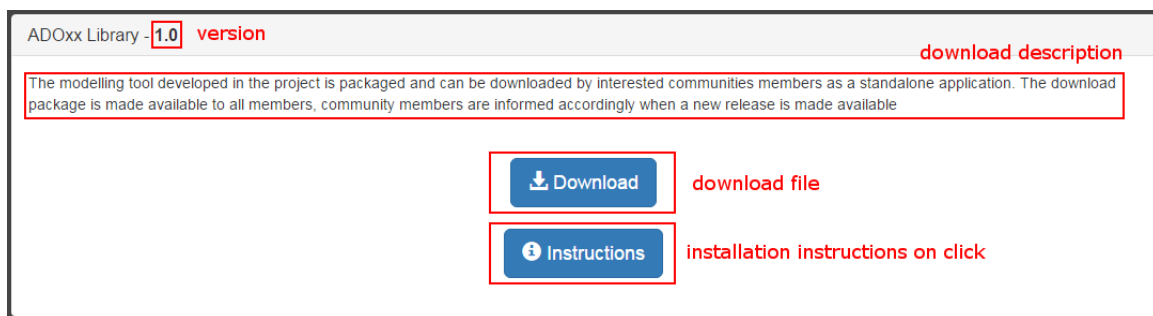


Figure 2.3: DownloadService user view

2.2 Special Version

At first use you will see a button “*Add new download!*”. If you click on this drop-down menu you will see a list and can select your wanted type (again, these types are predefined by the service operator).

If you press “*Edit Downloadlink*” you can add more than one item. You can write a short description for the file and if you have more files you have the possibility to rearrange them.

You also got the opportunity to edit the permissions for single files if you click on the lock button. You can then define which users are allowed to see the link. This setting has no effect on the target of the link (the file). If there are users who know the exact link they will be able to download the file.



Figure 2.4: First view of the DownloadService (special)

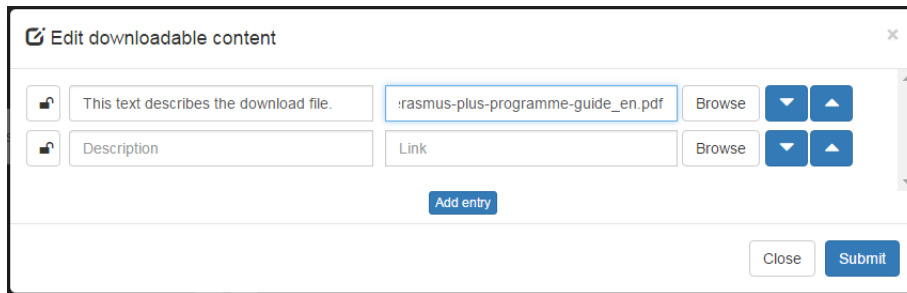


Figure 2.5: Edit Download links of DownloadService (special)

As in the original DownloadService you can configure the installation instructions, download descriptions, and the version. These are presented to the user in a slightly different way than in the original version which figure 2.6 illustrates. The version is kind of a title for the download list.



Figure 2.6: DownloadService (special) user view

3 DevOps Manual

3.1 Installation

The installation of the DownloadService (original or special version) may be performed similar to all other services. It can be built from source using maven as build system. The according target to create a deployable package is “*package*”. The profile “*local*” provides a well-documented configuration with sane default values. So the command to build the DownloadService using the command line interface can be “*mvn -Plocal package*”. This will create a deployable war archive in a new generated “*target*” folder.

Before a deployment on the application server is possible, the service has to be configured for the specific environment, which is configured in the file “*application.properties*”. If the configuration-section of the application-local.properties has not already been edited before building, the file can be found in the war archive at “*/WEB-INF/classes/application.properties*”.

```
1 app.url=http://localhost:8080/downloadlist
2
3 spring.datasource.url = jdbc:mysql://localhost:3306/
  downloadlist
4 spring.datasource.username = root
5 spring.datasource.password = cs8lorsim
6
7 omilab.activity = http://localhost:8080/logging
8 omilab.sid = SDL-VIE-01
9
10 omilab.repository = http://localhost:8080/repo/filemanager
```

What is left to do after the configuration, is the preparation of the database. When the option “*spring.jpa.hibernate.ddl-auto*” is set to “*validate*” the service expects to find valid schema at the database given and will exit, if it is not present. If the option is set to “*update*” it will update the database structure.

For the first set up, set the option to “*update*” and create a database named “*downloadlist*” (or how it is specified in the application.properties) with the following SQL command:

```
1 CREATE DATABASE downloadlist DEFAULT CHARACTER SET utf8
  DEFAULT COLLATE utf8_unicode_ci;
```

After the service is configured and the database schema is imported (or the database is created and the option set to update), the service may be deployed on the application

server. The service has been tested extensively with Apache Tomcat 7 and JRE 7, as well as MySQL 5.5.

3.2 Configuration

3.2.1 Original Version

The main concepts the DownloadService is using are an instance, a download_item, and a type.

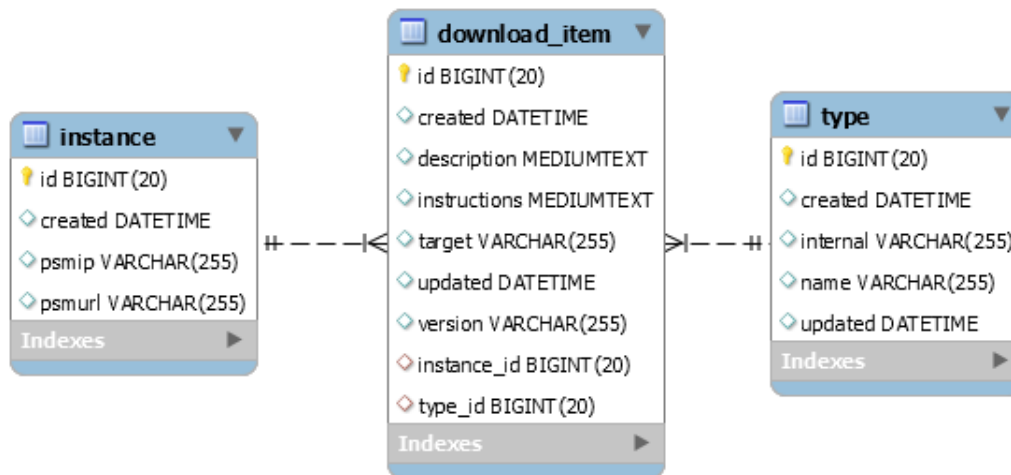


Figure 3.1: DownloadService Database scheme

In the table “*type*” there are stored an internal name and a name (that is visible to the user) besides the creation and update date and time.

A “*download_item*” has a type and has beside the creation and update date and time a download description (column “*description*”) and an installation instruction (column “*instructions*”) as well as the information where the file is stored (column “*target*”).

3.2.2 Special Version

The main concepts the DownloadService is using are an instance, a download_item, a type, and a table_element.

The table “*type*” is similar to the table in the original version. Also the “*download_item*” type is the same but with no “*target*” column.

The “*table_element*” makes the difference. Date and time are stored for the creation moment and the update moment in the columns “*created*” and “*updated*”. The name that will be visible to the user is saved in “*name*”. The access information that can be edited by a click on the lock button (see 2.5) is stored in “*roles*”. The order of the downloads is represented by the “*seq*” column. The information where the file is stored is deposited in “*target*”.

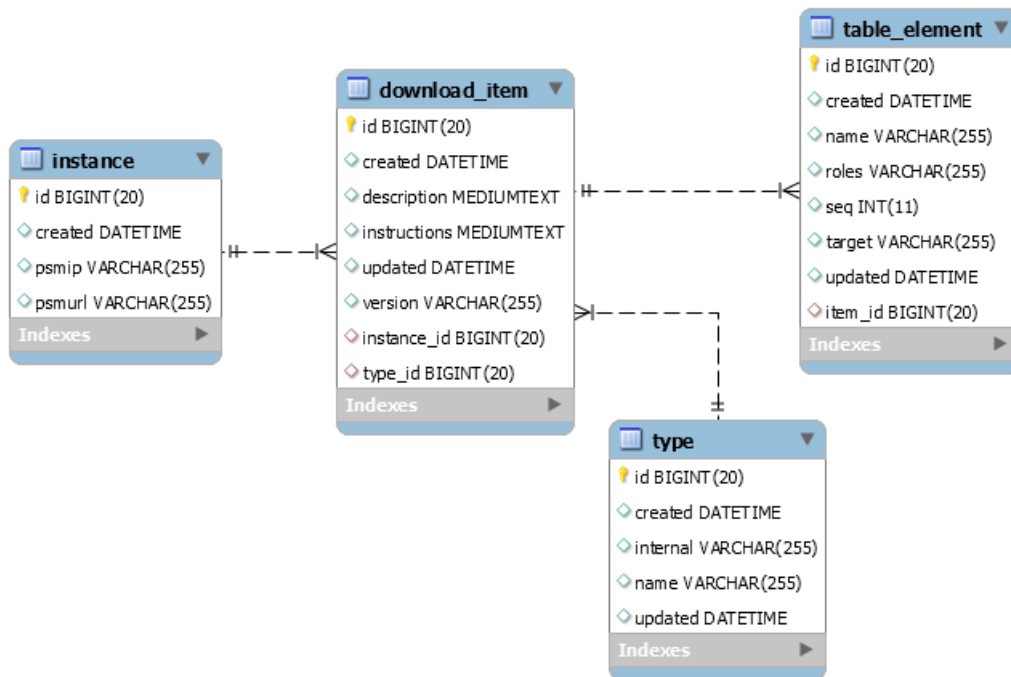


Figure 3.2: DownloadService (special) Database scheme

4 Contact

4.1 Service Developer

David Götzinger

Research Group Knowledge Engineering
Faculty of Computer Science
University of Vienna

Währinger Straße 29
1090 Vienna
AUSTRIA

Email: dgoetzing@cke.univie.ac.at