



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

EAGLE

EnhAnced Government Learning

www.fp7-eagle.eu

FP7-ICT-2013-11

Objective 8.2 Technology-enhanced learning;

Target outcome c): Holistic learning solutions for managing, reaching and engaging learners in the public administrations

Deliverable D5.6

EAGLE OER Learning Platform

WP 5 – OER Learning Platform Development
Lead Participant: Fraunhofer FOKUS

Approval Panel	Name / Partner short name	Department / Function	Date
Author	Arun Prakash Juhi Gaba, Horst Friedrich, Thomas-Frederick Gordon / Fraunhofer	System Quality Engineering (SQC), Digital Public Services (DPS) / WP5, T6.1, T6.2 Leader, Developers	01/07/2016
Author	Dietmar Glachs / SRFG	Internet of Things (IoT) / T5.3, T5.4 Leader, Developer	01/07/2016
Author	Eric Tobias / LIST	Embedded Assessment Research Group / T5.5 Leader, Developer	01/07/2016
Reviewer	Kheira Acem, Céline Thomase / LIST	Embedded Assessment Research Group / Developers	26/07/2016



This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement N°310806.



Table of Contents

1	LIST OF FIGURES	II
2	GLOSSARY	IV
3	SOFTWARE DETAILS	5
4	INSTALLATION GUIDE.....	6
4.1	EAGLE Platform and Portal.....	6
4.1.1	General Prerequisites.....	6
4.1.2	Configurations.....	6
4.1.3	Installation and Deployment Steps.....	7
4.2	EAGLE Knowledge Base.....	10
4.2.1	General Prerequisites.....	10
4.2.2	Configurations.....	11
4.2.3	Installation and Deployment Steps.....	13
4.3	EAGLE AIG Tool	15
4.3.1	General Prerequisites.....	15
4.3.2	Installation and Deployment Steps.....	16
4.4	EAGLE Process Maps and Argumentation Tool	17
5	DEVELOPER GUIDELINES	18
5.1	EAGLE Portlet development.....	18
5.2	EAGLE Knowledge Base Development	46
5.3	EAGLE AIG Development	51
5.4	EAGLE Process Maps and Argumentation Tool Development.....	52
6	REFERENCES	53
	APPENDIX 1 – RESOLVING GIT CONFLICTS WITH EGIT	54



1 List of Figures

Figure 1: EAGLE Portal Liferay Configurations.....	6
Figure 2: EAGLE Portal - Public Pages and Theme Setup.....	8
Figure 3: EAGLE Liferay Organization Management.....	8
Figure 4: EAGLE Portal - Liferay Site Membership.....	9
Figure 5: EAGLE Portal - Portlets Deployment.....	9
Figure 6: Liferay Marketplace.....	10
Figure 7 KiWi Storage Backend Configuration.....	11
Figure 8 Marmotta Core Configuration.....	12
Figure 9: Search Core Configuration.....	13
Figure 10 Eclipse Workspace.....	18
Figure 11: Eclipse Preferences.....	19
Figure 12 Preferences dialog box.....	19
Figure 13: JRE Type.....	20
Figure 14: JRE definition.....	20
Figure 15: Installed JREs.....	21
Figure 16: Help > Install New Software.....	21
Figure 17: Open Add Site Dialog.....	22
Figure 18: Add Liferay update site.....	22
Figure 19: install Liferay IDE.....	23
Figure 20: Liferay installation details.....	23
Figure 21: Accept license terms.....	24
Figure 22: Liferay IDE installation confirmation.....	24
Figure 23: Installed Plugin SDK View.....	25
Figure 24: Add Liferay Plugin SDK.....	26
Figure 25: Installed SDK.....	26
Figure 26: Server Runtime Environment View.....	27
Figure 27: Select Server.....	28
Figure 28: Liferay Tomcat Directory Location.....	28
Figure 29: Add Vaadin Update Site.....	29
Figure 30: Select Vaadin Software.....	30
Figure 31: Vaadin Review License.....	30
Figure 32: Help > Eclipse Marketplace.....	31
Figure 33: Resource Builder Editor.....	32
Figure 34: Installing Resource Builder Editor.....	32
Figure 35: Agree to License Agreement.....	33
Figure 36: Clone Git Repository.....	34
Figure 37: Source Repository Location.....	34
Figure 38: Branch Selection.....	35
Figure 39: Select Local Directory.....	35
Figure 40: Cloned Repository.....	36
Figure 41: Import Dialog Box.....	37
Figure 42: Maven Category.....	38
Figure 43: Maven Projects from Selected Directory.....	39
Figure 44: Add Working Set Name.....	40
Figure 45: Package Structure.....	41
Figure 46: Right click > New > Other.....	42
Figure 47: Select Maven Project.....	42



Figure 48: New Maven Project	43
Figure 49: Select Architype	43
Figure 50: Archetype Properties.....	44
Figure 51: Project Structure.....	45
Figure 52: Liferay Portal View	45
Figure 53: EAGLE KB Architecture Overview	46
Figure 54: Import Maven Projects	47
Figure 55: EAGLE Knowlege Base Workspace	47
Figure 56: Run Configuration	48
Figure 57: EAGLE Knowlege Backend Module creation.....	49
Figure 58: EAGLE Knowlege Backend Module Archetype selection & configuration.....	49
Figure 59: EAGLE Knowledge Backend Module Structure.....	50
Figure 60: Admin Section of the created EAGLE Module	51
Figure 61: Always start with Synchronize Workspace	54
Figure 62: If there are conflicts, it would be shown as follows	55
Figure 63: Go back to the conflicting resource in the Package View, and click Commit	56
Figure 64: Commit the changes locally	57
Figure 65: Click Merge on the project holding the conflicting resource	58
Figure 66: Click Merge	59
Figure 67: Merge Result	59
Figure 68: You will be shown conflicts like this	60
Figure 69: Open the merge tool on the conflicting file.....	61
Figure 70: Manully merge the conflicts.....	62
Figure 71:After manully merging, add the file to the index.....	62
Figure 72: Finally commit the resource	63
Figure 73: Commit and Push to remote repository	64
Figure 74: Conflict Resolution End Result.....	65



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

2 Glossary

FP7	Seventh Framework Programme
EC	European Commission
EAGLE	Enhanced Government Learning
EU	European Union
WP	Work Package
OSGi	Open Service Gateway Initiative
RAP	Remote Application Platform
EMF	Eclipse Modeling Framework
MDE	Model Driven Engineering
LOM	Learning Object Meta-data
OER	Open Educational Resource
HCM	Harvester Control Module
JDK	Java Development Kit
ETL	Extract Transform Load
SWORD	Simple Web-service Offering Repository Deposit
OAI-PMH	Open Archives Initiative - Protocol for Metadata Harvesting
FTP	File Transfer Protocol
DC	Dublin Core



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

3 Software Details

Title of the Software	EAGLE OER Learning Platform
Version	1.0
Release Date	29 July 2016
Relevant Licenses	GNU General Public License (GPL) Apache License 2.0 Eclipse Public License (EPL) v1.0
Programming Languages	Java, HTML, CSS, Velocity, GO, SPARQL
Source code	https://git.list.lu/groups/eagle
Developers	Arun Prakash, Dietmar Glachs, Juhi Gaba, Eric Tobias, Horst Friedrich, Thomas-Frederick Gordon



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

4 Installation Guide

The following instructions have been prepared to guide developers and IT professionals with installation, deployment and development the EAGLE OER Learning Platform (henceforth EAGLE Platform).

4.1 EAGLE Platform and Portal

4.1.1 General Prerequisites

The EAGLE Platform is built on Liferay which is a free and open source portal platform distributed under LGPL. The Liferay portal is built largely on Java, with the portlets conforming to both JSR 168 (Portlet Specification 1.0) and JSR 286 (Portlet Specification 2.0) standards. Experimental support for JSR 362 is available. Additionally, knowledge of JavaScript, Velocity, Freemarker scripting languages, HTML and CSS are required for developing and customizing the Liferay portal. Vaadin is used as the primary framework for developing EAGLE specific portlets. Vaadin is a Java based WebApps framework.

The following software is required as prerequisite for building, installing and running the EAGLE platform

- Java version - JDK 1.8u92
- Git distributed version control system.
- m2e Maven for Eclipse for Development
- For deployment – any machine that supports Liferay deployment (Linux 64-bit is preferable)
- Machine Requirements
 - No. of CPUS # 4
 - RAM Memory: Minimum 8GB
 - Hard disk: Minimum 20GB

4.1.2 Configurations

All configurations for the EAGLE (Liferay) portal are available in the master branch of the *portal* repository, <https://git.list.lu/eagle/portal/tree/master/eu.fp7.eagle.portal.liferay.config>.

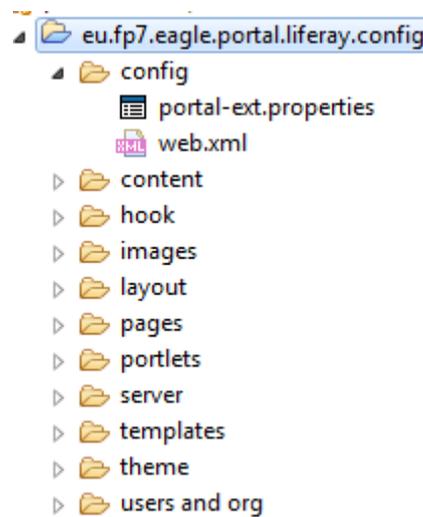


Figure 1: EAGLE Portal Liferay Configurations

The configurations are structured as shown in the



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

Figure 1, and consists of both the server side configurations, such as the *web.xml* and *portal-ext.properties* and also the individual static web content deployed on the various pages of the portal. The content are classified according the page they belong. Additionally, individual page templates and profile templates are also available.

4.1.3 Installation and Deployment Steps

Download & Extract

Download the copy of the Liferay Portal Tomcat bundle (version 6.2.5 GA6) from: <https://sourceforge.net/projects/lportal/files/Liferay%20Portal/6.2.5%20GA6/>. You may also choose other server bundles of your choice. Liferay v6.2.5 GA6 also come with tomcat, jetty, jboss and glassfish server bundles. Extract the server to the location of your choice.

Configure

Copy the *portal-ext.properties* and *web.xml* to *./liferay-portal-6.2-ce-ga6/* and *./liferay-portal-6.2-ce-ga6/tomcat-7.0.62/webapps/ROOT/WEB-INF* respectively.

Liferay comes built in with the default HSQL database. The Liferay documentation explains that this data base is not for production use but only for demo purposes. The steps required for the configuration of the Liferay database is provided here: <https://web.liferay.com/de/community/wiki/-/wiki/Main/Database+Configuration>

Liferay supports the use of SMTP server in order to send emails. The steps required to step up SMTP server to work with Liferay is provided here: <https://web.liferay.com/de/community/wiki/-/wiki/Main/SMTP+Setup>

Liferay also supports the use of SSL for securing the data communication between the client and server. The steps required to configure SSL for your EAGLE portal are provided here:

<https://web.liferay.com/de/community/wiki/-/wiki/Main/How+to+configure+https+feature>

and

<https://web.liferay.com/de/web/meera.success/blog/-/blogs/liferay-portal-ssl-configuration>

Deployment

1. Copy the hook from */eu.fp7.eagle.portal.liferay.config/hook/* to the to *./liferay-portal-6.2-ce-ga6/deploy* folder
2. Copy the theme and layout from */eu.fp7.eagle.portal.liferay.config/theme/* and */eu.fp7.eagle.portal.liferay.config/layout/* to *./liferay-portal-6.2-ce-ga6/deploy* folder.
3. Copy the latest version of the Profile page template (Profile-<timestamp>.lar) from *eu.fp7.eagle.portal.liferay.config/pages/* to the *./liferay-portal-6.2-ce-ga6/deploy* folder.
4. Start the Liferay server and login with the default admin credentials.
5. Under the Site Administration page, import the latest version of the EAGLE Public Pages (Eagle_Public_Pages-<timestamp>.lar) from *eu.fp7.eagle.portal.liferay.config/pages/*.
6. Apply the EAGLE theme.

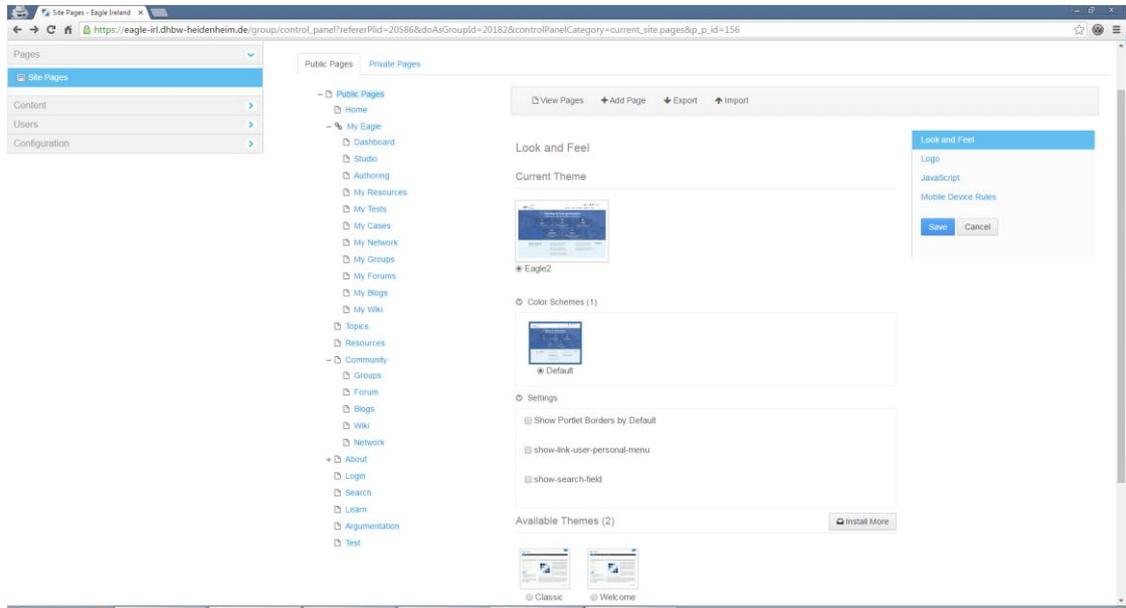


FIGURE 2: EAGLE PORTAL - PUBLIC PAGES AND THEME SETUP

- From the Control Panel, add the organization structure. More information regarding Liferay's organization management can be found here: <https://web.liferay.com/de/community/wiki/wiki/Main/Working+with+Organizations+and+Locations>

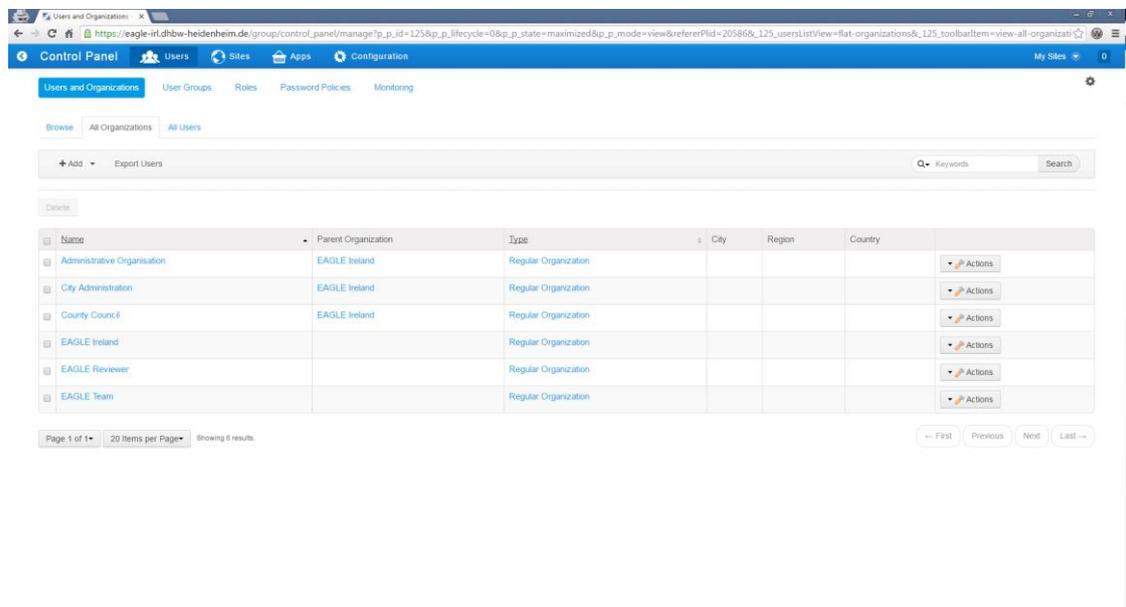


FIGURE 3: EAGLE LIFERAY ORGANIZATION MANAGEMENT

- Then go back to Site Administration > Users > Site Memberships and add all the organizations that you have created in previous step to the Eagle portal site. It is enough to add the root / parent organizations. All child organizations and users in each organization are then automatically added to the site.

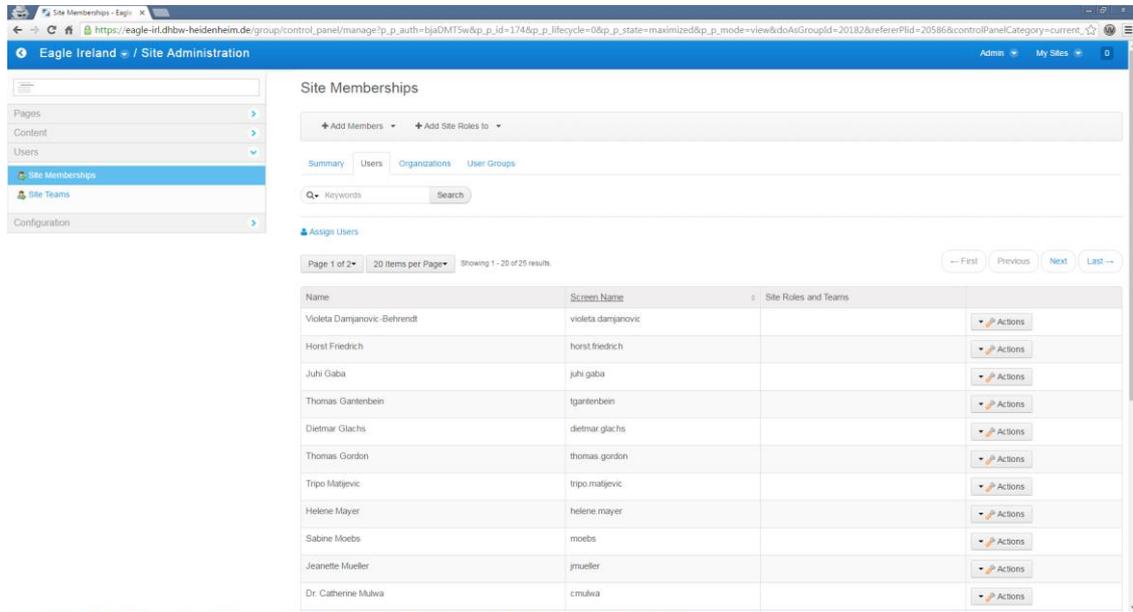


FIGURE 4: EAGLE PORTAL - LIFERAY SITE MEMBERSHIP

- Now deploy the portlets from `/eu.fp7.eagle.portal.liferay.config/portlets/` to `../liferay-portal-6.2-ce-ga6/deploy` folder. The portlets are automatically deployed and are available to use from the portlet addition page. All EAGLE specific portlets are classified under the EAGLE applications category.

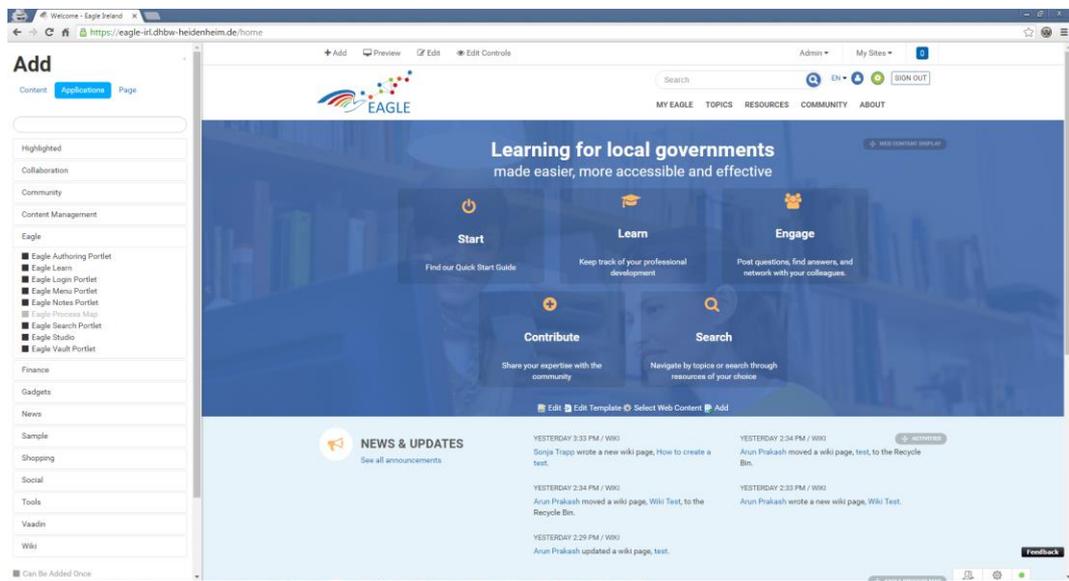


FIGURE 5: EAGLE PORTAL - PORTLETS DEPLOYMENT

- Additional new third-party portlets can be added to your EAGLE portal through the Liferay Marketplace.

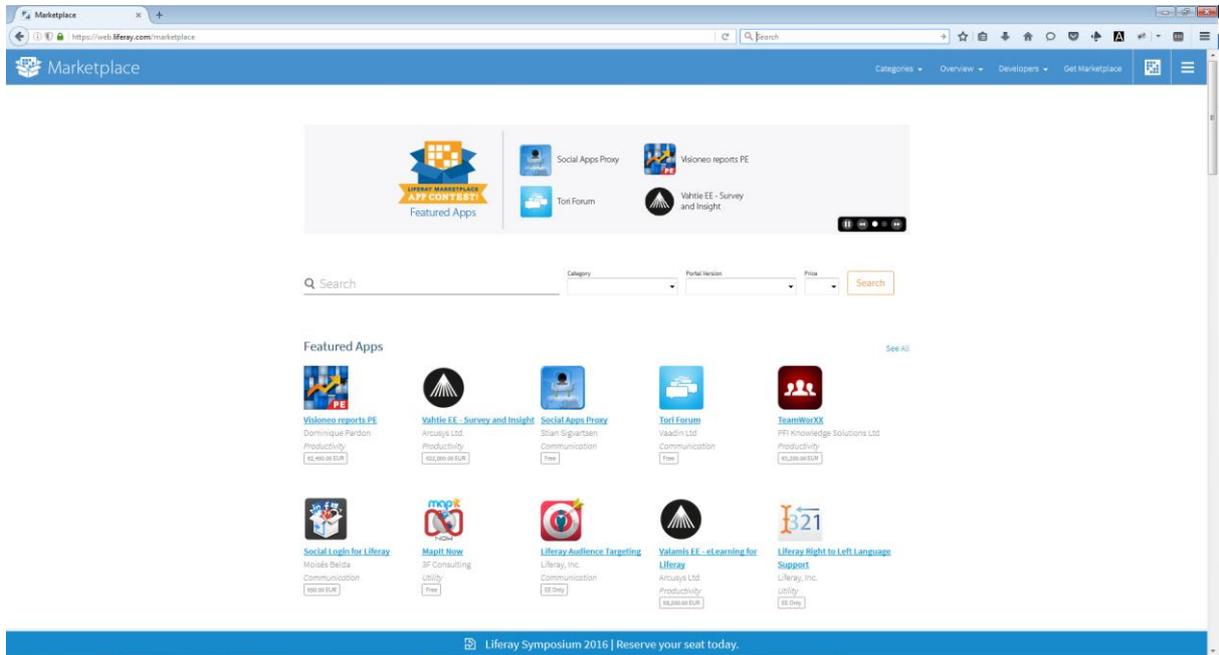


FIGURE 6: LIFERAY MARKETPLACE

11. Specific workflows or other portal and sever configurations such as email templates, etc can be managed under: Control Panel > Configuration
12. Your portal is now ready for testing and production use.

4.2 EAGLE Knowledge Base

The EAGLE Knowledge Base (short EAGLE KB) represents the backbone of the EAGLE Platform. It manages the metadata of the information collected in the EAGLE Platform. The EAGLE KB builds on and extends the Linked Data Plattform¹ Apache Marmotta².

The EAGLE KB benefits from the built-in services from Marmotta but also provides tailored services for managing OER related metadata within the EAGLE Platform.

4.2.1 General Prerequisites

The following software is required as prerequisite for building, installing and running the EAGLE Knowledge Base

- Java version - JDK 1.8u92
- Git distributed version control system.
- m2e Maven for Eclipse for Development
- For deployment – any machine that supports deployment of Java Enterprise Applications
- Apache Tomcat (version 7.x or above) as the runtime system
- PostgreSQL as the relational database backend system for Apache Marmotta

¹ <https://www.w3.org/TR/ldp/>

² <http://marmotta.apache.org/>

- Machine Requirements
 - No. of CPUS # 4
 - RAM Memory: Minimum 8GB
 - Hard disk: Minimum 20GB

4.2.2 Configurations

Apache Marmotta provides an administration page where all the deployed modules are shown. A basic installation of Apache Marmotta shows the Core Services, Storage Backend, Users & Security which comprise the basic runtime modules of Marmotta. The most important configuration setting is the used Marmotta Backend³. While the basic setting must be configured during deployment of the EAGLE KB, the database of choice may be configured after deployment by using the administrative interface. The EAGLE KB uses the default triple store KiWi⁴ that runs on top of relational databases (currently H2, PostgreSQL or MySQL). Figure 7 outlines the configuration page for the used database.

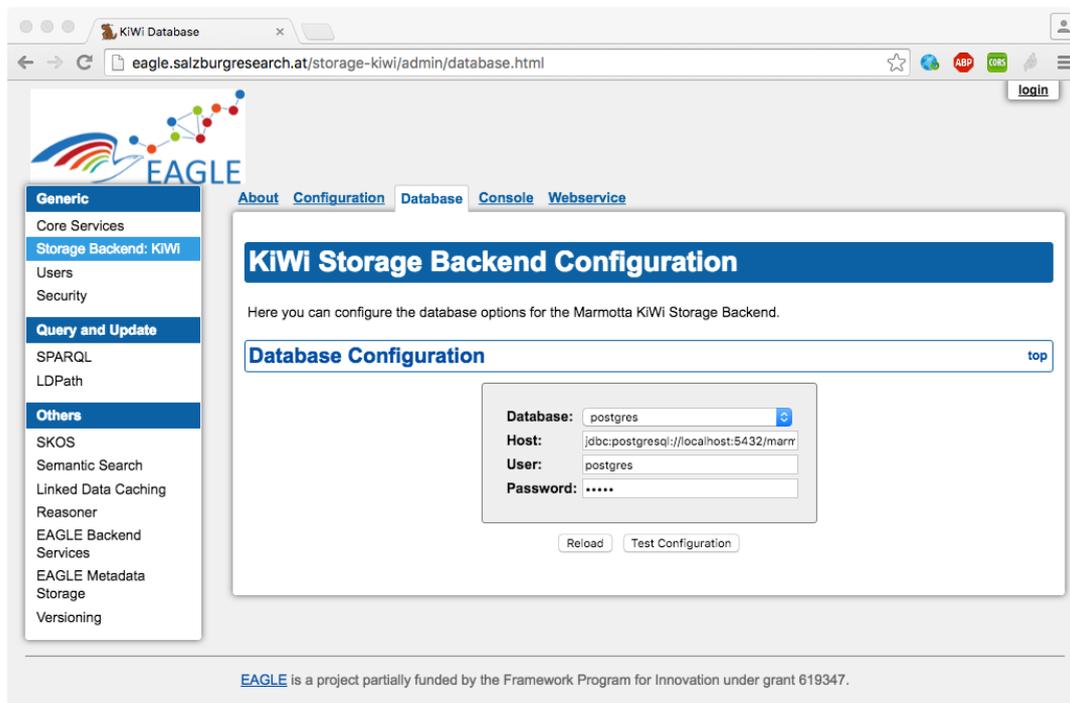


FIGURE 7 KIWI STORAGE BACKEND CONFIGURATION

Each of the Apache Marmotta modules provides its own configuration page. However, the Apache Marmotta Core Configuration Page (see Figure 8) allows managing of all configuration values at a single page.

³ <http://marmotta.apache.org/platform/backends.html>

⁴ <http://marmotta.apache.org/kiwi/>

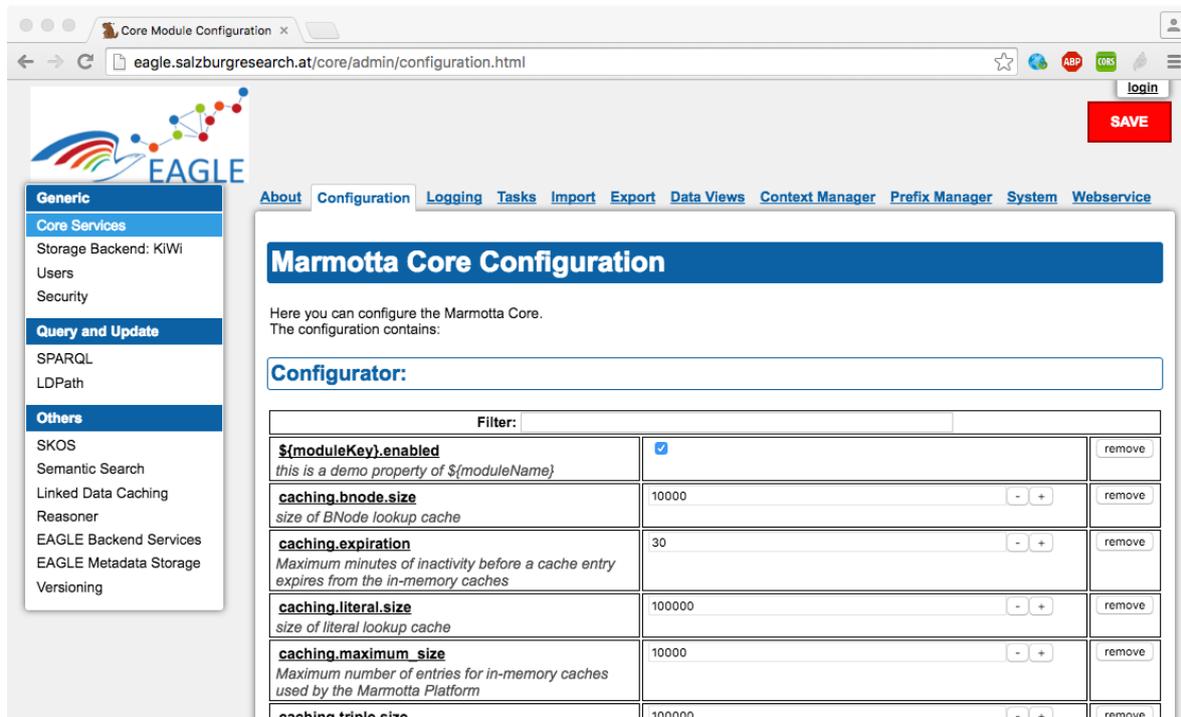


FIGURE 8 MARMOTTA CORE CONFIGURATION

Important values with respect to the EAGLE KB are

- **marmotta.home:** This configuration value points to the home directory of the EAGLE KB. It finally contains the configuration files, the search indexes and – unless changed – the H2 database files.
- **kiwi.context:** This setting denotes the prefix for the URI generation when new meta data is stored. This value must be set to <http://knowledge.eagle-learning.eu/>
- **kiwi.host:** This value denotes the internet address of the host running the EAGLE KB. For the production environment of the EAGLE platform, his value must be set to <http://eagle.salzburgresearch.at/>

Search Configuration

The EAGLE KB maintains the metadata in a graph-oriented database. The retrieval of data however is executed by means of semantic search facilities that must be configured for each running instance of the EAGLE KB. A search program constitutes on one hand the configuration of the index itself and on the other hand the access methods to retrieve the required information when filling the search index with data. Figure 9 outlines the search program for the OER search.

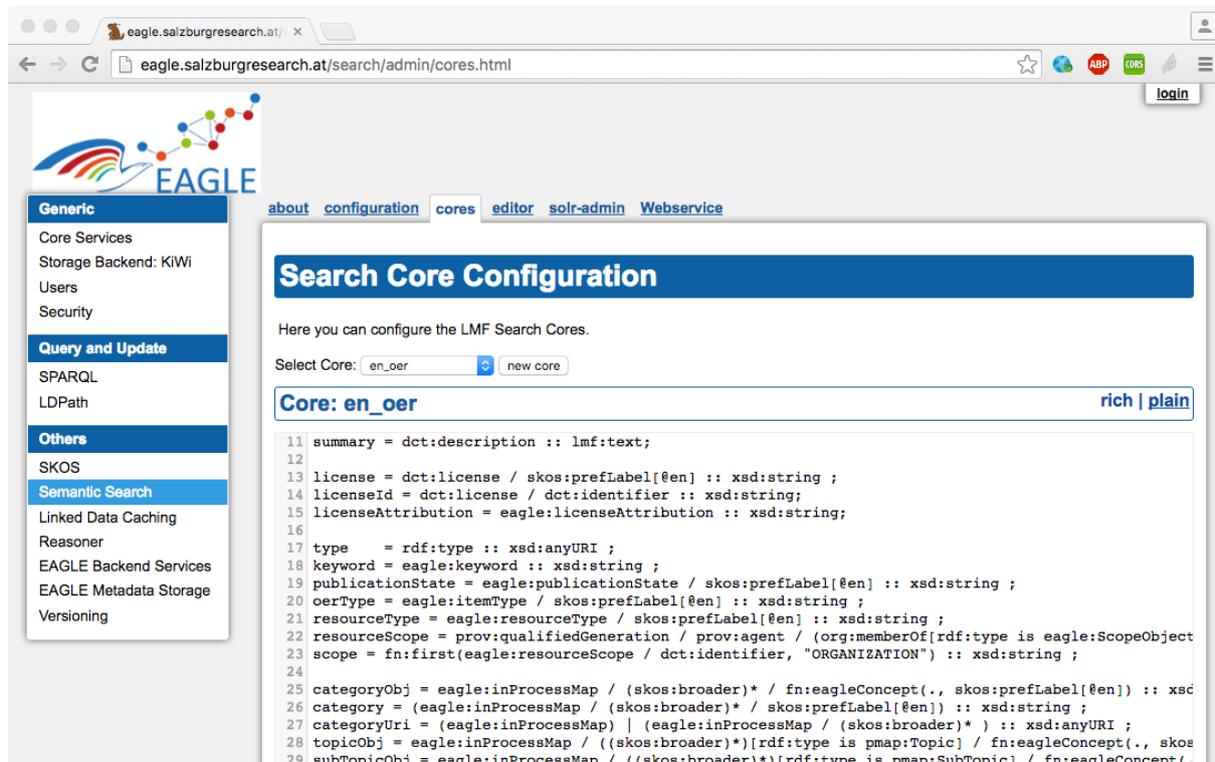


FIGURE 9: SEARCH CORE CONFIGURATION

Any search program is executed whenever metadata is created, stored or deleted. Using the changed resource as a starting point and traversing the semantic relationships stated in the search program identifies the information relevant when searching the changed resource. During the search program execution locally stored resources are processed the same way as remote resources out of a publicly available catalogue.

The distinct search programs are part of D5.5.

4.2.3 Installation and Deployment Steps

Deploy binary distribution

The EAGLE KB builds on Apache Marmotta which in turn requires Apache Tomcat as its runtime environment. Deployment of the EAGLE KB is therefore the provision of the binary distribution of the EAGLE KB, e.g. the assembled WAR (Web Archive) file, to the hosting Tomcat installation by copying the file to the *webapp*-folder of Tomcat. Tomcat recognizes the provision of the WAR file and will unpack and activate the EAGLE KB.

Upon first start of EAGLE KB, the *marmotta.home* configuration value is verified and the corresponding directory is created unless already present.

Build & Deploy EAGLE from source

In order to build the EAGLE KB from source, the prerequisites outlined in section 4.1.1 are required. The following steps will help to assemble the EAGLE KB ready for deployment on Apache Tomcat.



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

Step 1: Checkout Apache Marmotta

Apache Marmotta is an open source project. The source-code is available from <http://marmotta.apache.org> or from it's mirror site <https://github.com/apache/marmotta>.

For obtaining the source code issue the command

```
git clone https://github.com/apache/marmotta.git
```

or download the source code as a ZIP file. Once the source code is downloaded and unpacked, Apache Marmotta must be installed into the local Maven Repository by running the command

```
mvn clean install -DskipTests=true
```

Step 2: Check-Out/Download required dependencies

EAGLE KB uses Maven as it's build tool. Maven will resolve all required packages which are not part of the source-code distribution and download them. However, some of the dependencies of the EAGLE KB are neither part of the source-code distribution nor part of the

The EAGLE KB uses other open source projects that are related to Apache Marmotta but not directly part of the source code distribution of Apache Marmotta. These components have been developed at Salzburg Research

1. [skosjs: A Java-Script based editor for SKOS Thesauri](#)
2. [marmotta-search: The search component for using Apache SOLR](#)

Step 3: Check-Out EAGLE KB Source

The source code of the EAGLE KB is available at <https://git.list.lu/eagle/knowledge-space>. Issue the command

```
git clone https://git.list.lu/eagle/knowledge-space.git
```

This will clone the repository and create a copy on the local machine in the current directory. The subdirectories of the knowledge-space are

- **eagle-marmotta-backend**: The Server-Side component including the required bundles (Apache Marmotta Modules). This folder again contains a subdirectory named „eagle“ which contains the eagle service itself.
- **eu.fp7.eagle.knowledge**: This holds the knowledge model as well as some functions used by the „old“ Knowledge-Adapter in Liferay
- **third-party**: This holds the required packages which are not part of the source-code distribution of Apache Marmotta

Step 4: Build Dependencies - SKOSJS

Go to the `third-party/skosjs` directory and issue the command

```
mvn package
```



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

Step 5: Build Dependencies – marmotta-search

Go to the `third-party/skosjs` directory and issue the command

```
mvn clean install -DskipTests=true
```

Step 5: Build Marmotta 3.4.0 – Snapshot

Go to the `eagle-marmotta-backend/eagle` directory and issue the command

```
mvn clean install -DskipTests=true
```

This will compile all the distinct components of the EAGLE KB create a sub-directory named `target` in the directory and finally place the assembled distribution file (`eagle.war`) in the `target` directory.

Step7: Running EAGLE KB

Go to `./knowledge-space/eagle-marmotta-backend/eagle` and run

```
mvn tomcat7:run
```

This will download the Apache Tomcat runtime and finally deploy the `eagle.war`. Note: This is the preferred mode during development. For production purposes, deployment of the binary distribution in an existing Apache Tomcat environment is recommended.

4.3 EAGLE AIG Tool

The AIG tool is composed of three components. The first is TAO which functions as item bank, test authoring tool, and test taking tool. TAO also manages results. However, the functionalities needed for EAGLE are not supported out of the box. TAO is built modularly which allowed us to extend it with custom modules to implement the functionality needed in EAGLE. The second component is a REST service endpoint, SIREN, that exposes the third component, the AIG core libraries.

4.3.1 General Prerequisites

4.3.1.1 TAO

TAO⁵ is an open source e-Testing platform developed by Open Assessment Technologies S.A.⁶. The requirements listed below are for the recommended respectively minimum versions required to run TAO (3.0) as used in the EAGLE project:

- PHP 5.6,
- Apache 2.4.7,
- MySQL 5.5.

On the client-side, TAO can be run in any modern web browser running JavaScript. Mobile devices are supported as well but the quality of the user experience depends on screen size. This in turn also holds

⁵ https://en.wikipedia.org/wiki/TAO_%28e-Testing_platform%29

⁶ <http://www.taotesting.com>



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

true for displaying TAO in an iFrame which sometimes sees clipping issues. For specific client-side requirements we recommend to check the OAT requirements page⁷.

4.3.1.1 AIG services and libraries

SIREN is packaged as a web application archive (WAR). You will need a Java Application Server to run SIREN. You can install a new webserver or reuse an existing one. We have actively tested the application on Apache Tomcat 8. The WAR itself requires Java 8.

SIREN built as a WAR contains all dependencies and does not feature any other requirement. However, users can tweak some parameters by altering some of the properties set in a properties file. The file contains documentation but we discourage that users modify these without a good grasp of the consequences.

4.3.2 Installation and Deployment Steps

4.3.2.1 TAO and custom modules

The team at OAT has compiled an installation guide for TAO on Ubuntu which we would like to recommend. It details the installation steps and specificities for Apache, PHP, and MySQL as well as their configuration. For this project, we have used Ubuntu 14.04.2 LTS (64 bit). Visit <http://forge.taotesting.com/projects/tao/wiki/InstallUbuntuApacheMySQL> for a complete guide. We also recommend that you always install TAO with the English (en-US) default language. This will assure that, should a translation be missing, the default the localization module falls back to exists (as English is supported by default).

The provided custom TAO package for EAGLE contains all extensions to existing modules and newly developed modules. To install these modules, it is sufficient to install the modules from the TAO Extension manager once TAO has installed successfully. For the more technical users, extensions can also be installed by an installation script. To read more we recommend http://forge.taotesting.com/projects/tao/wiki/How_to_manage_your_Extensions.

As some of the custom modules need to address SIREN, please be aware that you need to provide the modules with the location of SIREN. The *taoAigFacade* folder which contains the Façade module code contains a *constants.php* file located under *taoAigFacade/includes*. The properties *TAO_AIG_FACADE_SIREN_HOST* and *PORT* respectively will need to be modified with the correct address and port number to be able to address the deployed SIREN endpoint.

4.3.2.2 AIG services and libraries

To install the AIG services it suffices to compile SIREN as a WAR (the compilation requires the inclusion of its AIG dependency), respectively, retrieve an already compiled WAR and deploy it on your Java Application Server. Once deployed, SIREN will automatically start to listen for incoming traffic and will handle it accordingly to its specification. Should any modification be made to the properties file that resides within the WAR (you can open it as you can any other archive like ZIP or RAR), you will need to redeploy SIREN. In case the modification has been made directly on the server, the application server will require to be restarted. No templates are deployed by default.

⁷ <http://www.taotesting.com/get-tao/system-requirements>



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

4.4 EAGLE Process Maps and Argumentation Tool

The installation and developer guide for the EAGLE Process Maps and Argumentation Tool are documented as part of deliverables D6.9 and D6.10 respectively.

5 Developer Guidelines

The following developer guidelines have been prepared for developing EAGLE Portlets for Liferay 6.2.5 CE GA6+ with Vaadin 7.5.0 using Eclipse Luna IDE.

5.1 EAGLE Portlet development

Step 1: Installing appropriate Java on your system.

Install JDK 1.8 update 92 from:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Step 2: Download and install the Eclipse IDE

Download the Eclipse Modelling Luna Package from the Eclipse download site.

<http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/lunars2>

Step 3: Create a new workspace to the directory of your choice.

Once the workspace is successfully created, you should get a screen similar to Figure 10.

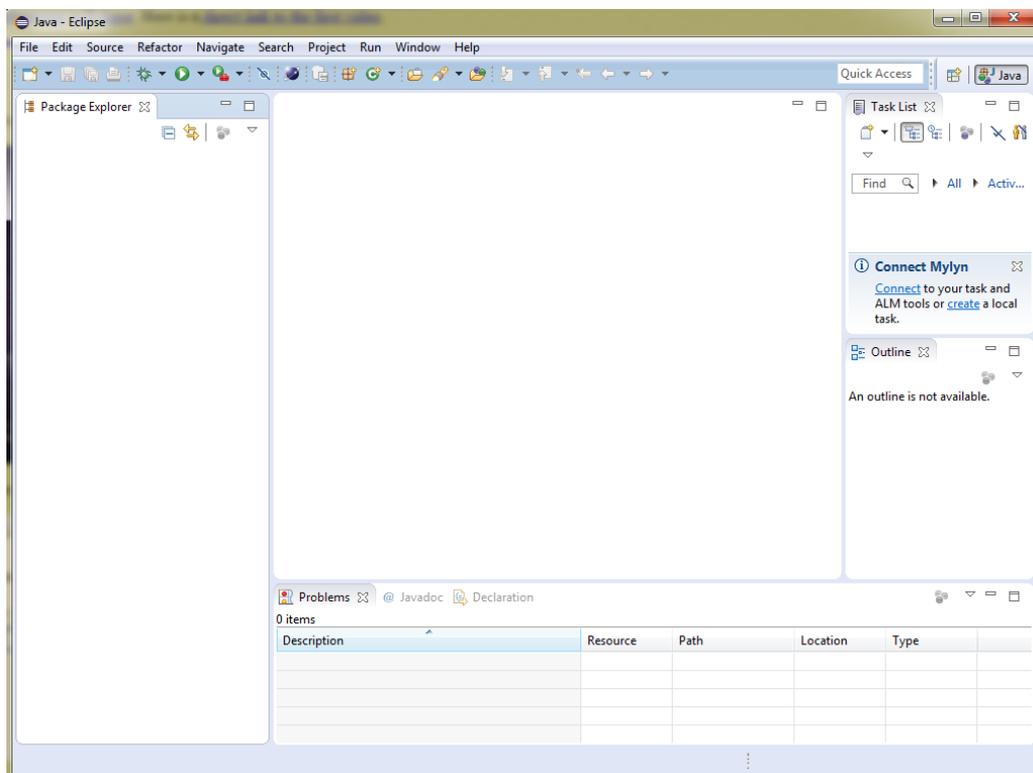


FIGURE 10 ECLIPSE WORKSPACE

Step 4: Configure Java in your eclipse workspace.

- In the Eclipse menu go to: Window->Preferences

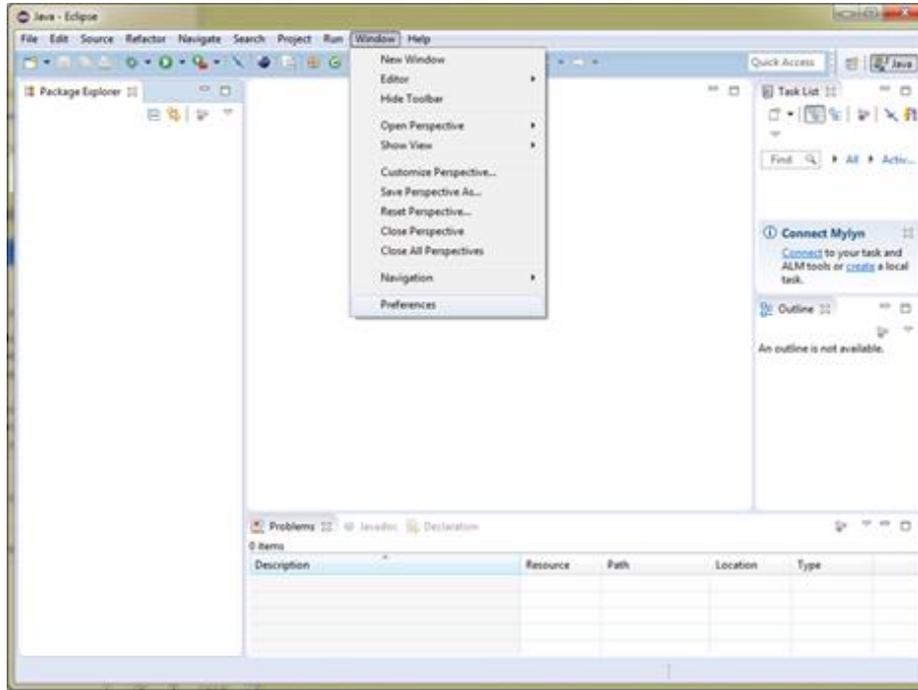


FIGURE 11: ECLIPSE PREFERENCES

- Select Java category and expand it.

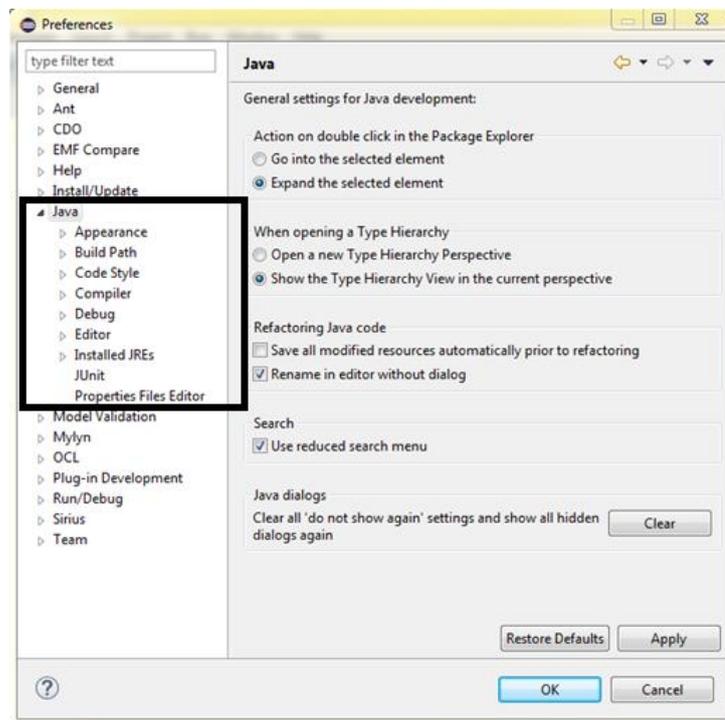


FIGURE 12 PREFERENCES DIALOG BOX

- Select the Installed JREs option from the expanded view under java category. Click on the Add button. Select Standard VM as the type of JRE n go next.

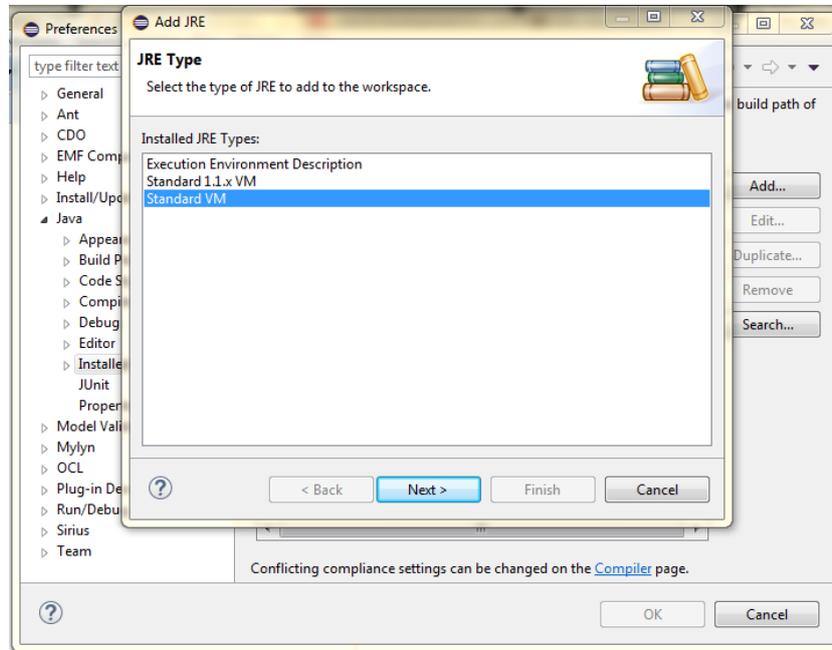


FIGURE 13: JRE TYPE

- In the next window, Click on the Directory button and navigate to: Computer > Windows C: > Program Files > Java > jdk1.8.0_92, then click Ok

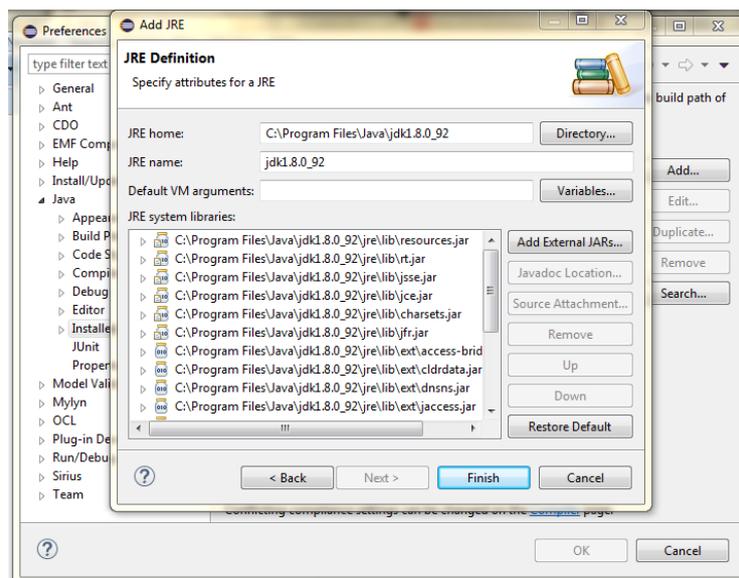


FIGURE 14: JRE DEFINITION

- Now you should see the installed JRE, select it and click OK.

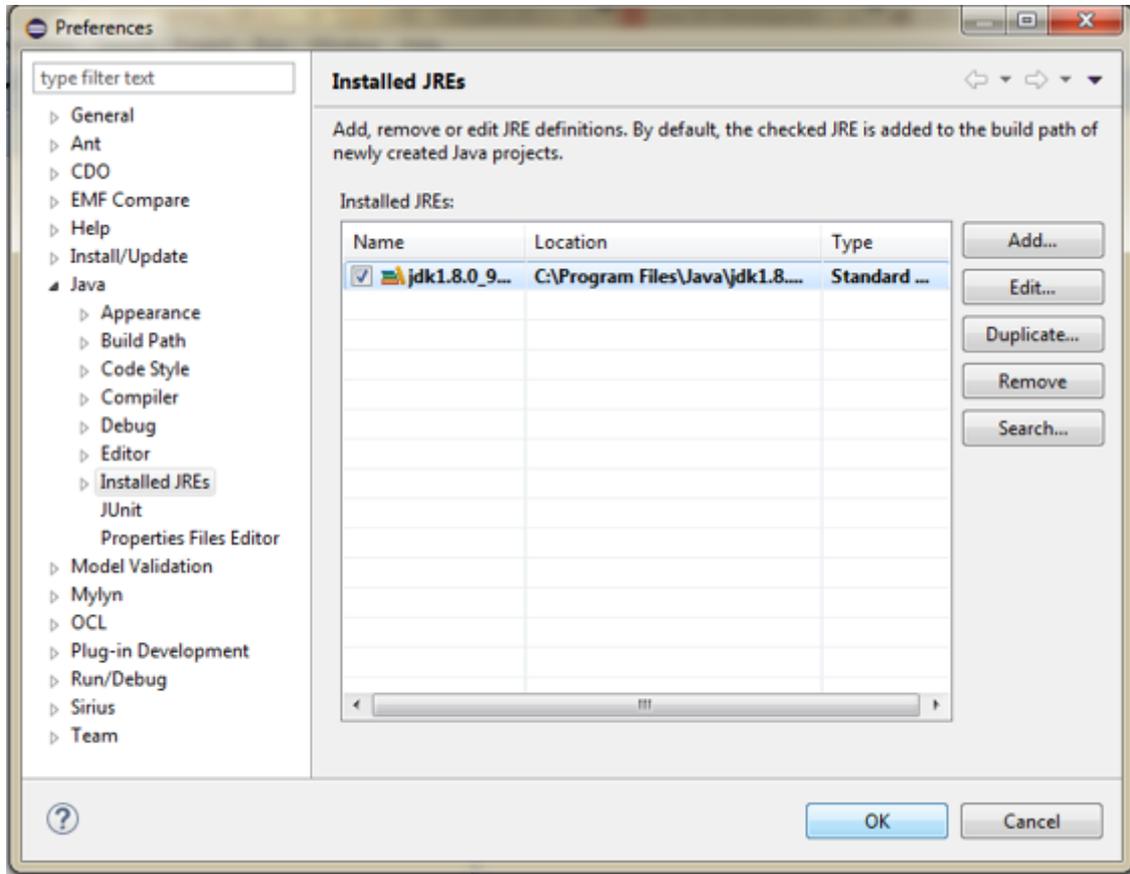


FIGURE 15: INSTALLED JRES

Step-5: Installing Liferay.

- In the Eclipse menu go to Help > Install New Software...

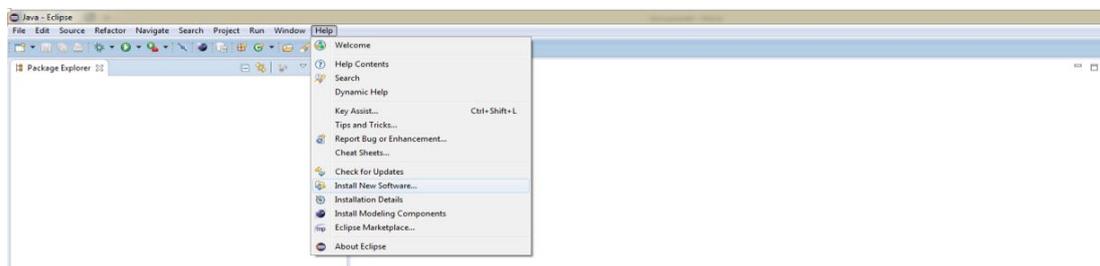


FIGURE 16: HELP > INSTALL NEW SOFTWARE...

- Click "Add..." button to open Add Site dialog.

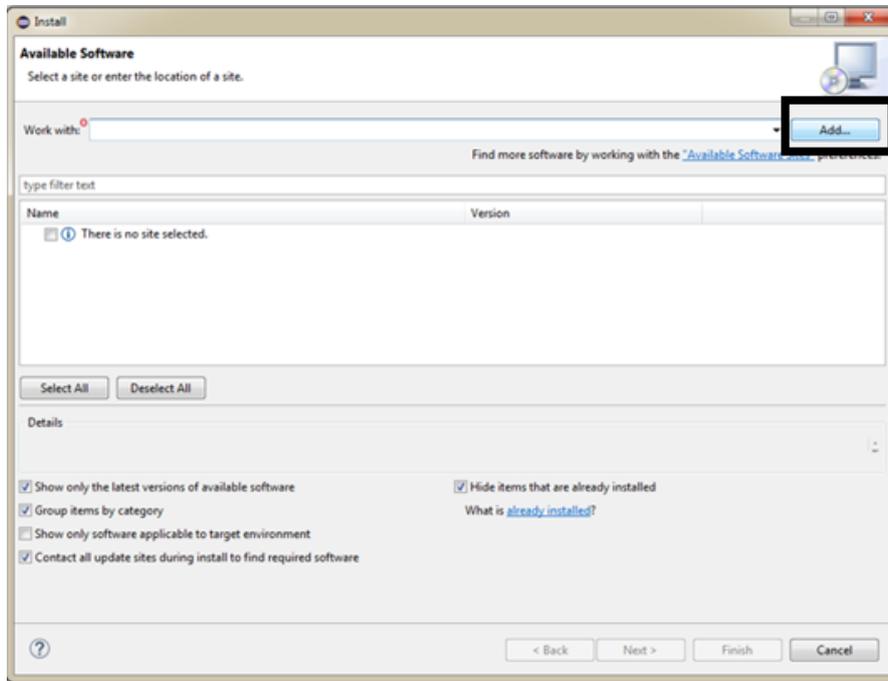


FIGURE 17: OPEN ADD SITE DIALOG

- Type in "Liferay IDE" (version 2.2.2, ga3) for name, and add the following update site as Location and select OK.

<https://sourceforge.net/projects/lportal/files/Liferay%20IDE/2.2.2%20GA3/updatesite/>



FIGURE 18: ADD LIFERAY UPDATE SITE

- Select Liferay IDE from the combo selection box and go Next.

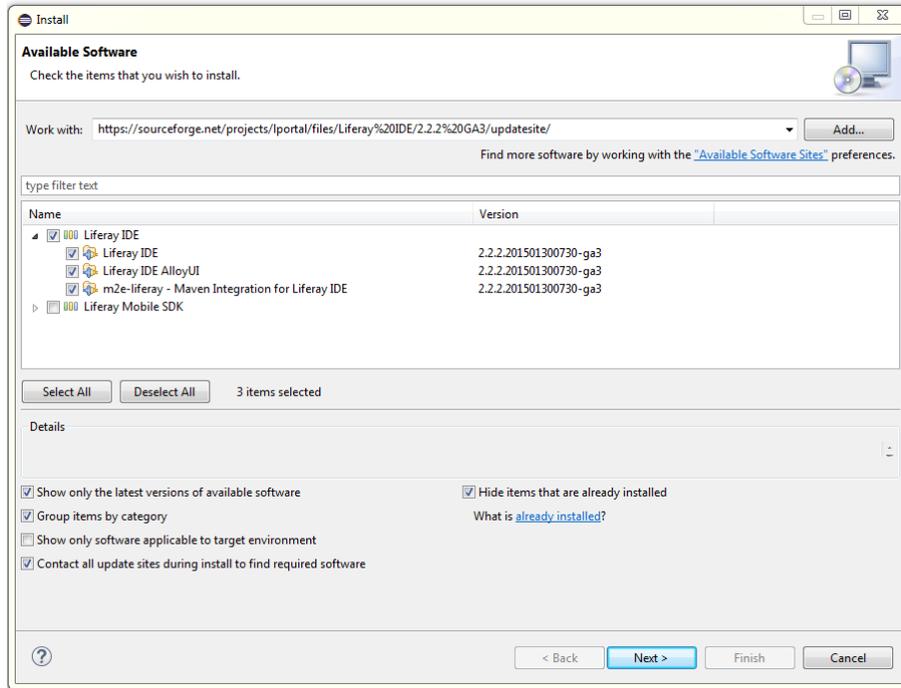


FIGURE 19: INSTALL LIFERAY IDE

- In the next window you will see the install details, go next.

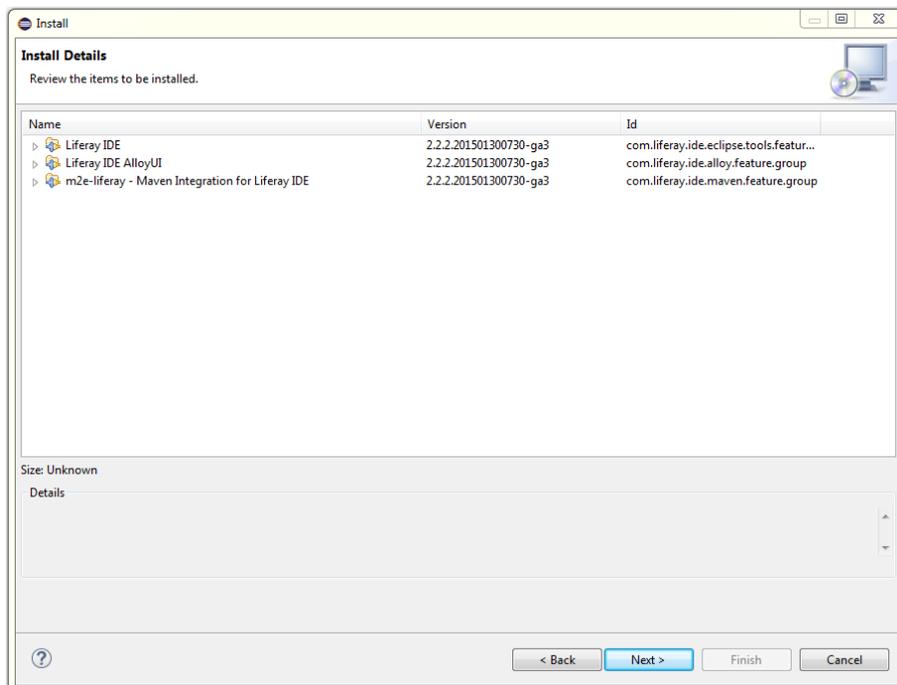


FIGURE 20: LIFERAY INSTALLATION DETAILS

- Accept the terms of the license agreements, click Finish.

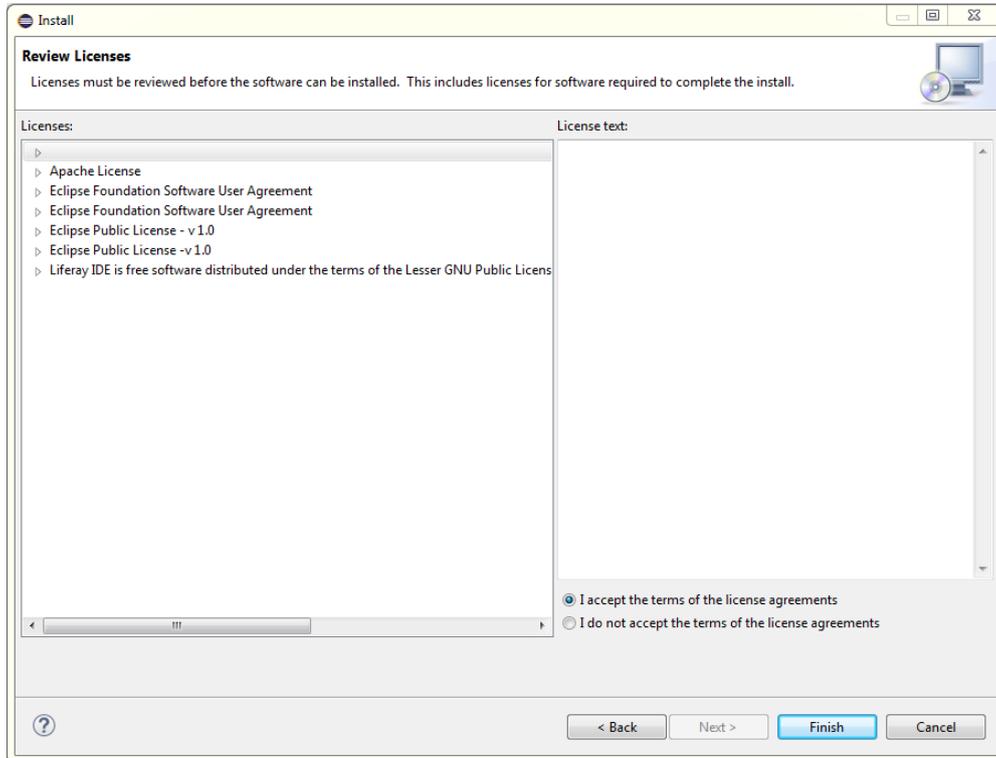


FIGURE 21: ACCEPT LICENSE TERMS

- After plugins download and install, restart eclipse.
- After you restart, go to Help > About Eclipse and you should see an Icon badge for Liferay IDE that shows you have it properly installed.

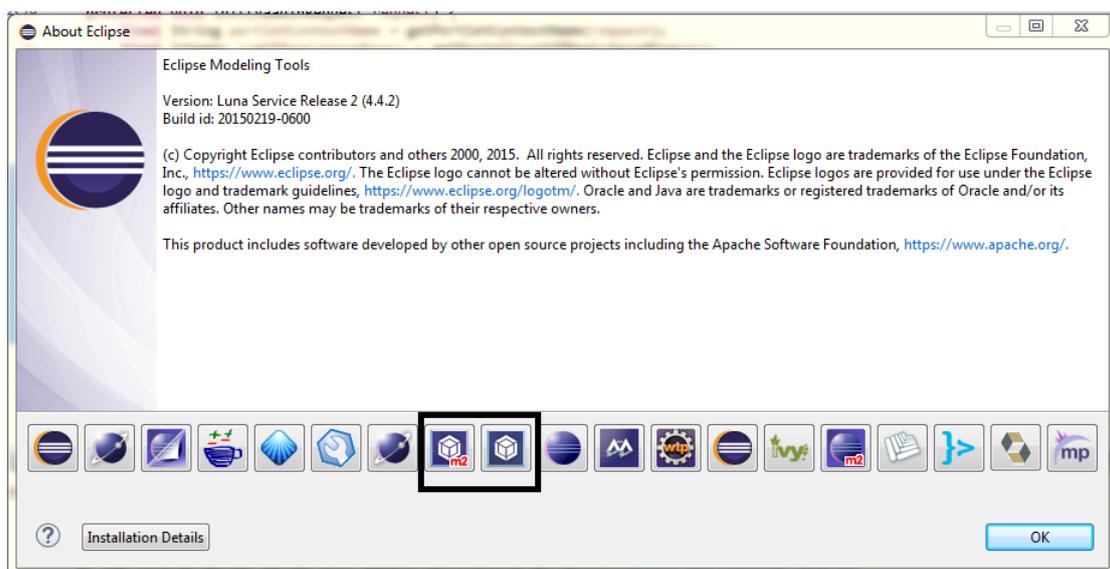


FIGURE 22: LIFERAY IDE INSTALLATION CONFIRMATION

- Browse to the location of your Plug-ins SDK installation.

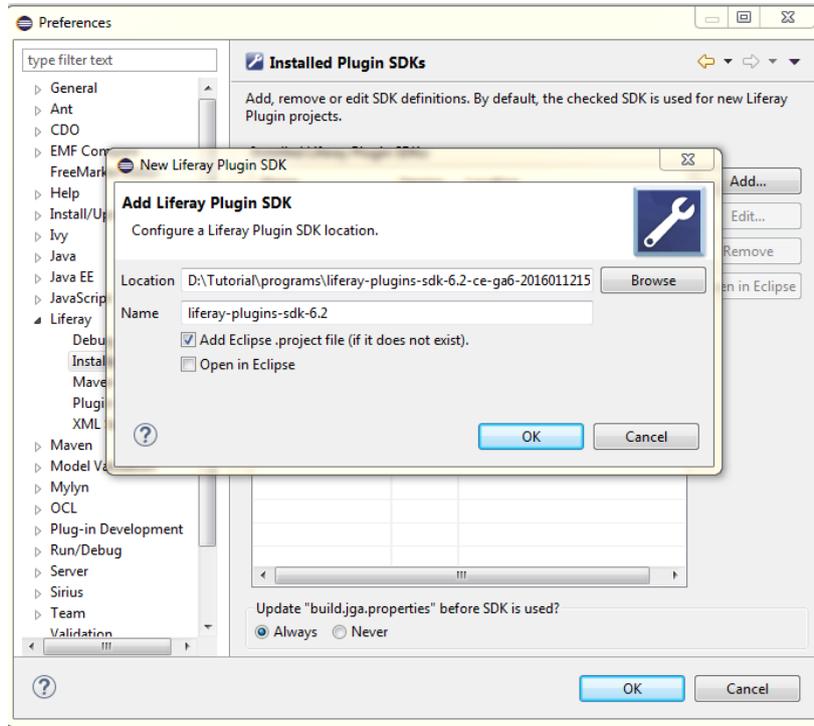


FIGURE 24: ADD LIFERAY PLUGIN SDK

- Select OK and you should see your SDK in the list of Installed SDKs and Select it.

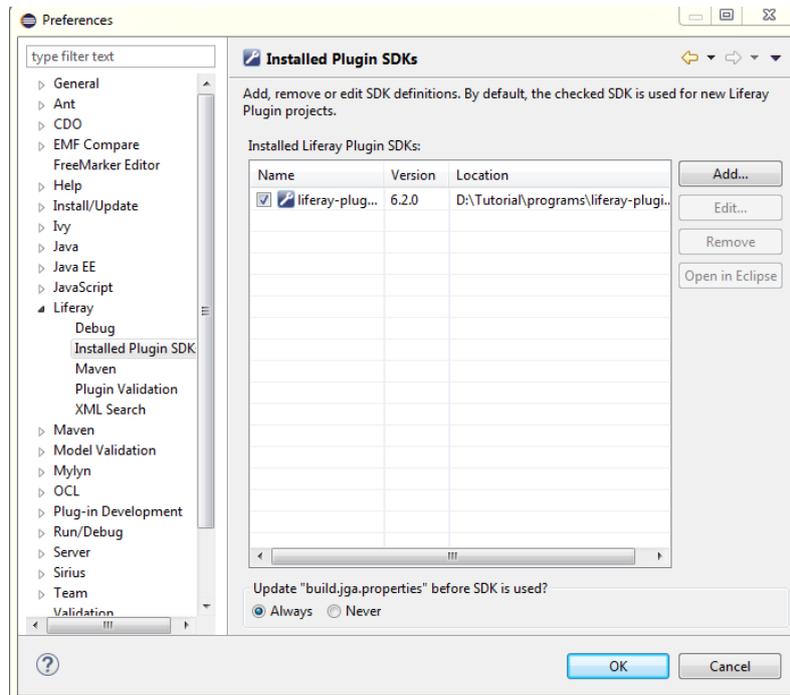


FIGURE 25: INSTALLED SDK

Step 7: Liferay Portal Tomcat Runtime / Server Setup.

- Download the copy of the Liferay Portal Tomcat bundle (version 6.2, ga6) from:
<https://sourceforge.net/projects/lportal/files/Liferay%20Portal/6.2.5%20GA6/>
- In the Eclipse menu go to Window > Preferences > Server > Runtime environments

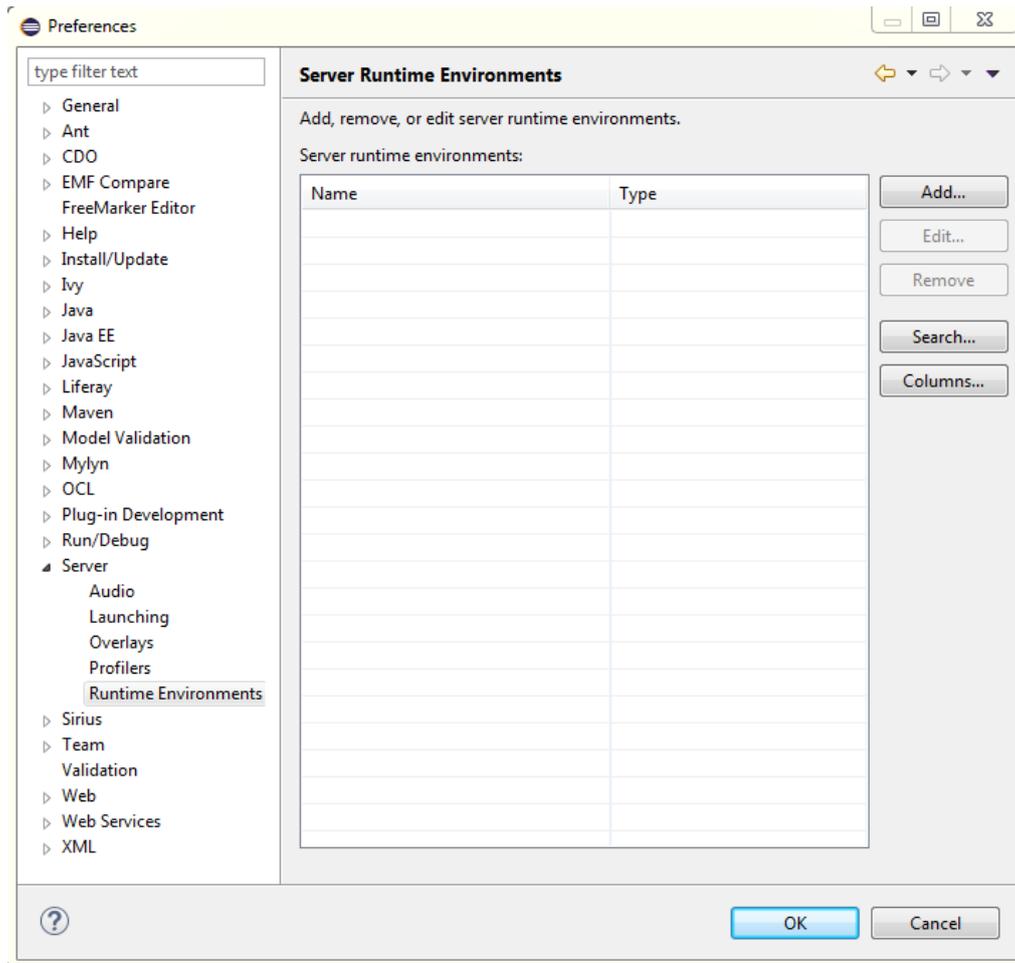


FIGURE 26: SERVER RUNTIME ENVIRONMENT VIEW

- .Click Add to add a new Liferay runtime
- In the Dialog box find Liferay v6 Tomcat under the Liferay, Inc. category and click Next.

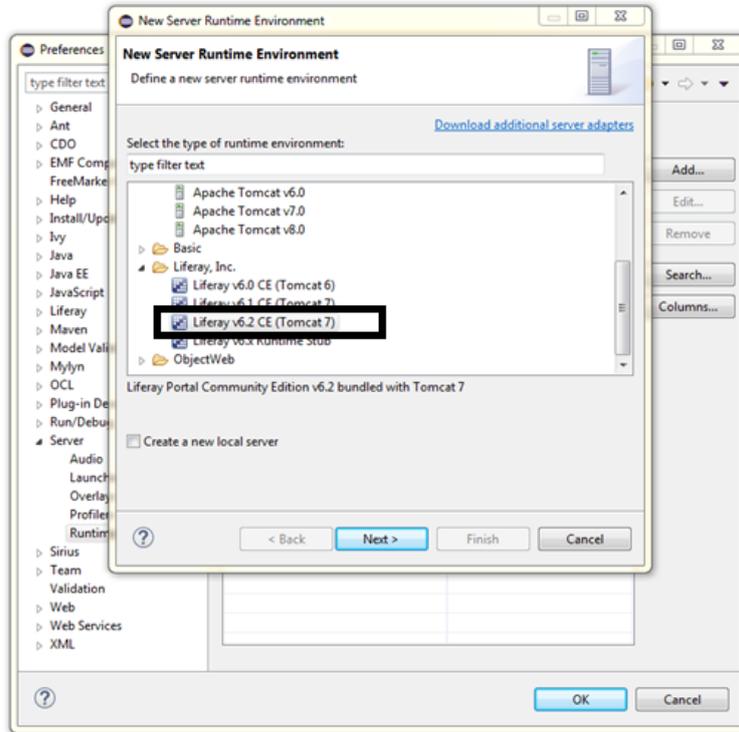


FIGURE 27: SELECT SERVER

- In the next step browse to the location of the liferay-portal-6.0.2 directory and click finish.

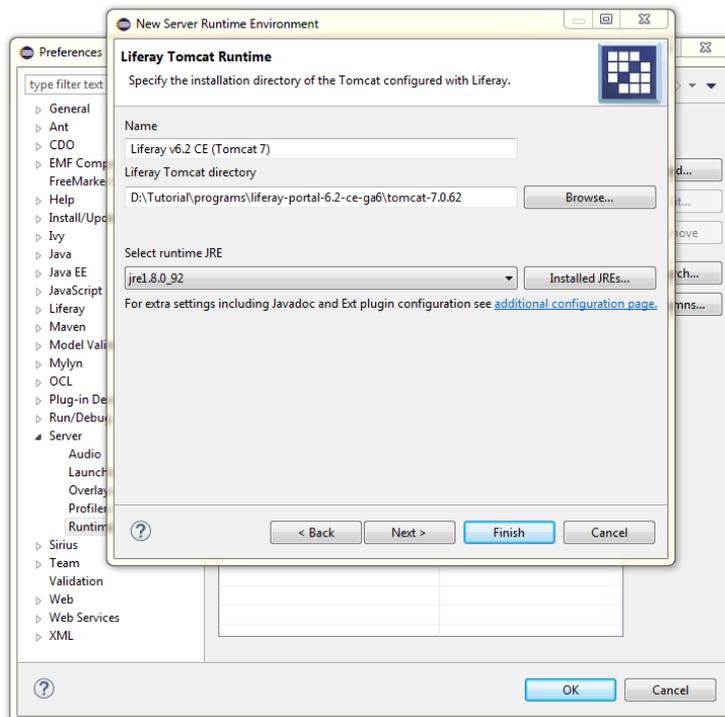


FIGURE 28: LIFERAY TOMCAT DIRECTORY LOCATION

Step 8: Install Vaadin Plugin for Eclipse

- In the Eclipse menu go to Help > Install New Software...
- Click "Add..." button to open Add Site dialog.
- Type in "Vaadin Plugin" for name, and add the following update site as Location.
<http://vaadin.com/eclipse> and select OK.

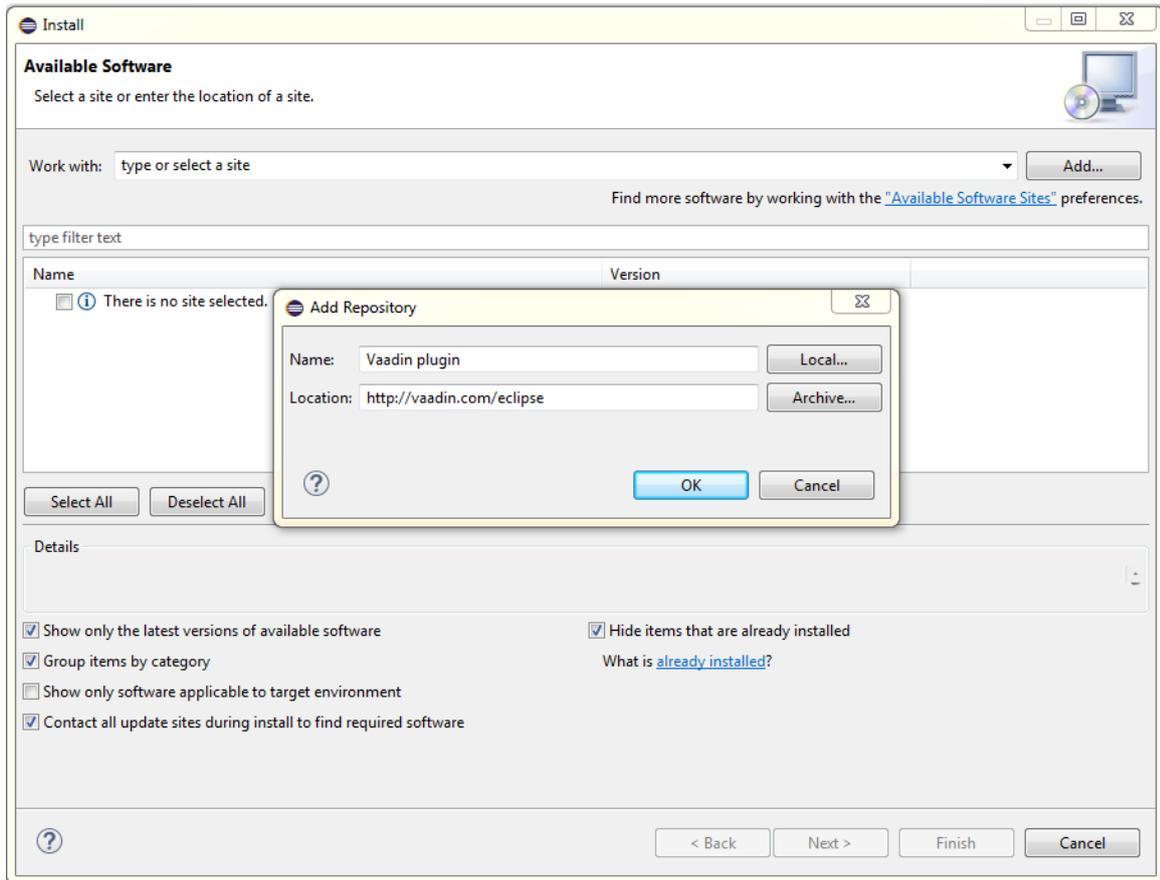


FIGURE 29: ADD VAADIN UPDATE SITE

- Select everything from the combo selection box and click Next.

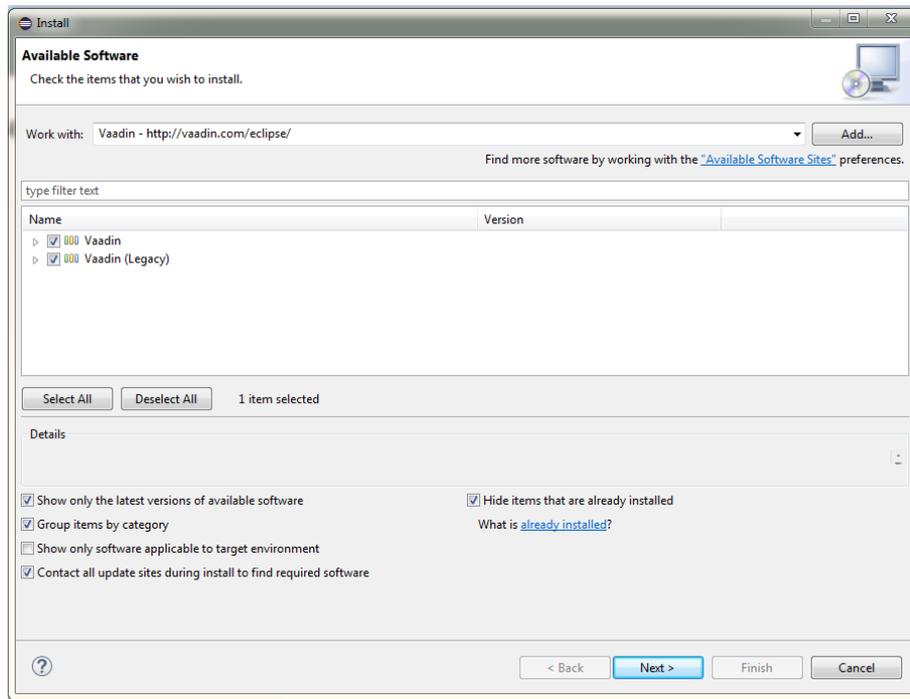


FIGURE 30: SELECT VAADIN SOFTWARE

- In the next window you will see the install details, go next.
- Accept the terms of the license agreements, click Finish.

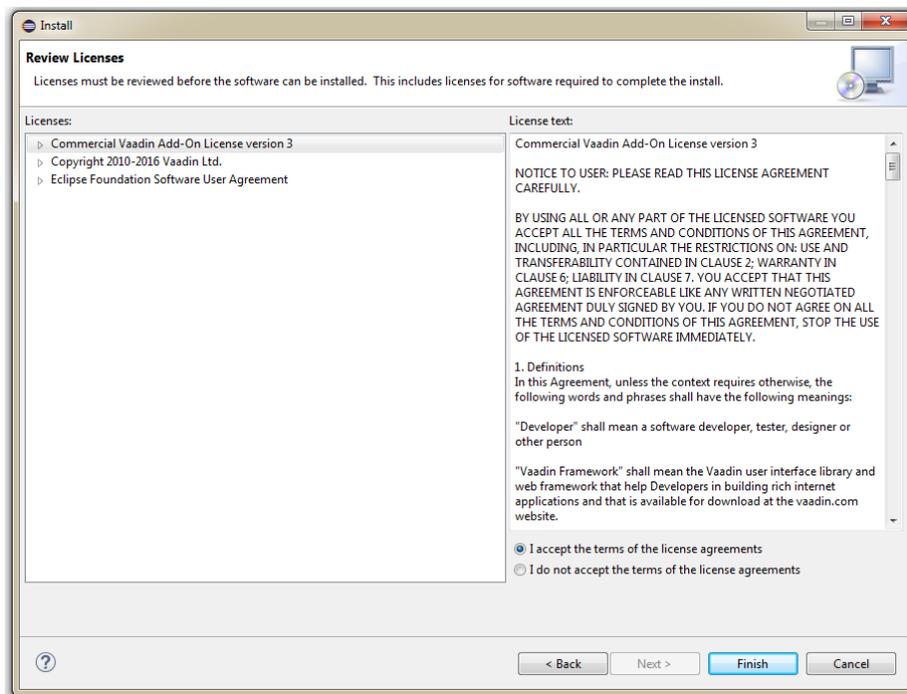


FIGURE 31: VAADIN REVIEW LICENSE

Contract Number
619347

- After the plugins download and install, restart eclipse.

Step 9: Install Resource Bundle Editor

- In the Eclipse menu go to Help > Eclipse Marketplace

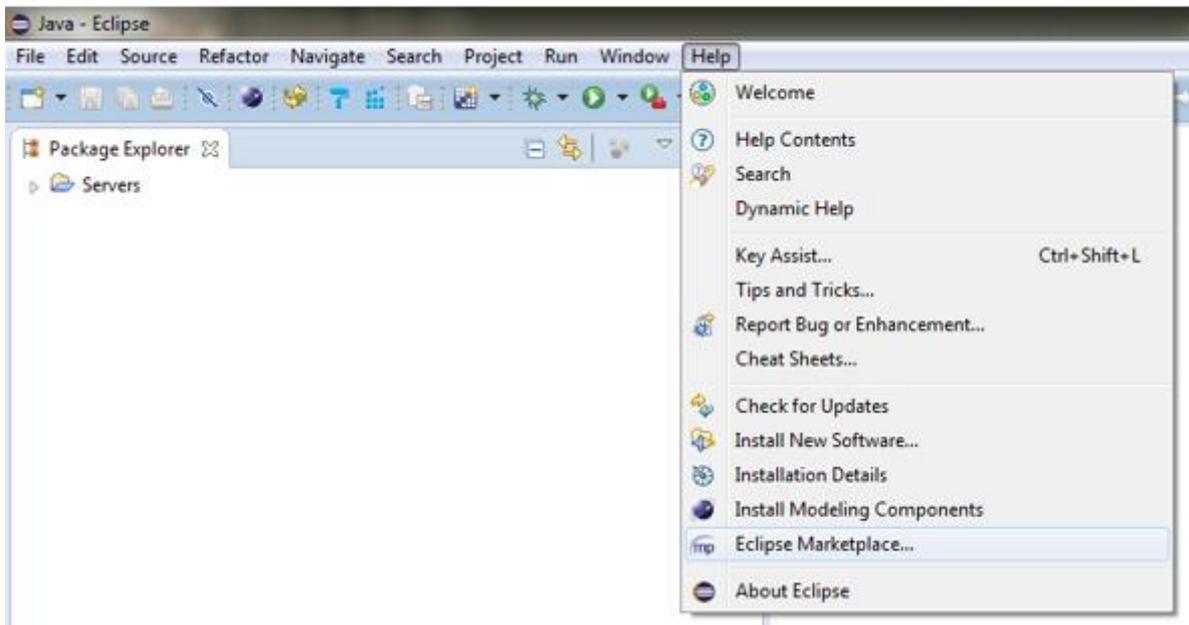


FIGURE 32: HELP > ECLIPSE MARKETPLACE

- In the dialog box type in “resource” in the search field

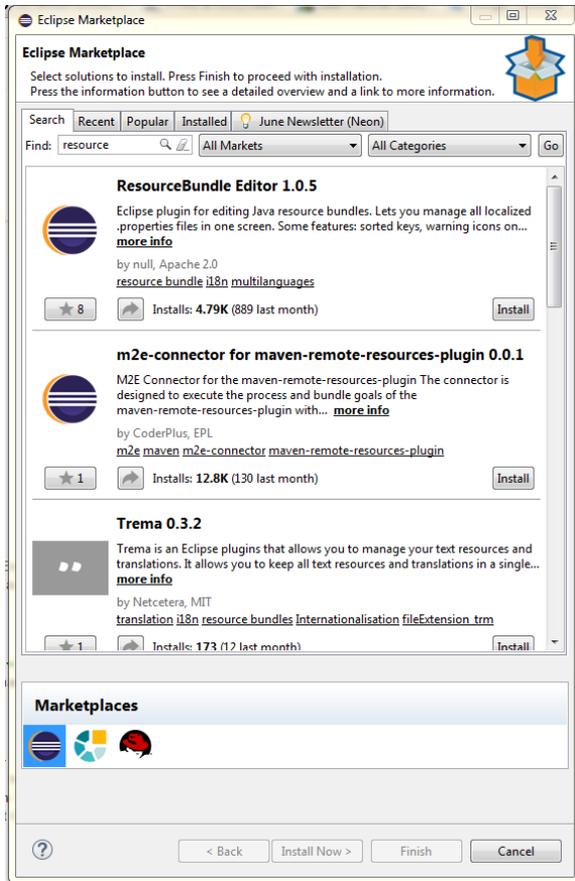


FIGURE 33: RESOURCE BUILDER EDITOR

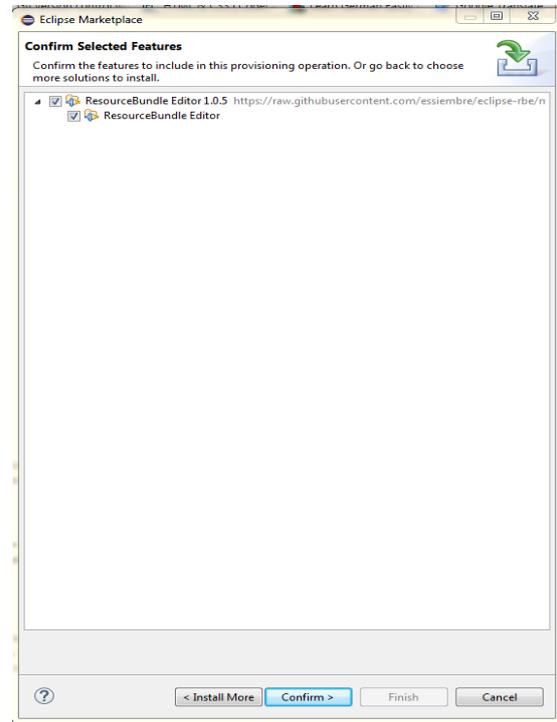


FIGURE 34: INSTALLING RESOURCE BUILDER EDITOR

- Install the Resource Builder Editor
- In the next window, click on the confirm button
- Select the terms of the license agreement and click finish

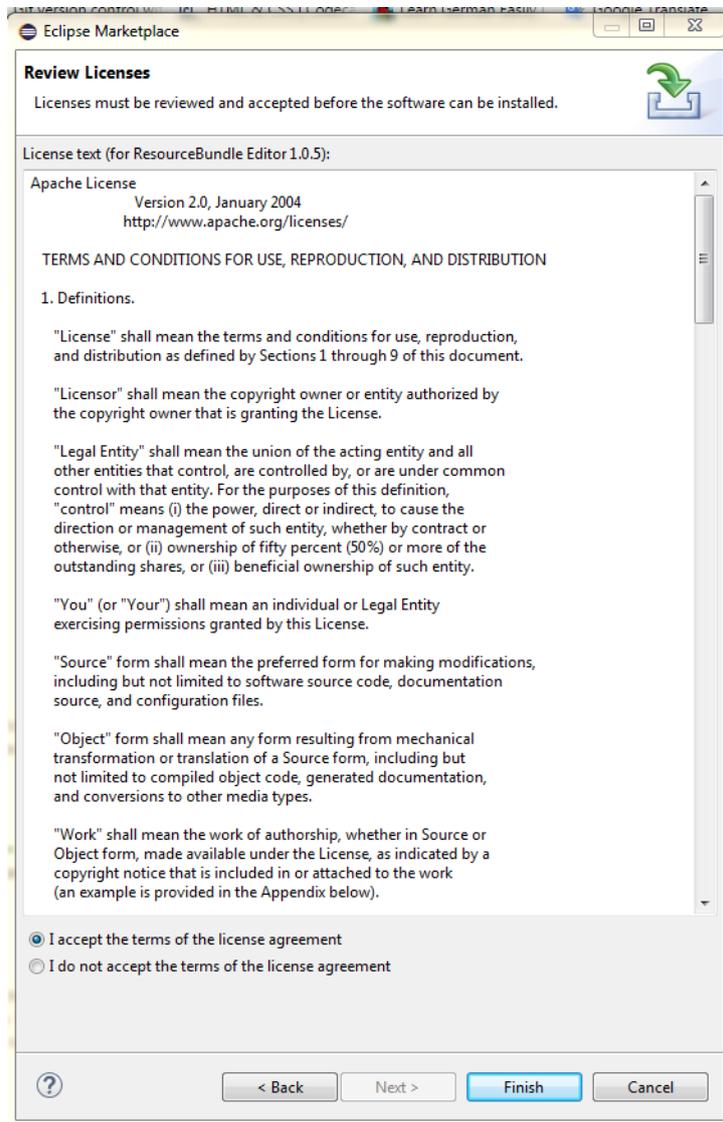


FIGURE 35: AGREE TO LICENSE AGREEMENT

Step 10: Importing branch from git.

- Switch to the Git view in your workspace.
- For creating a portlet, one must have the projects from Services, Knowledge, Portal Build, Plugins, Cache and Hook.
- Cloning the projects from service branch
 - Select the Clone a Git Repository option.

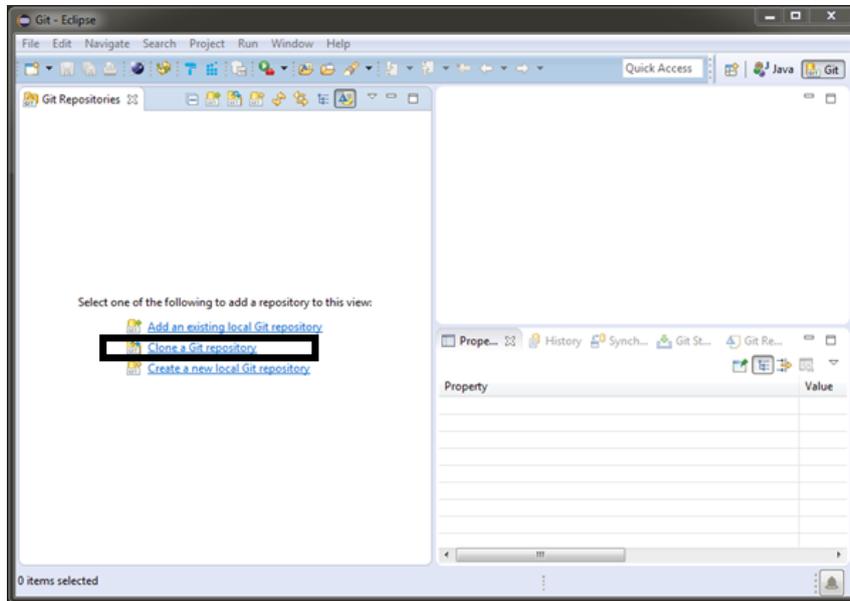


FIGURE 36: CLONE GIT REPOSITORY

- In the dialog box, add the following URI for getting the projects from git and click Next
 ssh://git@git.list.lu/eagle/services.git

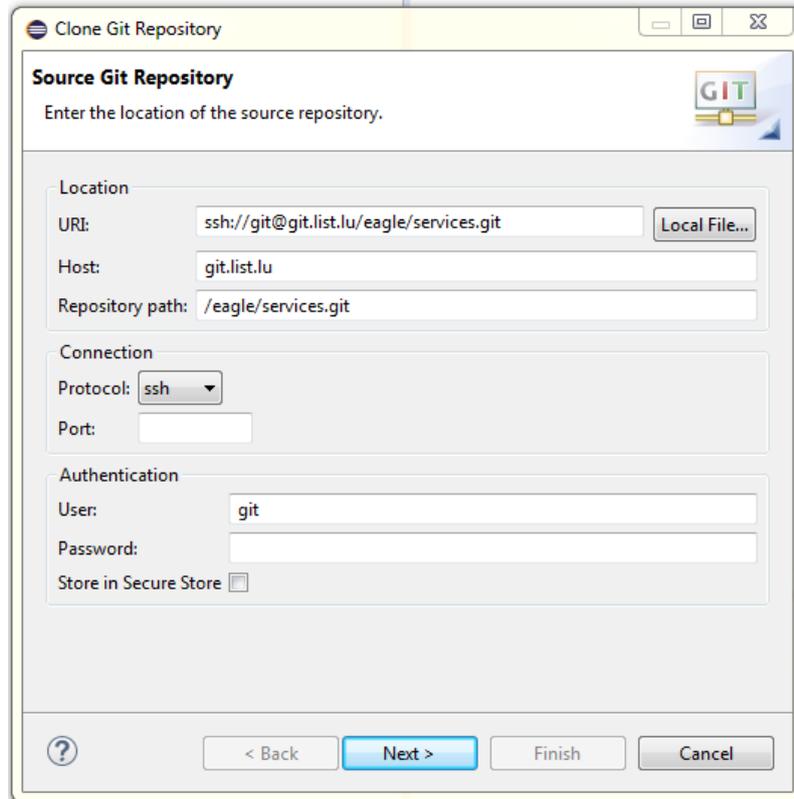


FIGURE 37: SOURCE REPOSITORY LOCATION

- From the next dialog box, Select Master as Branch

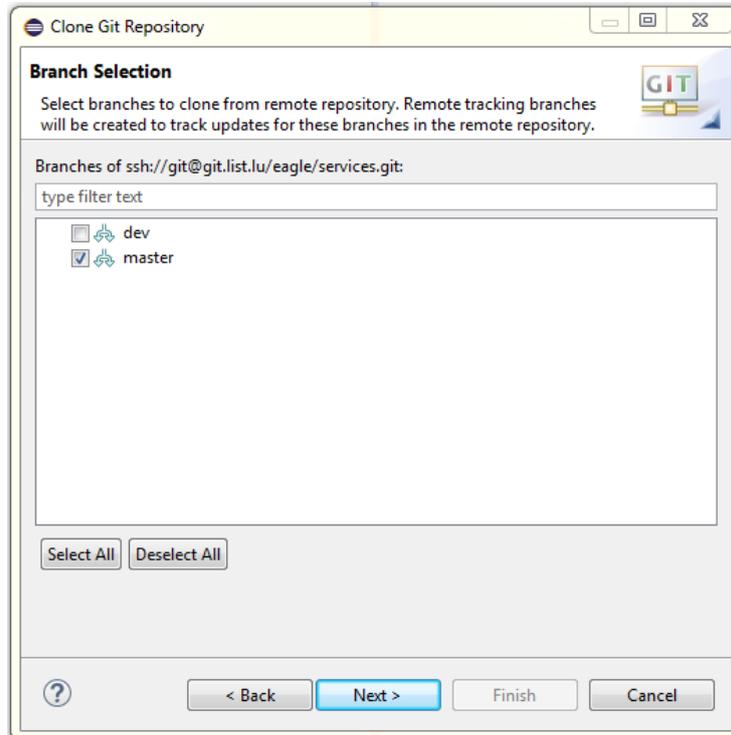


FIGURE 38: BRANCH SELECTION

- In the next step, Browse to the desired directory, where you want to save it.

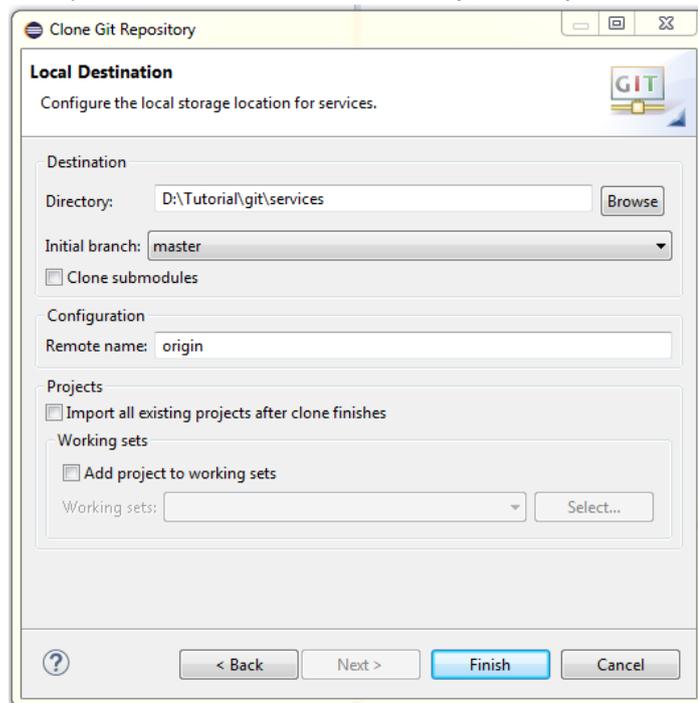


FIGURE 39: SELECT LOCAL DIRECTORY

- Click Finish.
- You will get the following repository

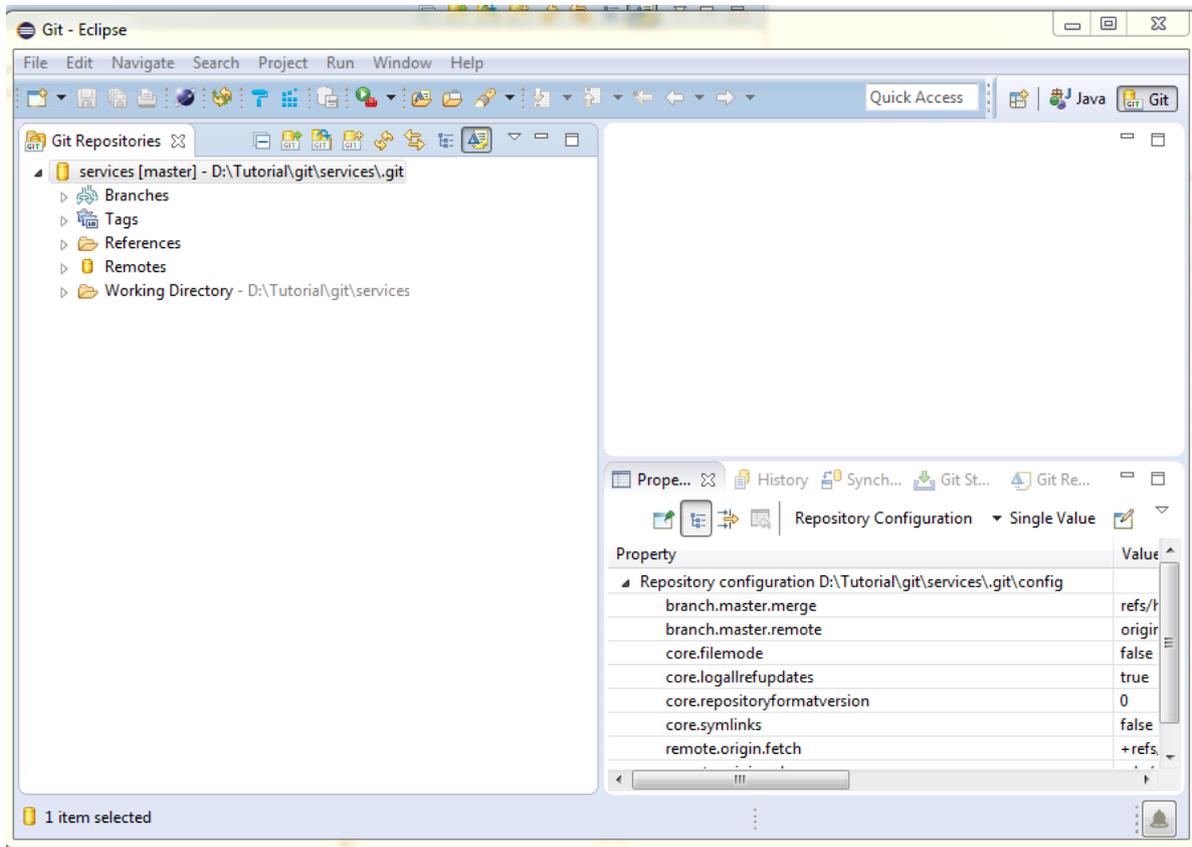


FIGURE 40: CLONED REPOSITORY

Note: You should follow same steps as mentioned in Step 9 above, to clone the Other Repositories. Following are the URI that should be used.

- Knowledge : `ssh://git@git.list.lu/eagle/knowledge-space.git`
- Portal Build, Portal Plugins and Hook is all under the “**portal**” repository in git. So please use the URI: `ssh://git@git.list.lu/eagle/portal.git`
- Cache: `ssh://git@git.list.lu/eagle/cache.git`

Step 10: Importing projects from git.

- Under package Explorer, Right click > Import.

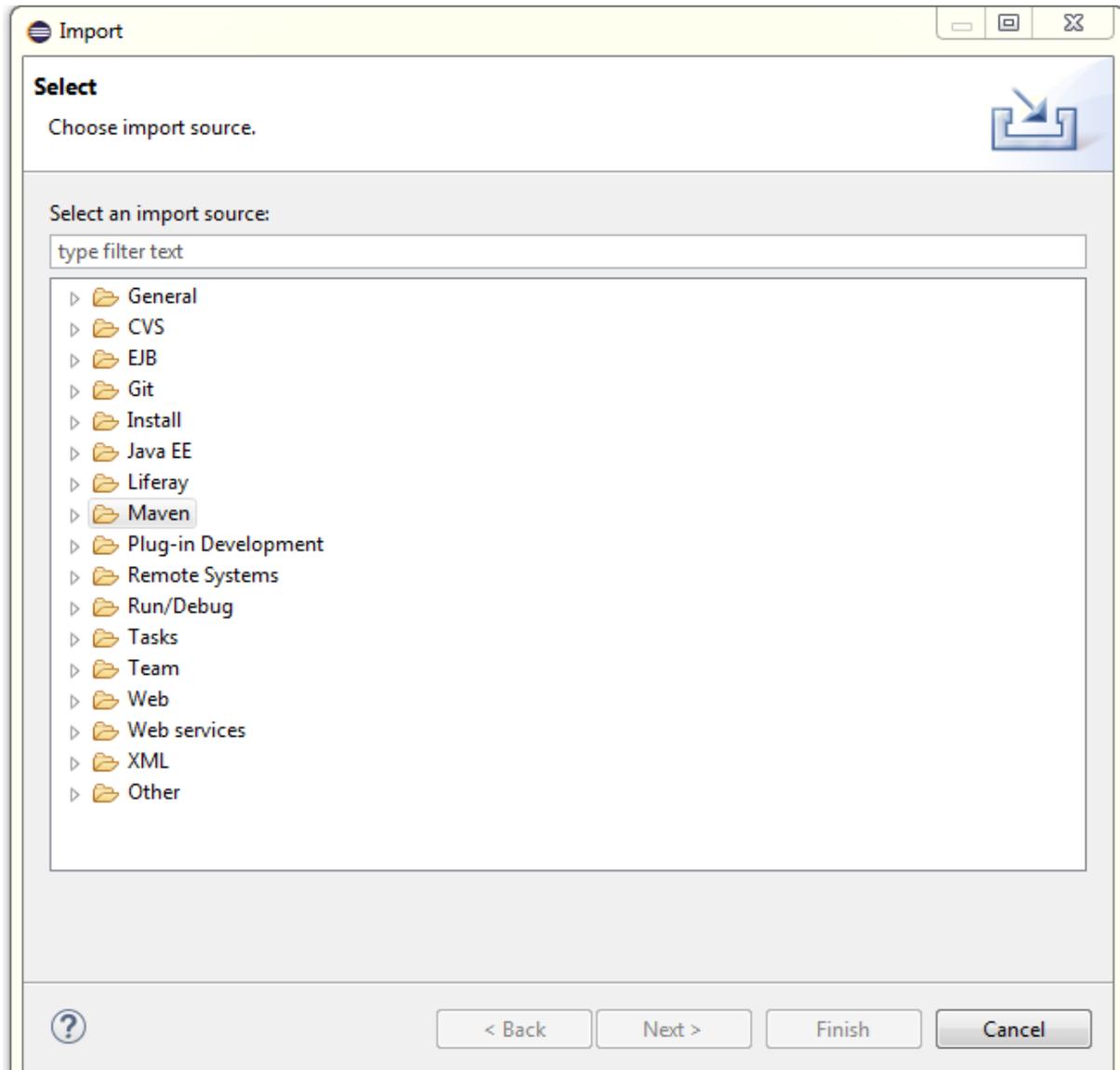


FIGURE 41: IMPORT DIALOG BOX

- Select Maven category and expand it. Under Maven, select the Existing maven Projects Option and click next.

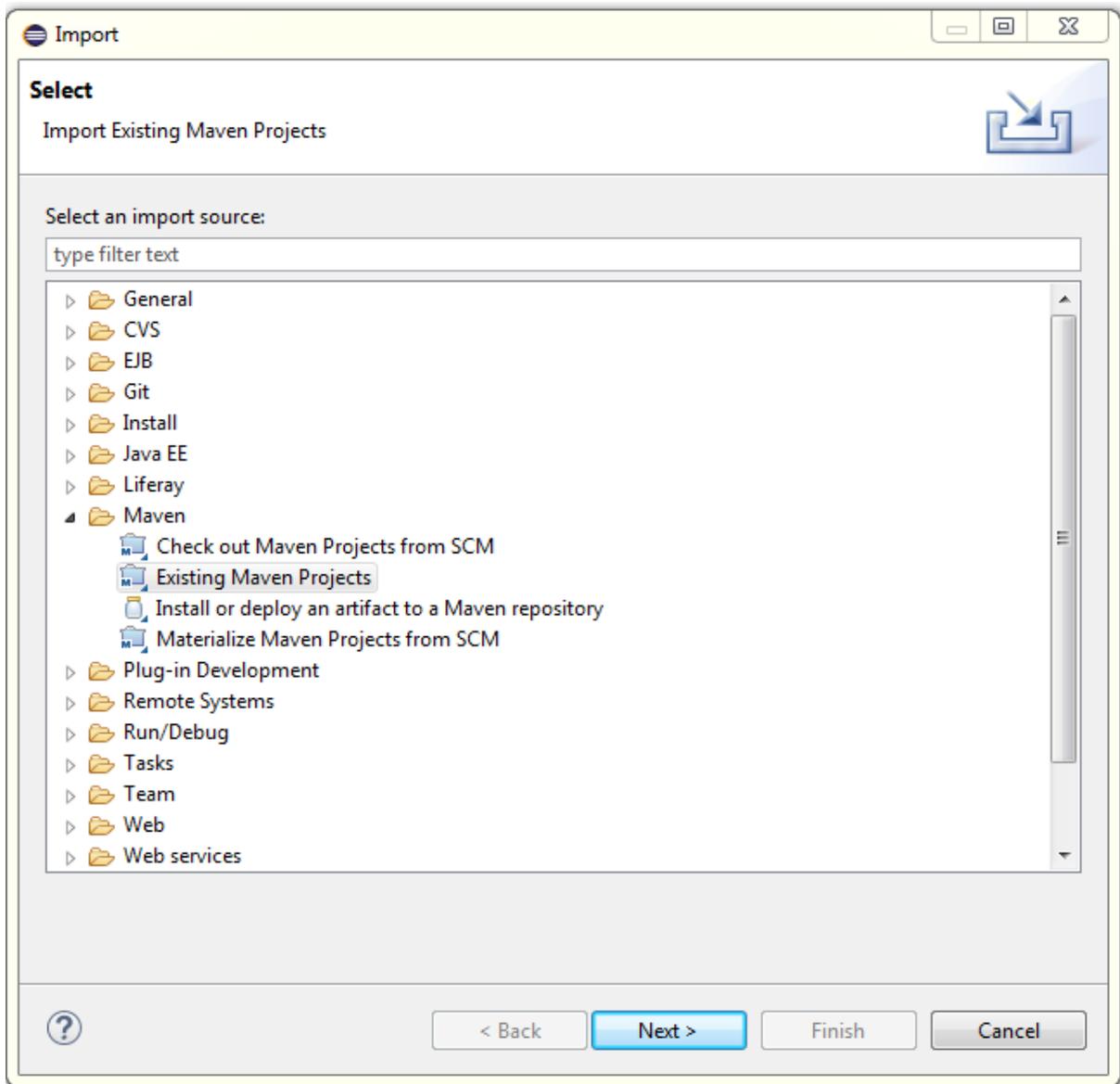


FIGURE 42: MAVEN CATEGORY

- In the next Window, browse to the location where you saved the git branches, while cloning them.
- The figure below shows the import from services branch. Select the projects you want to import.

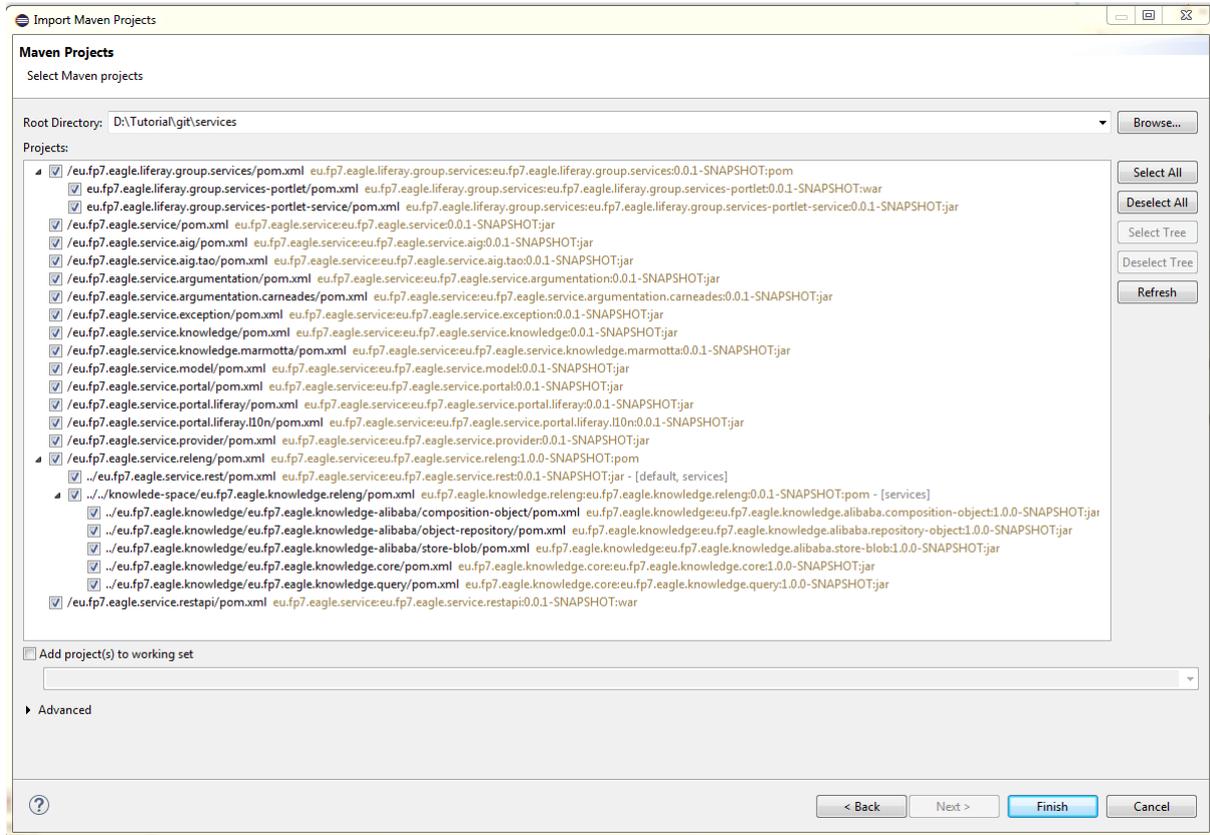


FIGURE 43: MAVEN PROJECTS FROM SELECTED DIRECTORY

- Next, click on the Add project(s) to working set check box.
- Give “services.plugins” as the working set name and click Finish.

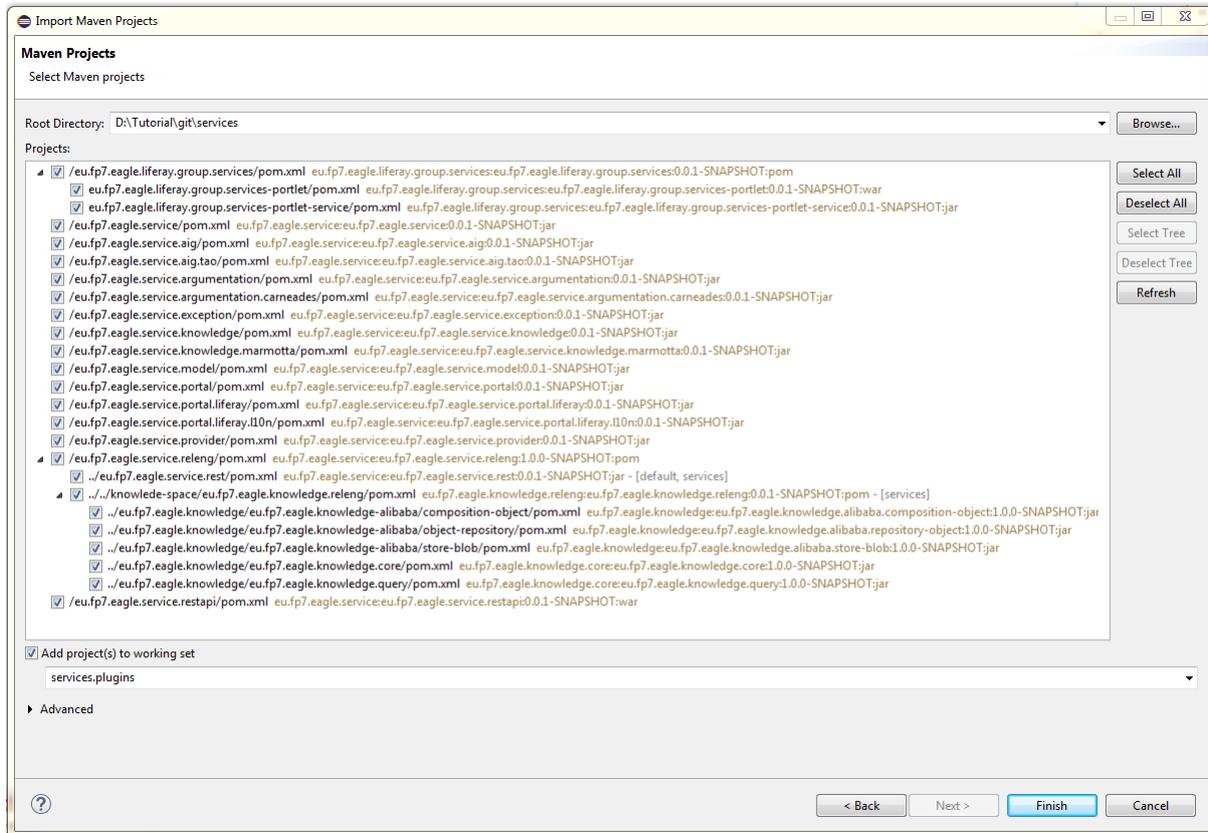


FIGURE 44: ADD WORKING SET NAME

- This will import all the projects under Services branch in the workspace.

NOTE: Using the same steps, import all the projects from other repositories also.

Step 11: Package Structure

- The Package structure after importing all the projects should be structured as follows under corresponding working set.

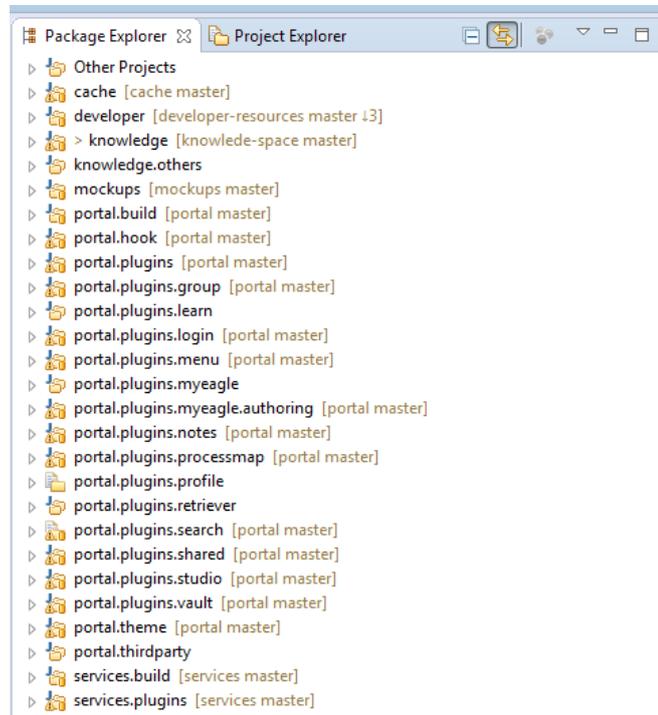


FIGURE 45: PACKAGE STRUCTURE

- In order to build the workspace, launch the maven run configuration located at: `eu.fp7.eagle.portal.releng/eu.fp7.eagle.portal.releng.all.launch`
 - Make sure that the run configuration is launch in the online mode the first time, so as to allow maven to download all the required libraries from the maven central and the other repositories defined in the releng pom.xml

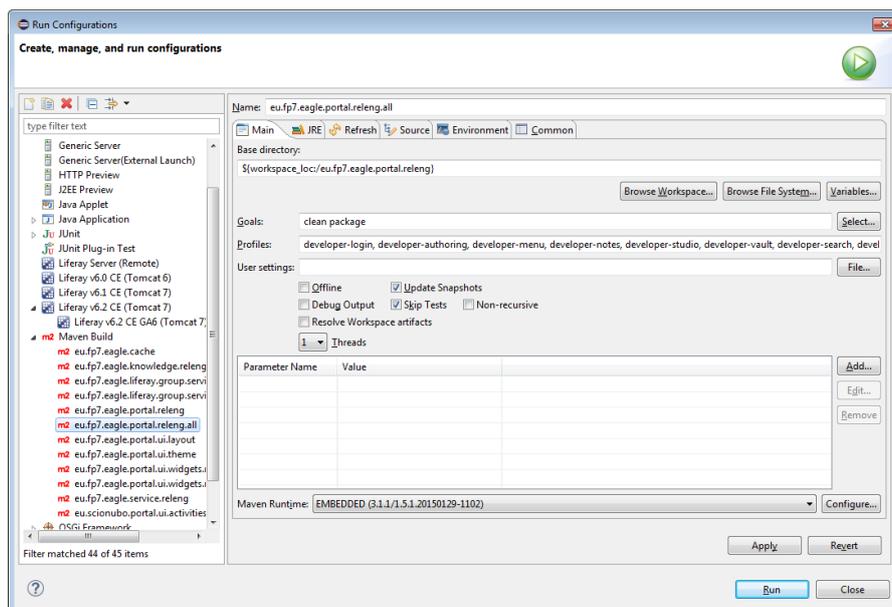


FIGURE 46: RUN CONFIGURATION FOR MAVEN BUILD

Step 12: Creating maven project using Liferay Vaadin archetype

- Under package Explorer, Right click > New > Other.

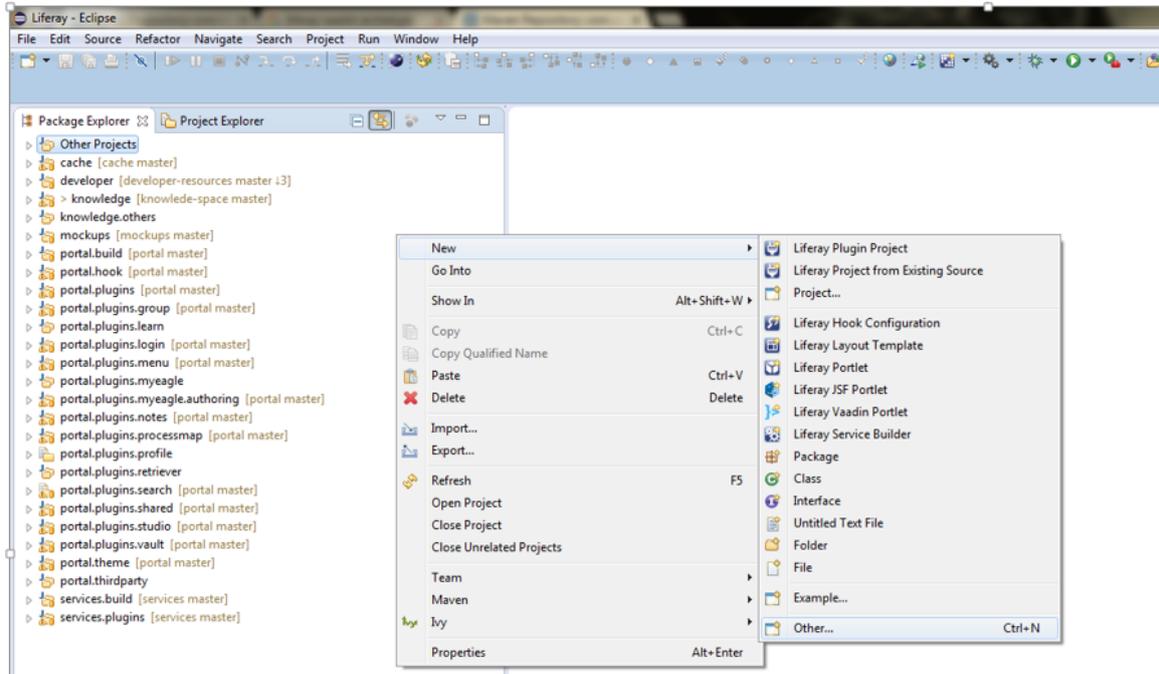


FIGURE 47: RIGHT CLICK > NEW > OTHER.

- Select Maven > Maven Project from the dialog box and click Next.

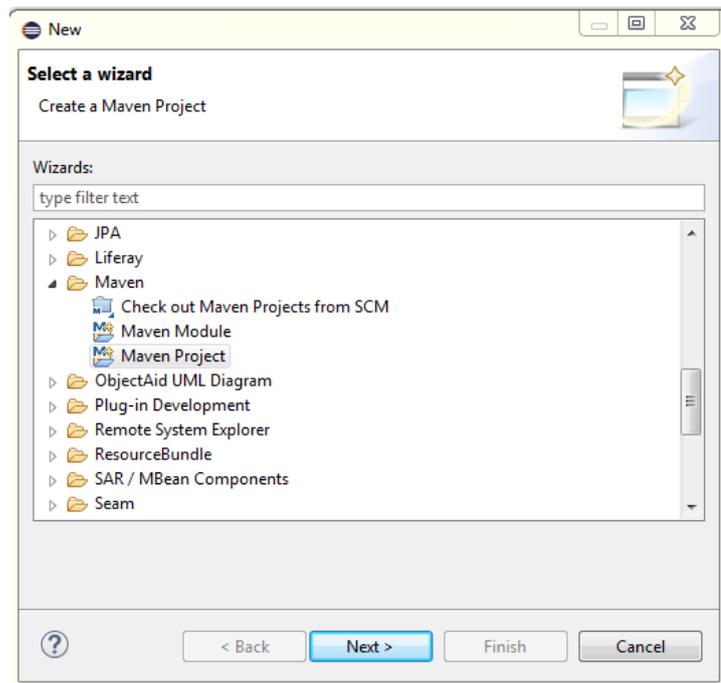


FIGURE 48: SELECT MAVEN PROJECT

- In the next dialog box, make sure that the option for creating simple maven project is unchecked.
- Add this project to the desired working set using the drop down option and click Next.

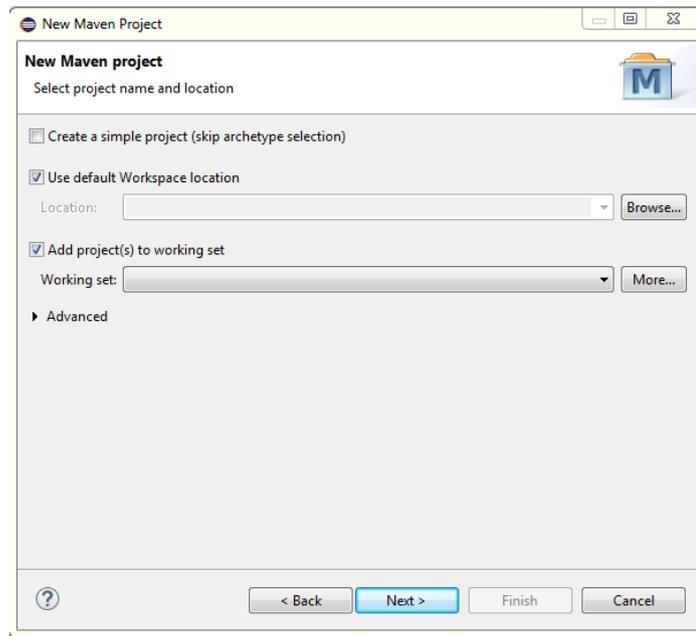


FIGURE 49: NEW MAVEN PROJECT

- In the next step, type in Liferay in the filter box, and select vaadin-archetype-liferay-portlet (version 7.5.0) as an archetype and click Next.

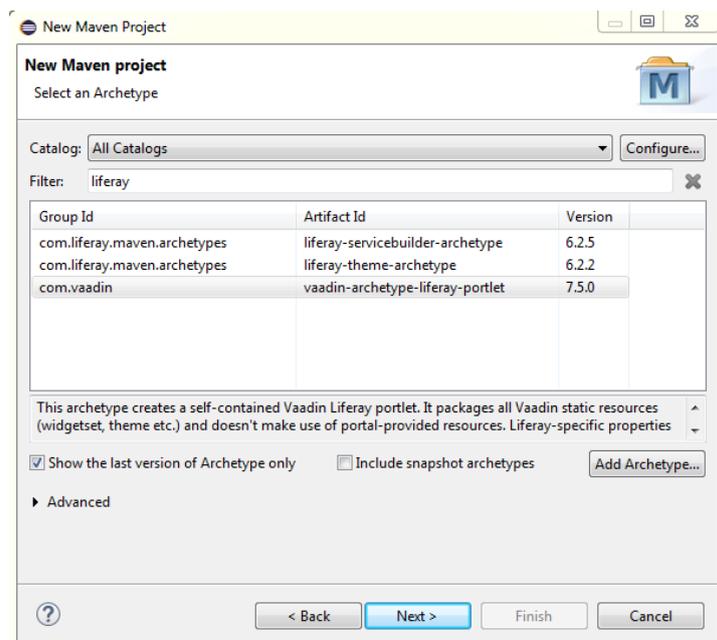


FIGURE 50: SELECT ARCHETYPE

- In the next step, specify the Archetype parameter (give name which suits your requirement).
- Group Id must contain “eu.fp7.eagle.portal.ui.portlet” as prefix and then the name of portlet you are working on. For example, if I am creating a test portlet, my Group Id will be “eu.fp7.eagle.portal.ui.portlet.test”.
- Edit the properties available from the archetype accordingly.
- Version must be [6.2.2,0]
- UiClassName can be given with a prefix Eagle (example, EagleDashboardUI), or just the portlet you are working on (example, DashboardUI). Just give some meaningful name.
- Theme must be “eagle”.
- LiferayCategoryName must be “Eagle”.

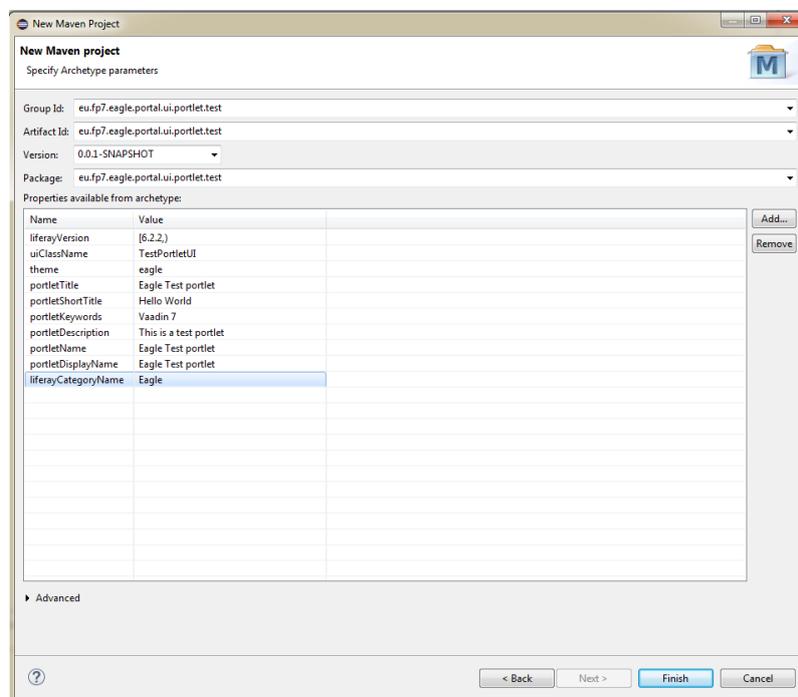


FIGURE 51: ARCHETYPE PROPERTIES

- After the project is successfully created, you should get the structure as in Figure 52.

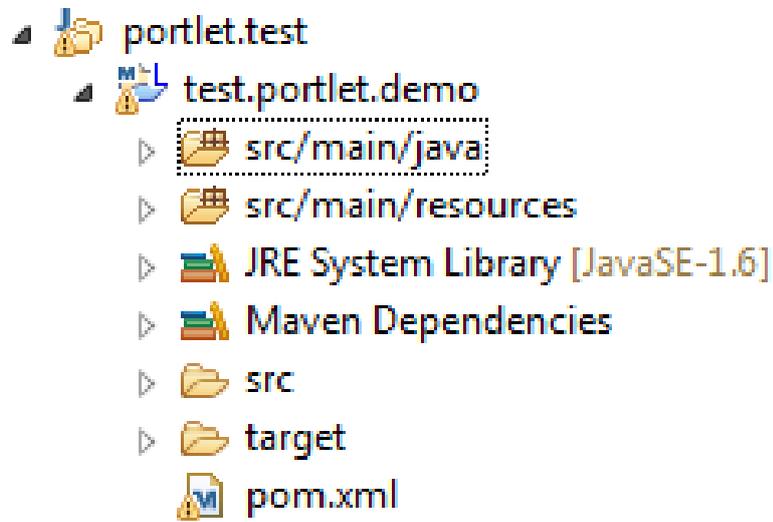


FIGURE 52: PROJECT STRUCTURE

- Start the Liferay server, and drag and drop the portlet project to deploy it to the server.
- After successfully deploying the portlet into the server, open the Liferay portal in web browser.
- Sign in using the correct credentials.
- Click + Add, It will open a window on the left side of the screen.

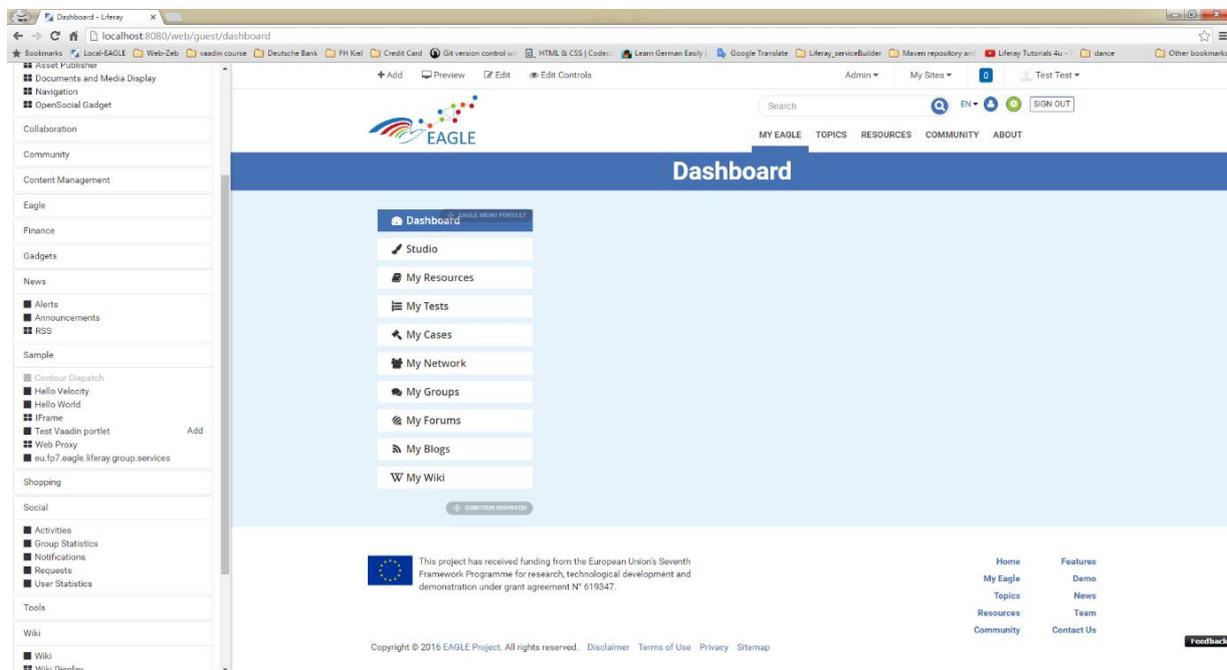


FIGURE 53: LIFERAY PORTAL VIEW

- Look for the portlet you deployed and add it to the page you wish to add in.

5.2 EAGLE Knowledge Base Development

The EAGLE Knowledge Base makes extensive use of Apache Marmotta and its extensible architecture. Figure 54 shows an overview of the general architecture of EAGLE KB.

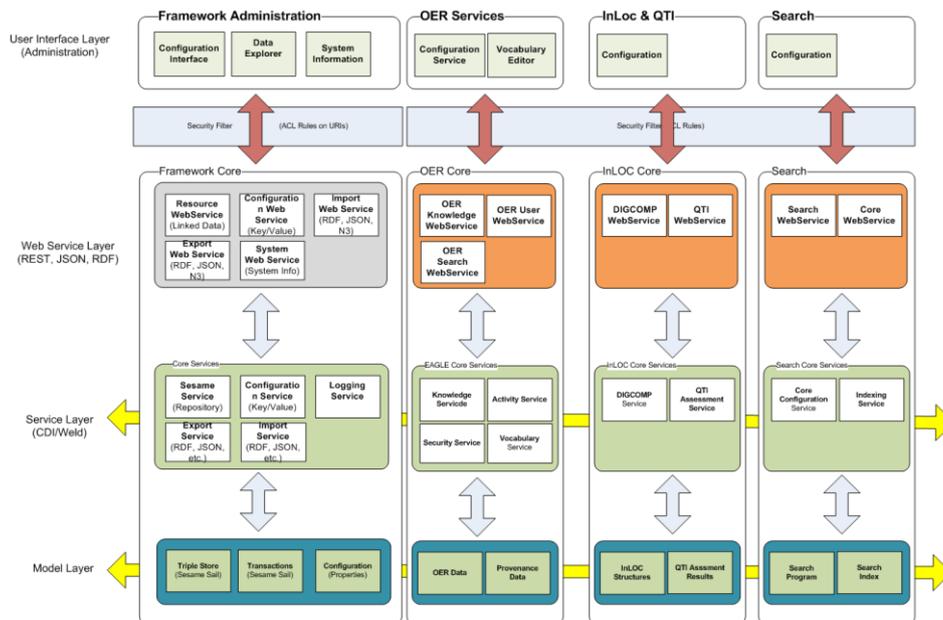


FIGURE 54: EAGLE KB ARCHITECTURE OVERVIEW

As shown Figure 54, the EAGLE KB uses the core of Apache Marmotta, thus receives standardized services, storage capabilities. Thus, the management of OER metadata is possible at triple-level out of the box. In order to add higher-level OER related functionality, such as adding, updating and adapting OER's, additional functional modules are provided and assembled to comprise the EAGLE KB.

The remainder of this section shows the required development steps when adding a new module to the EAGLE KB.

Step 1: Download Source Code

Follow the steps in section 4.2.3 in order to obtain the source code.

Step 2: Prepare Development Workspace

Note: Development of EAGLE KB follows the principles of Apache Maven, therefore any Java development environment with Apache Maven support (Eclipse, Netbeans, IntelliJ) may be used. However, for the subsequent steps, the Eclipse IDE has been used.

Open Eclipse and create a new workspace, or use the workspace from section 5.1, step 3. Choose "Import ..." menu item from the "File" menu. In the Import Wizard, as shown in Figure 55, choose "Existing Maven Projects" and click "Next".

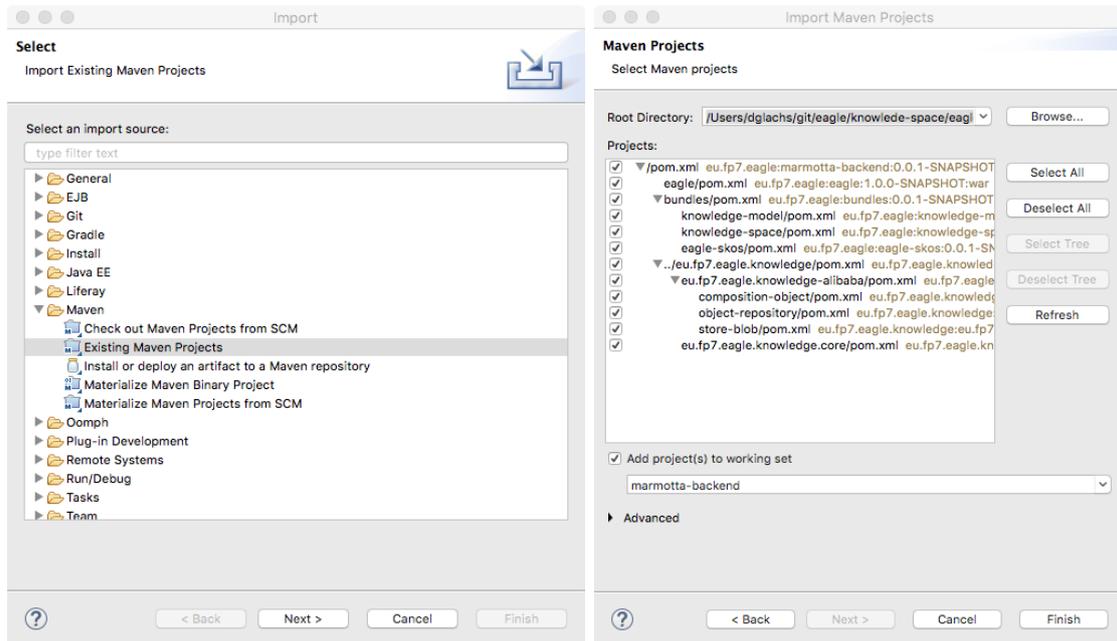


FIGURE 55: IMPORT MAVEN PROJECTS

Use the “Browse...” button to select the source code directory named “eagle-knowledge-backend” from the file system. The pom.xml file lists all dependencies and thus will ensure, all listed projects are created in the workspace. The “Project Explorer” will show the distinct projects as outlined in Figure 56.

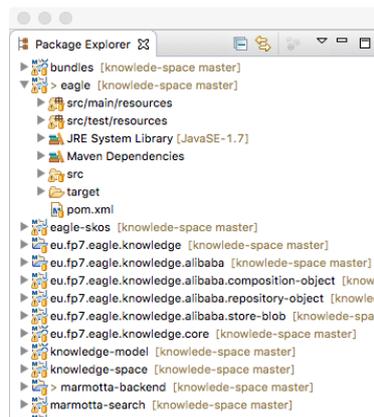


FIGURE 56: EAGLE KNOWLEDGE BASE WORKSPACE

The project holding the service runtime is named “eagle”.

Step 3: Execute Knowledge Backend

Right-Click the “eagle” project and choose “Run As ... → Maven Build ...”! The Eclipse Run Wizard as shown in Figure 57 appears.

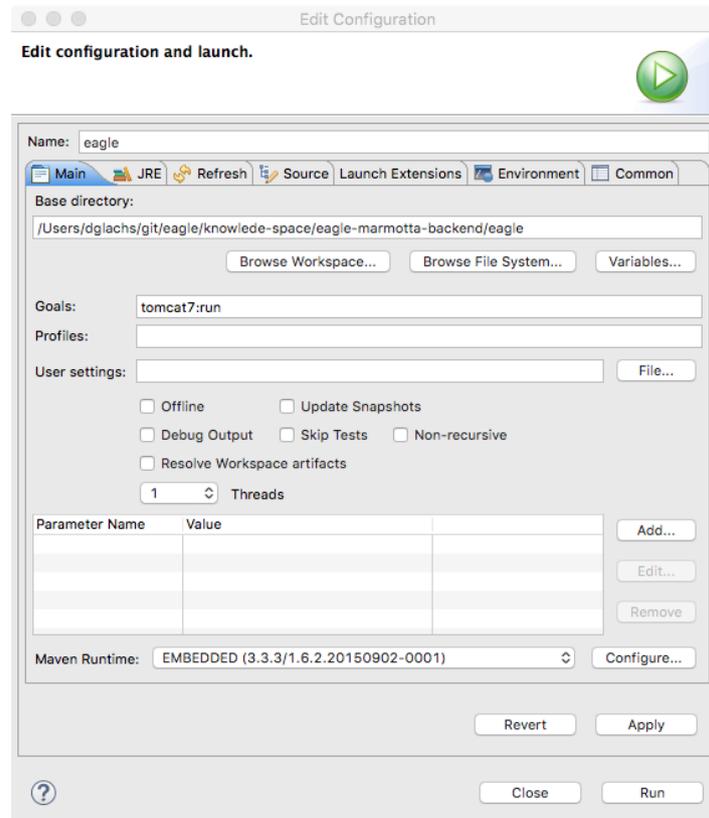


FIGURE 57: RUN CONFIGURATION

Enter “tomcat7:run” in the Goals field and choose “Run”. This will cause Apache Maven to

1. download all other dependencies, including Apache Marmotta (unless already present),
2. download Apache Tomcat as the runtime environment,
3. compile and assemble the EAGLE KB Web Archive ,
4. start Apache Tomcat, and to
5. finally deploy & start the EAGLE Knowledge Base

When executed for the first time, this process might take some time depending on the internet connection when downloading.

To stop the tomcat server, the process in the Console View of Eclipse must be terminated.

Step 4: Add a new functional module to the knowledge base

The actual main functionality is contained in the following modules

- knowledge-model: Holds the basic service to access the object connection required to provide high-level object access rather than low-level “triple access” and also adds the EAGLE Ontology contained in “eu.fp7.eagle.knowledge.core” as a dependency.
- knowledge-space: Provides the service layer and the Web-Interface layer as outlined in Figure 54, in particular for the OER services.
- marmotta-search: Provides the search capabilities and connects with the Apache Marmotta Event Mechanism to get informed on relevant resource updates.
- eagle-skos: Provides the editor for SKOS vocabularies.

When adding a new functional module to the EAGLE KB it is best practice to use the provided Apache Marmotta Archetypes. For this, the “File → New → Other ...” menu option is used. This will show the “Create” Wizard from Eclipse. Choose the option “Maven Module” and click “Next”. Enter the desired module name, and choose the “bundles” project as the parent project.

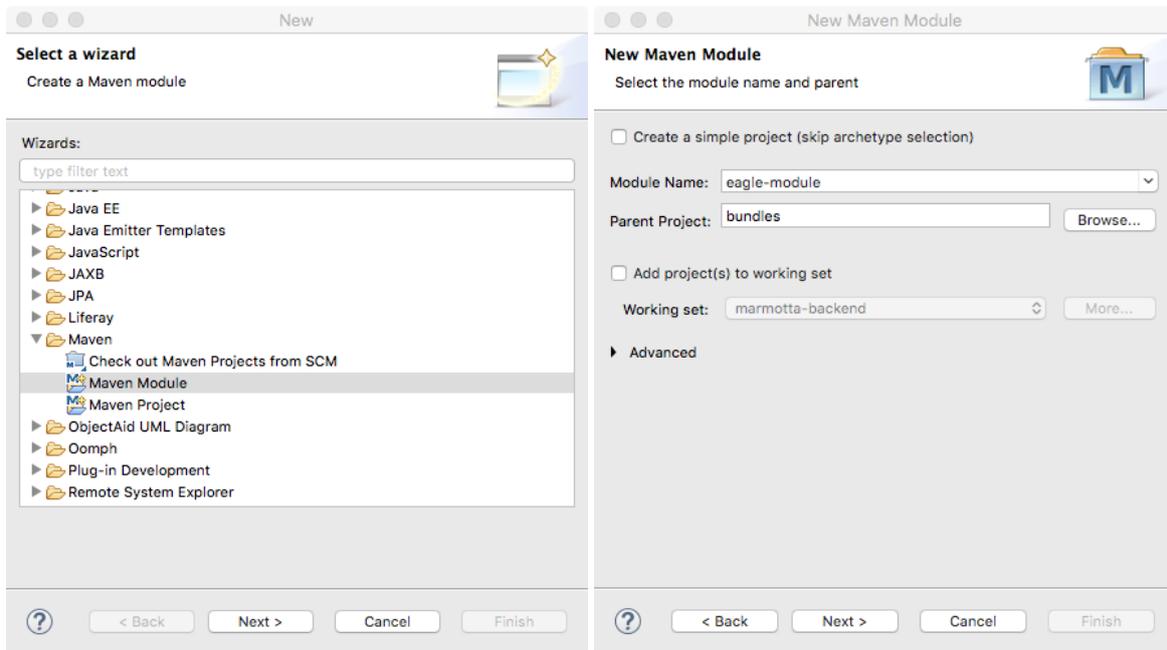


FIGURE 58: EAGLE KNOWLEGE BACKEND MODULE CREATION

Ensure, the “Create a simple project” option in the second page of the wizard is unchecked and proceed to the next page to select the Apache Marmotta Web Module Archetype.

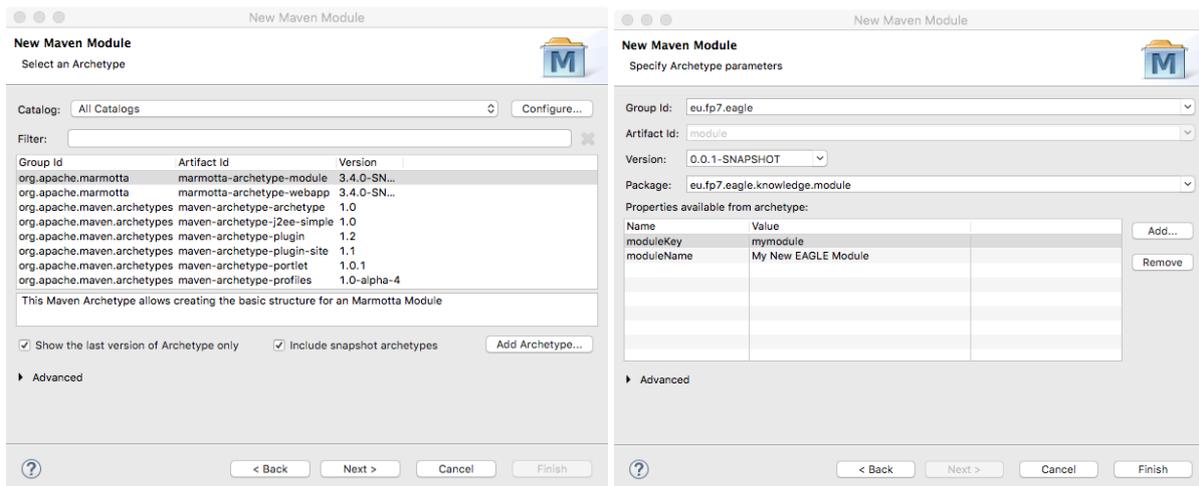


FIGURE 59: EAGLE KNOWLEGE BACKEND MODULE ARCHETYPE SELECTION & CONFIGURATION

The wizard lists all available archetypes. Ensure the “Include snapshot archetypes” option is checked since EAGLE contributed bug fixes and source code to Apache Marmotta and these contributions are available only in the latest snapshot releases. The wizard will show the Apache Marmotta Archetypes and choose the “marmotta-archetype-module” from the list and proceed to the last page where additional parameters must be provided:

- The `moduleKey` denotes the URI part in the context of the EAGLE Knowledge Base. This name must not contain special characters!
- The `moduleName` is used to identify the module in the admin page of the EAGLE Knowledge Base.

Click “Finish” to finalize the module creation wizard. A new project representing the new module is created and shown in the project explorer as shown in Figure 60.

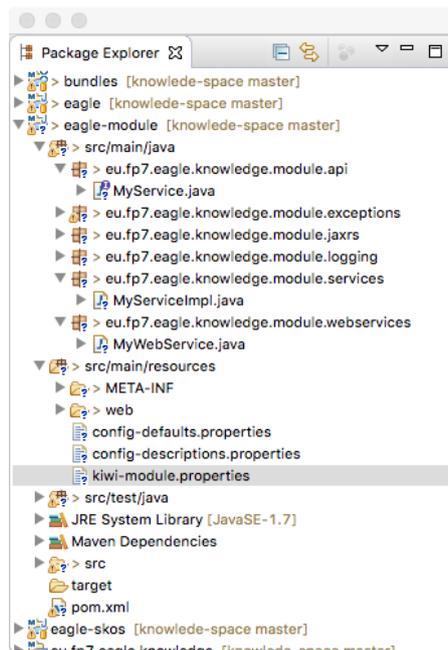


FIGURE 60: EAGLE KNOWLEDGE BACKEND MODULE STRUCTURE

The package structure of the module is divided into three main parts:

- **API:** This package holds interfaces that are used by other modules by means of dependency injection.
- **Service:** This package provides the implementation for the interfaces in the API package.
- **Web Service:** This package publishes RESTful interfaces to the outer world. For processing the requests, the web services will inject the functionality from the API package or from other dependent modules.

This corresponds to the horizontal layers in the architecture overview given in Figure 54.

Step 5: Add the new module as a dependency

The creation wizards of Eclipse automatically add the new module to the structure of the EAGLE KB. This however is not sufficient to have the module deployed along with the existing modules. To enable the new module, it is required to add the dependency to the “eagle” project (the project constituting the web archive). This may be done in the source tree of the `pom.xml` file or by using the graphical editor of the `pom.xml` file.

Step 6: Test the new Service

In order to test the availability of the new service it is required to repackage the EAGLE Knowledge Base. This can be done by right clicking on the “eagle” project and choosing “Run as → Maven install” from the context menu. Upon this, the service may be restarted as outlined in Step 3.

By pointing a browser to the URL <http://localhost:8080/eagle> the administration interface of the EAGLE KB appears. The new module is then listed as shown in Figure 61.

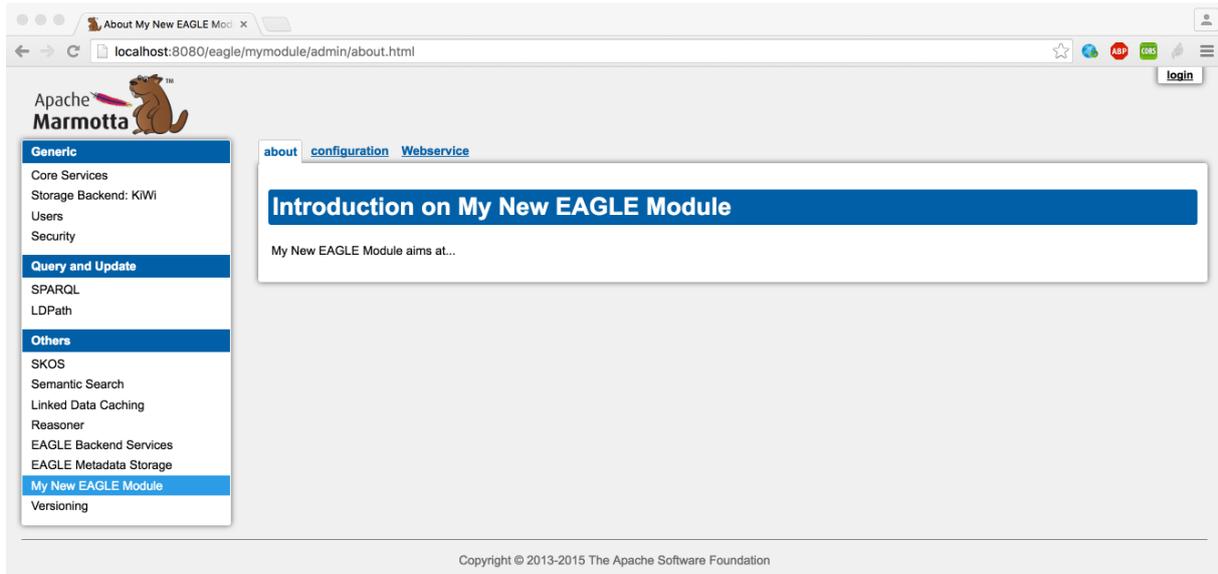


FIGURE 61: ADMIN SECTION OF THE CREATED EAGLE MODULE

5.3 EAGLE AIG Development

All AIG components can be obtained from Git. These can be easily cloned into any local Git repository and modified at the developers’ leisure. All TAO unrelated projects are written in Java 8 while the TAO extensions require PHP as specified above. How to install TAO has also been explained before. Their website also features a community which allows to work on TAO itself but this is out of the scope of this project.

The following are the current repositories to clone:

- <git@git.list.lu:ASSESS/taoAigFacade.git>
- <git@git.list.lu:ASSESS/taoTestGeneration.git>
- <git@git.list.lu:eagle/AIG.git>
- <git@git.list.lu:eagle/SIREN.git>

The instance of TAO we are working on and recommend can also be obtained from <https://git.list.lu/eagle/tao-for-eagle>.

To clone these projects, we recommend to follow the same procedure as outlined for the EAGLE portlet. The AIG and SIREN projects are also Maven projects and should be imported as such to ensure their dependencies are correctly resolved. A custom dependency regarding null annotations can be obtained from git@git.list.lu:DKD/Null_Analysis_Annotations.git or switched to use, for example, Eclipse’s own annotations for null analysis.



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

For working on TAO and the respective modules, we recommend the developer first to a running installation of TAO and then point their IDE to the location of the source files. This ensures that any changes the developer makes can be easily tested on a running instance. Please be sure to not do this on your production environment.

5.4 EAGLE Process Maps and Argumentation Tool Development

The installation and developer guide for the EAGLE Process Maps and Argumentation Tool are documented as part of deliverables D6.9 and D6.10 respectively.



Contract Number
619347

Document Title
EAGLE OER Learning Platform

Deliverable Nature
P
Dissemination level
PU
Version
1.0

6 References

- [1] Wikipedia. (2016) OSGi. [Online]. Available: <http://en.wikipedia.org/wiki/OSGi>
- [2] Vogella. (2014, Dec.) Eclipse Extension Mechanism. [Online]. Available: <http://www.vogella.com/tutorials/EclipseExtensionPoint/article.html>
- [3] L. Vogel. (2012, May) Using dependency injection in java - introduction - tutorial. [Online]. Available: <http://www.vogella.com/tutorials/DependencyInjection/article.html>
- [4] Wikipedia. (2016) Dependency Injection. [Online]. Available: https://en.wikipedia.org/wiki/Dependency_injection
- [5] R. C. M. U. BOB”, “Design principles and design patterns,” Last verified 2009-01-14. [Online]. Available: http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf
- [6] A. Aldazabal, T. Baily, F. Nanclares, A. Sadovykh, C. Hein, M. Esser, and T. Ritter, *Automated Model Driven Development Processes*, ser. ISBN: 978-3-8167-7645-1, in Proceedings of the ECMDA workshop on Model Driven Tool and P. Integration, Eds. Fraunhofer IRB Verlag, Stuttgart, 2008.
- [7] Eclipse. (2016) Remote application platform. [Online]. Available: <http://www.eclipse.org/rap/>
- [8] ——. (2014, Dec.) Tycho. [Online]. Available: <https://eclipse.org/tycho/>

Appendix 1 – Resolving Git Conflicts with EGit

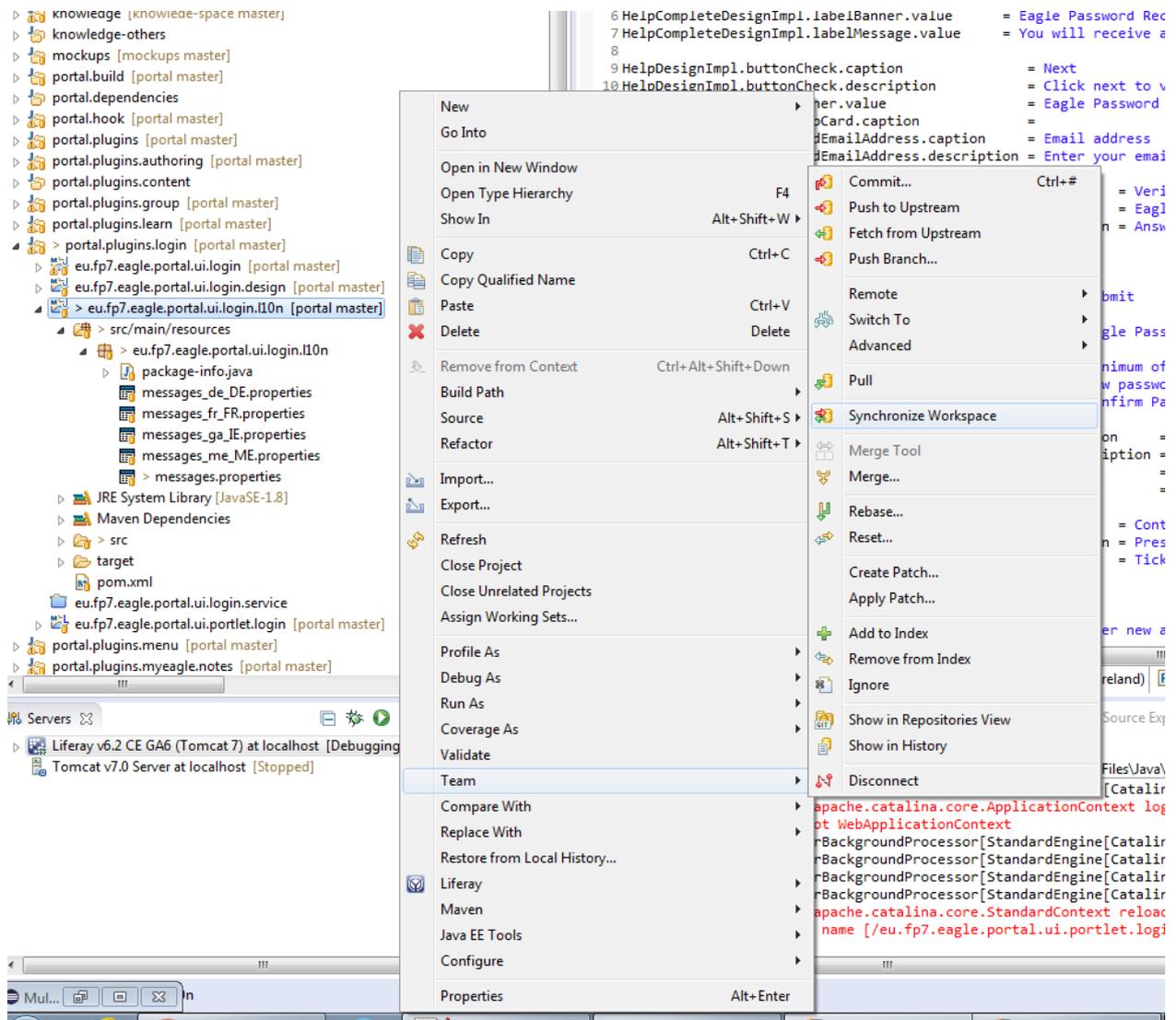


FIGURE 62: ALWAYS START WITH SYNCHRONIZE WORKSPACE

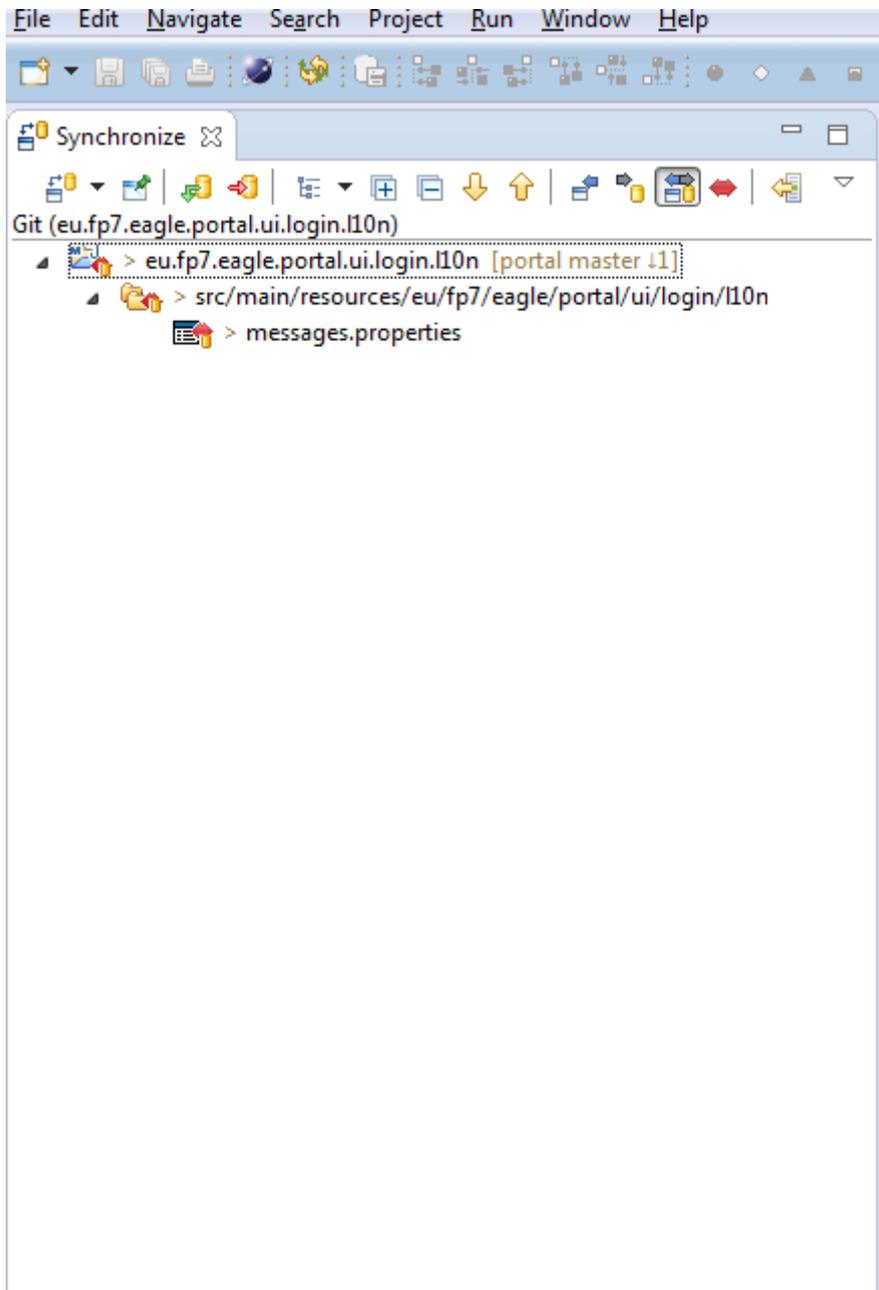


FIGURE 63: IF THERE ARE CONFLICTS, IT WOULD BE SHOWN AS FOLLOWS

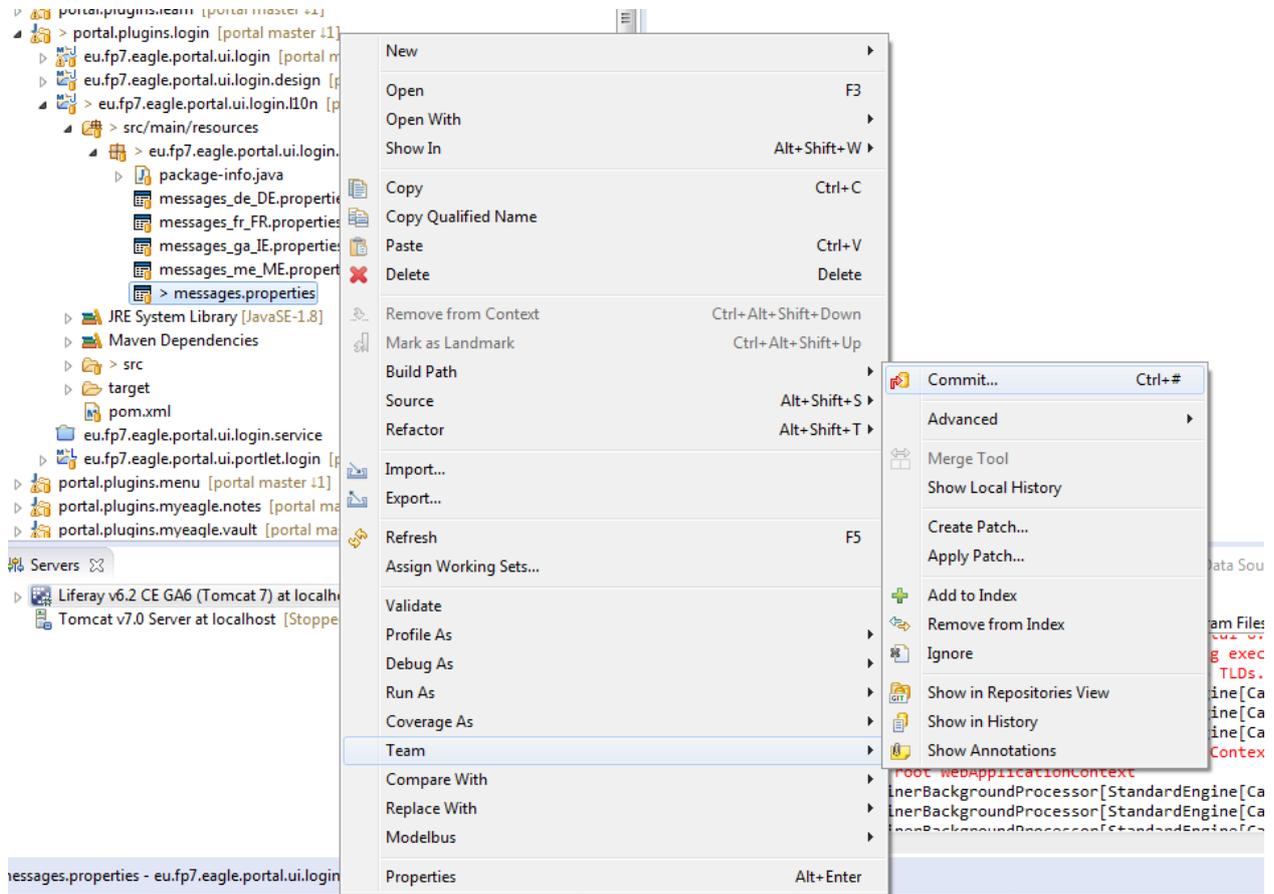


FIGURE 64: GO BACK TO THE CONFLICTING RESOURCE IN THE PACKAGE VIEW, AND CLICK COMMIT

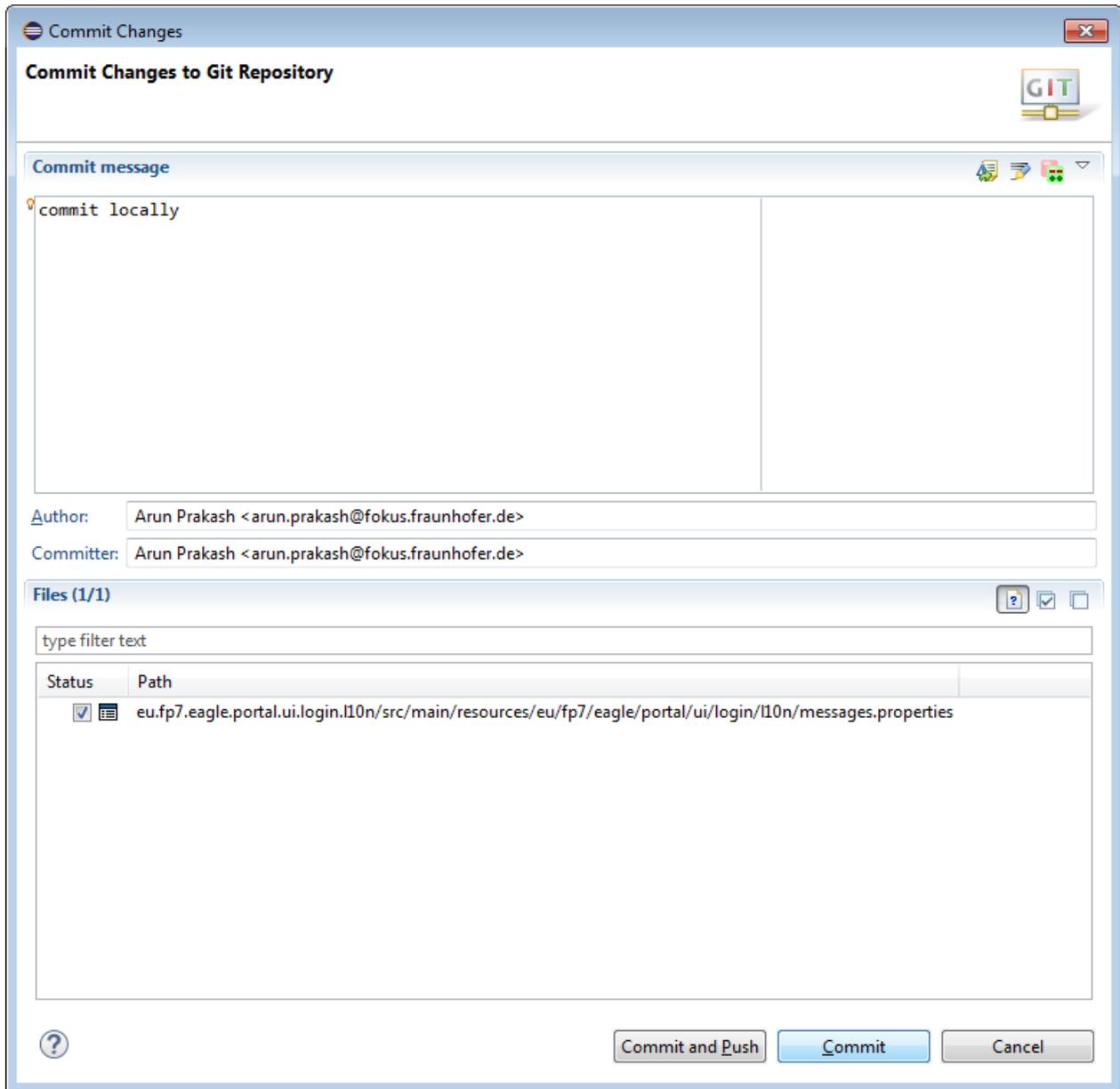


FIGURE 65: COMMIT THE CHANGES LOCALLY

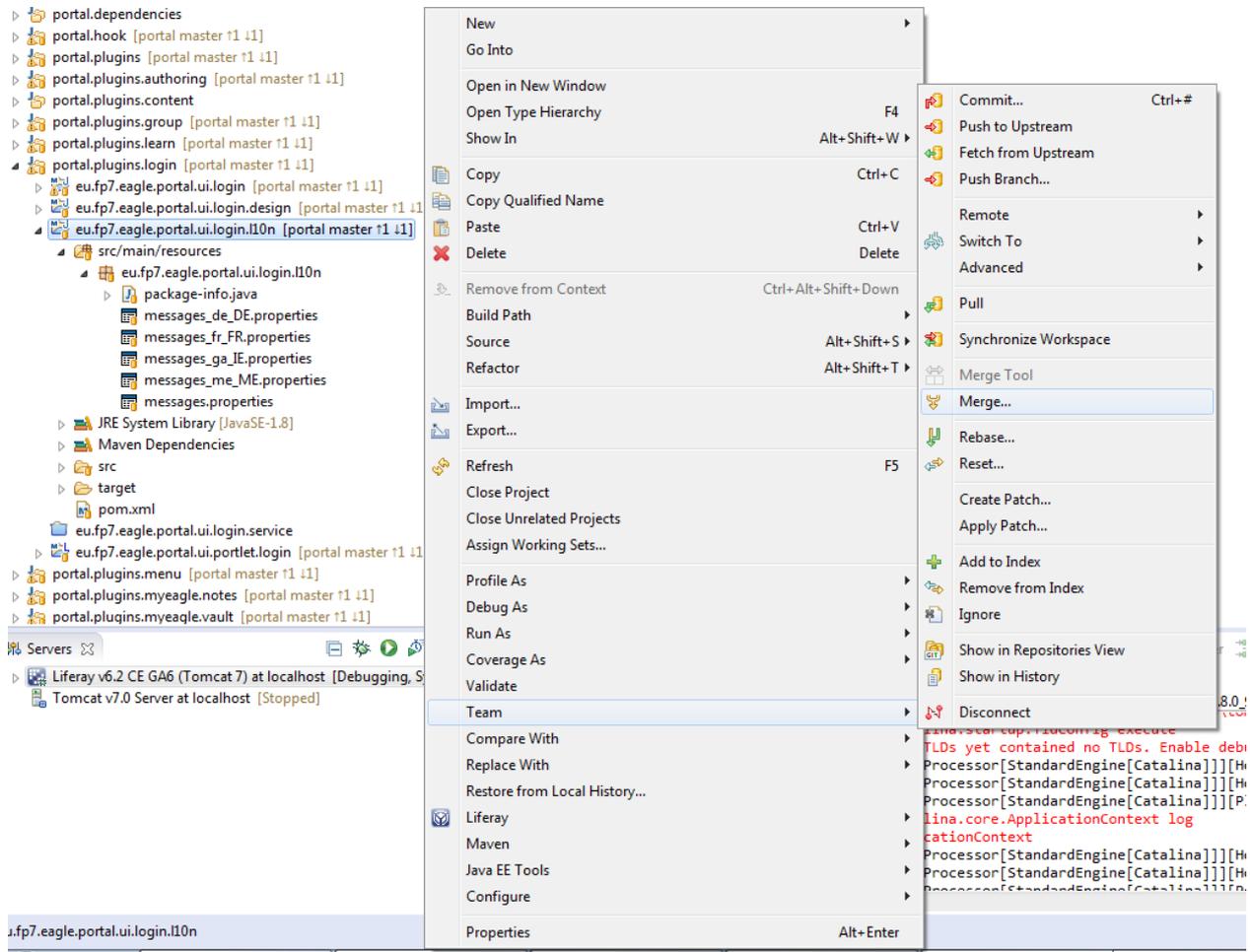


FIGURE 66: CLICK MERGE ON THE PROJECT HOLDING THE CONFLICTING RESOURCE

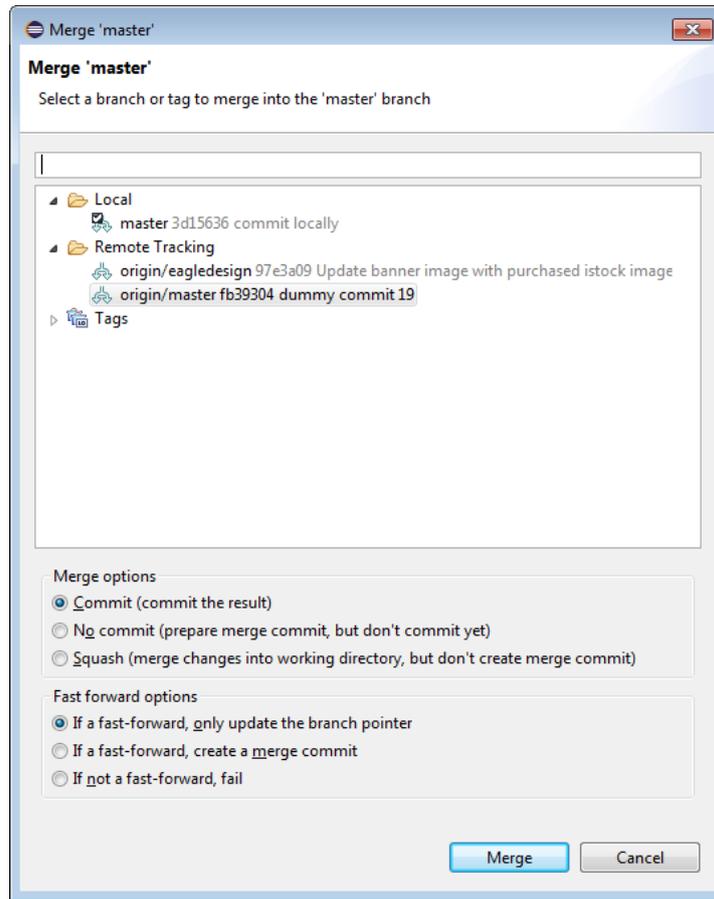


FIGURE 67: CLICK MERGE

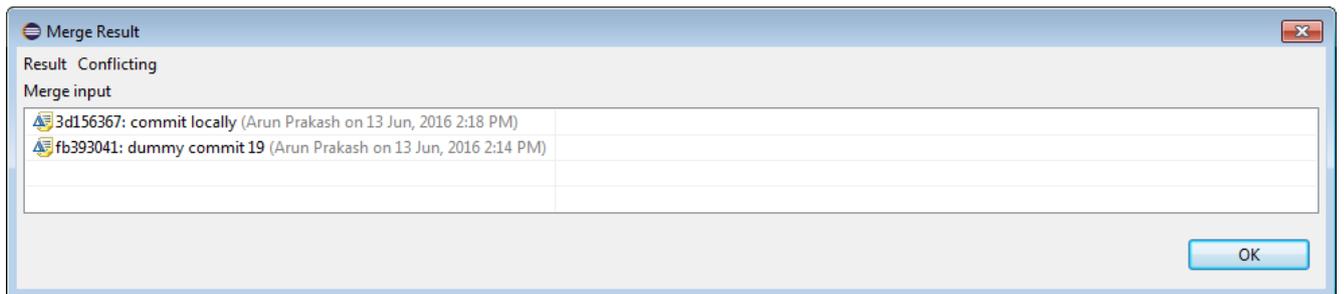


FIGURE 68: MERGE RESULT

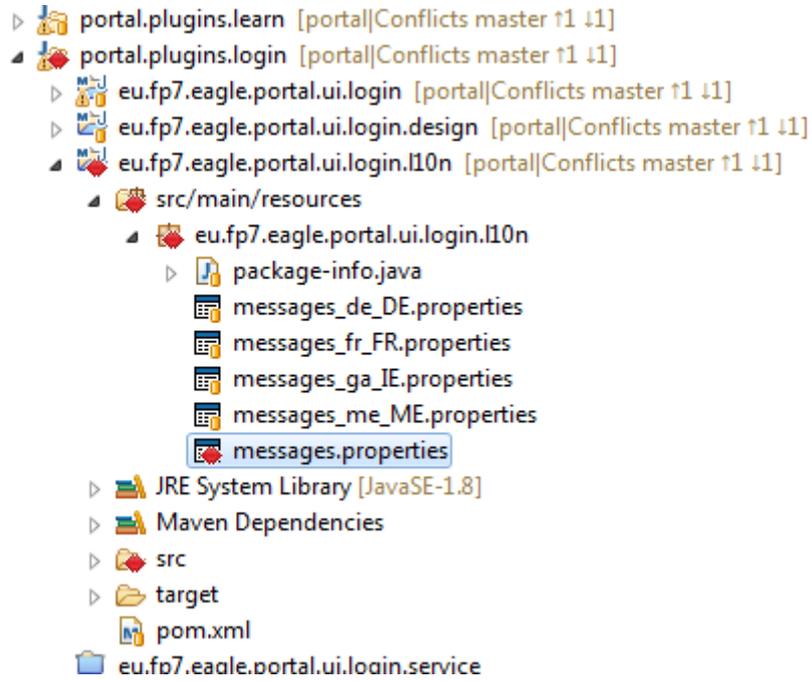


FIGURE 69: YOU WILL BE SHOWN CONFLICTS LIKE THIS

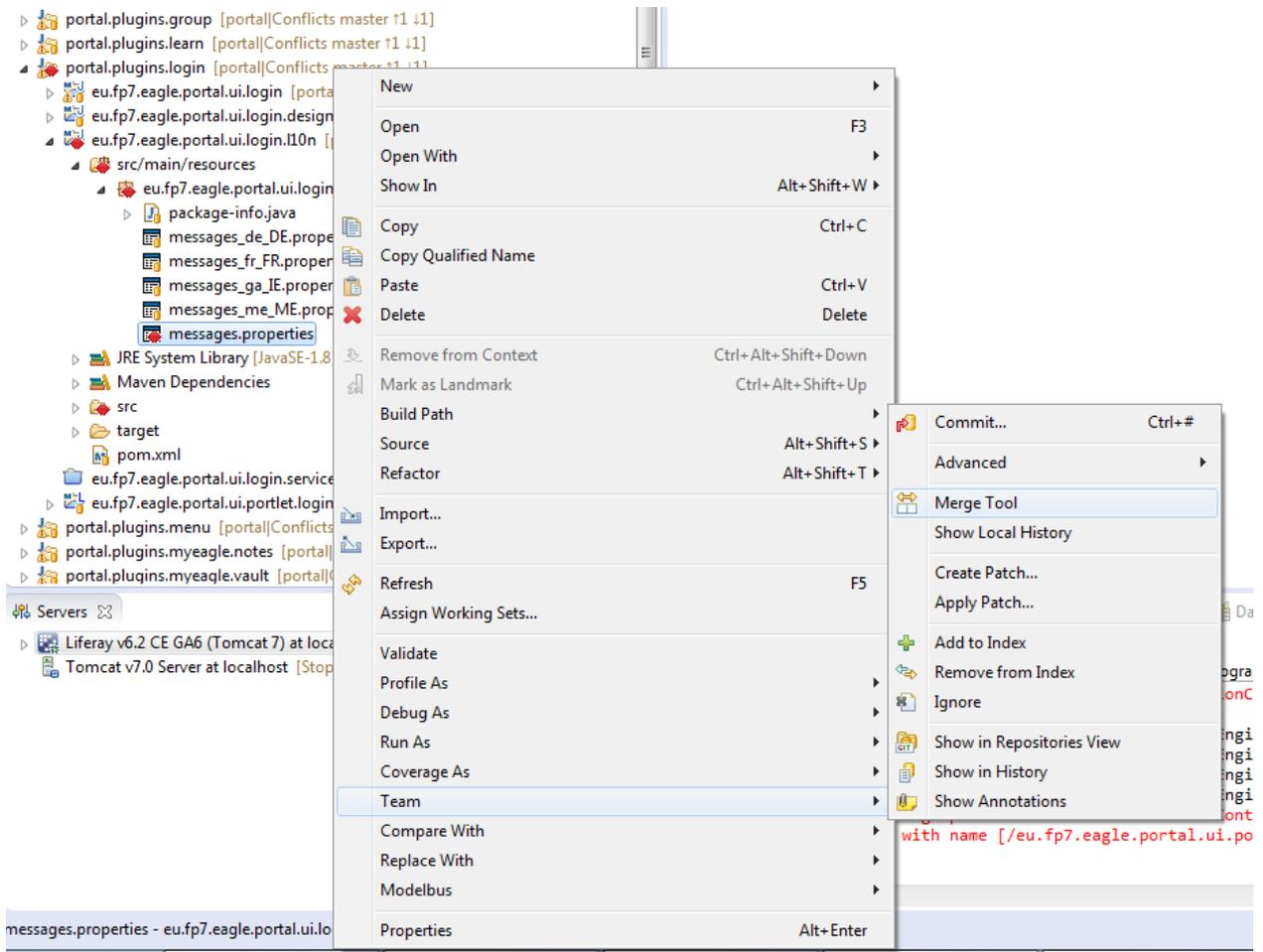


FIGURE 70: OPEN THE MERGE TOOL ON THE CONFLICTING FILE

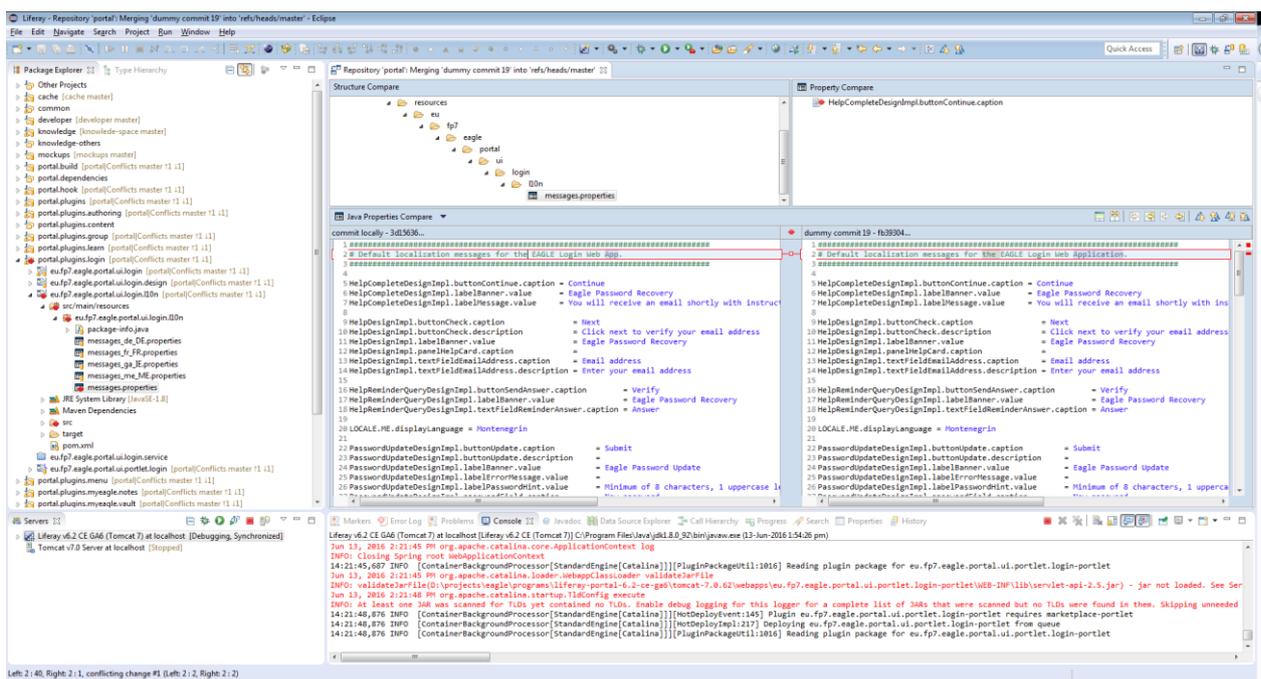


FIGURE 71: MANULLY MERGE THE CONFLICTS

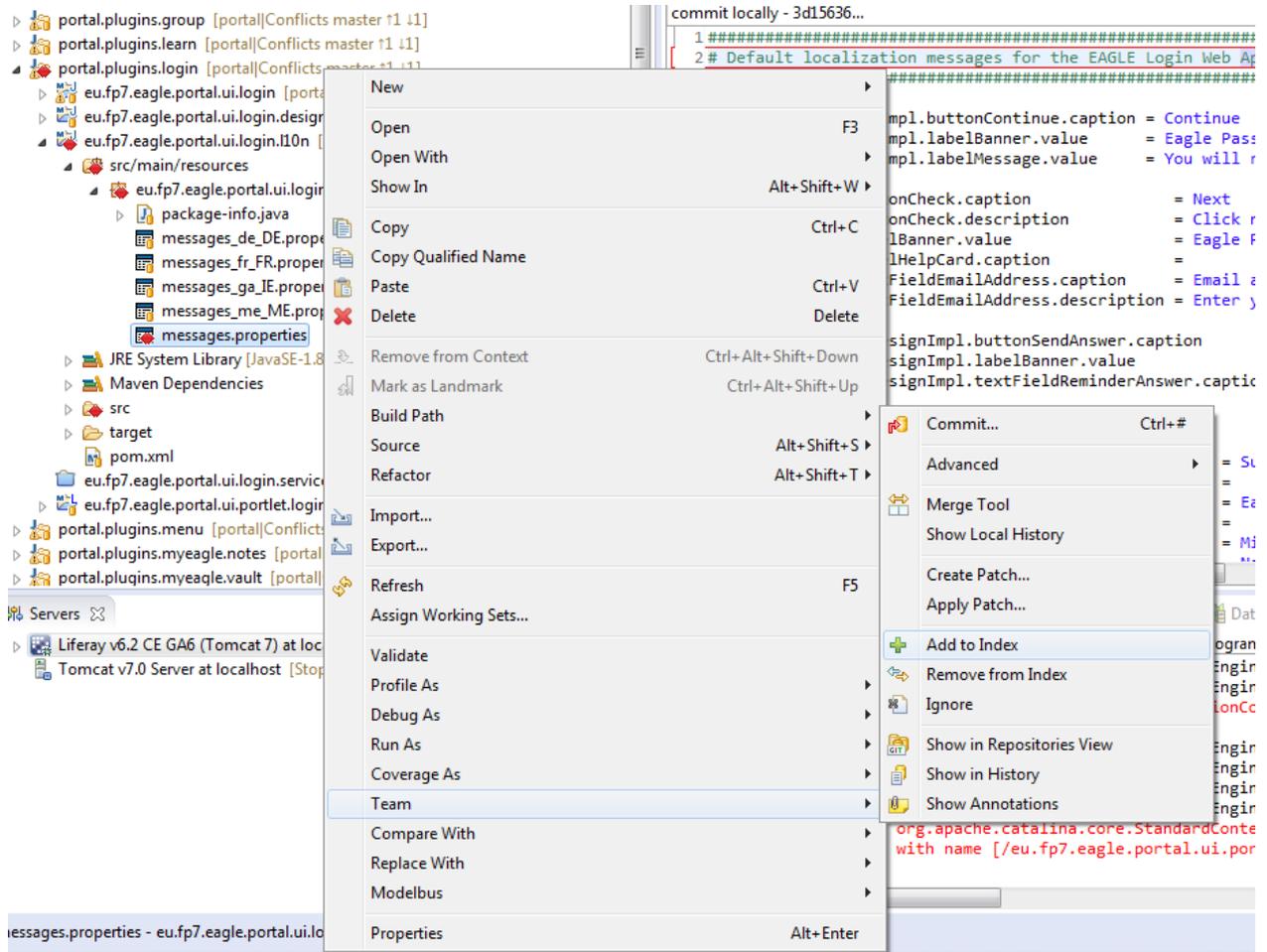


FIGURE 72: AFTER MANUALLY MERGING, ADD THE FILE TO THE INDEX



Contract Number
619347

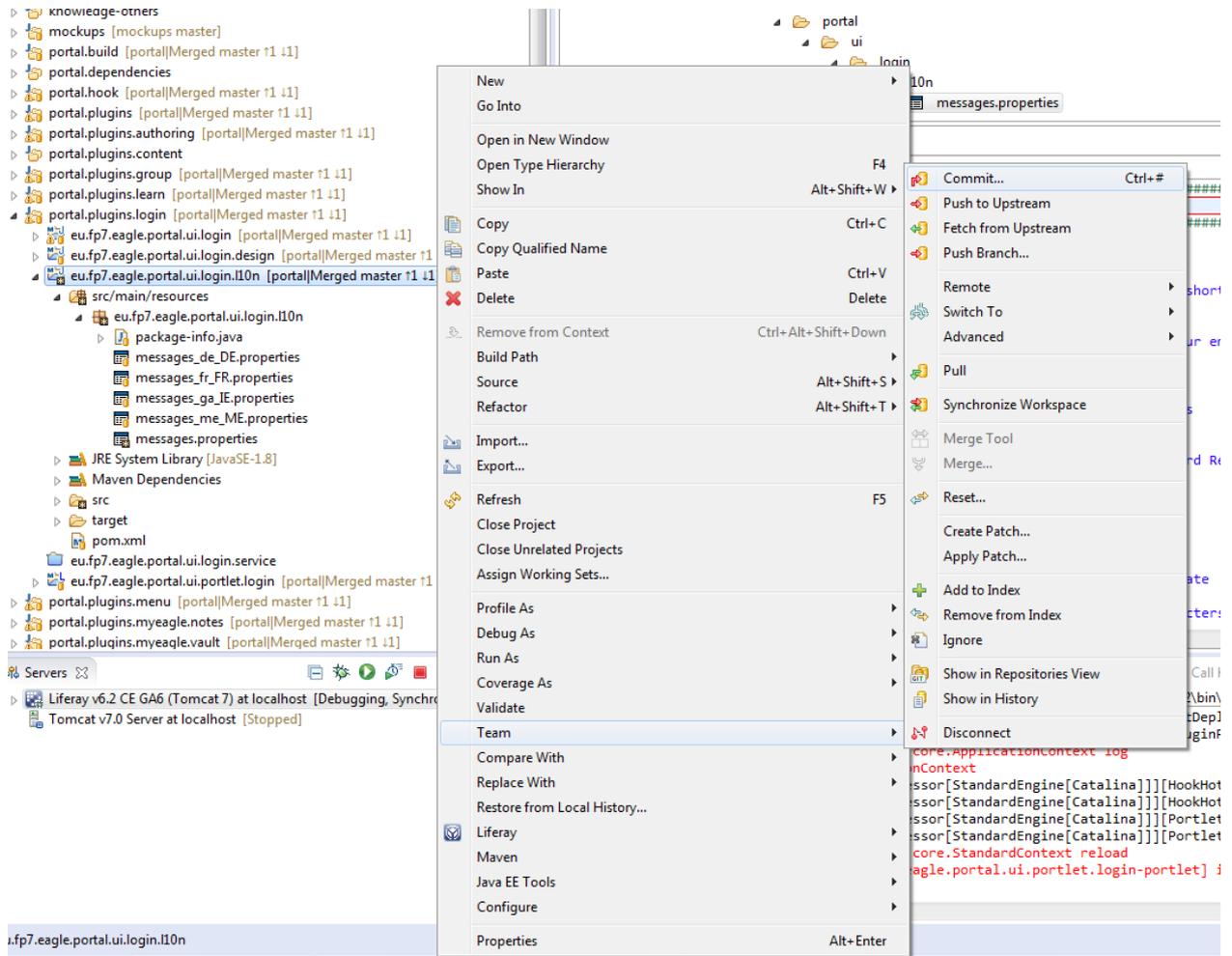


FIGURE 73: FINALLY COMMIT THE RESOURCE

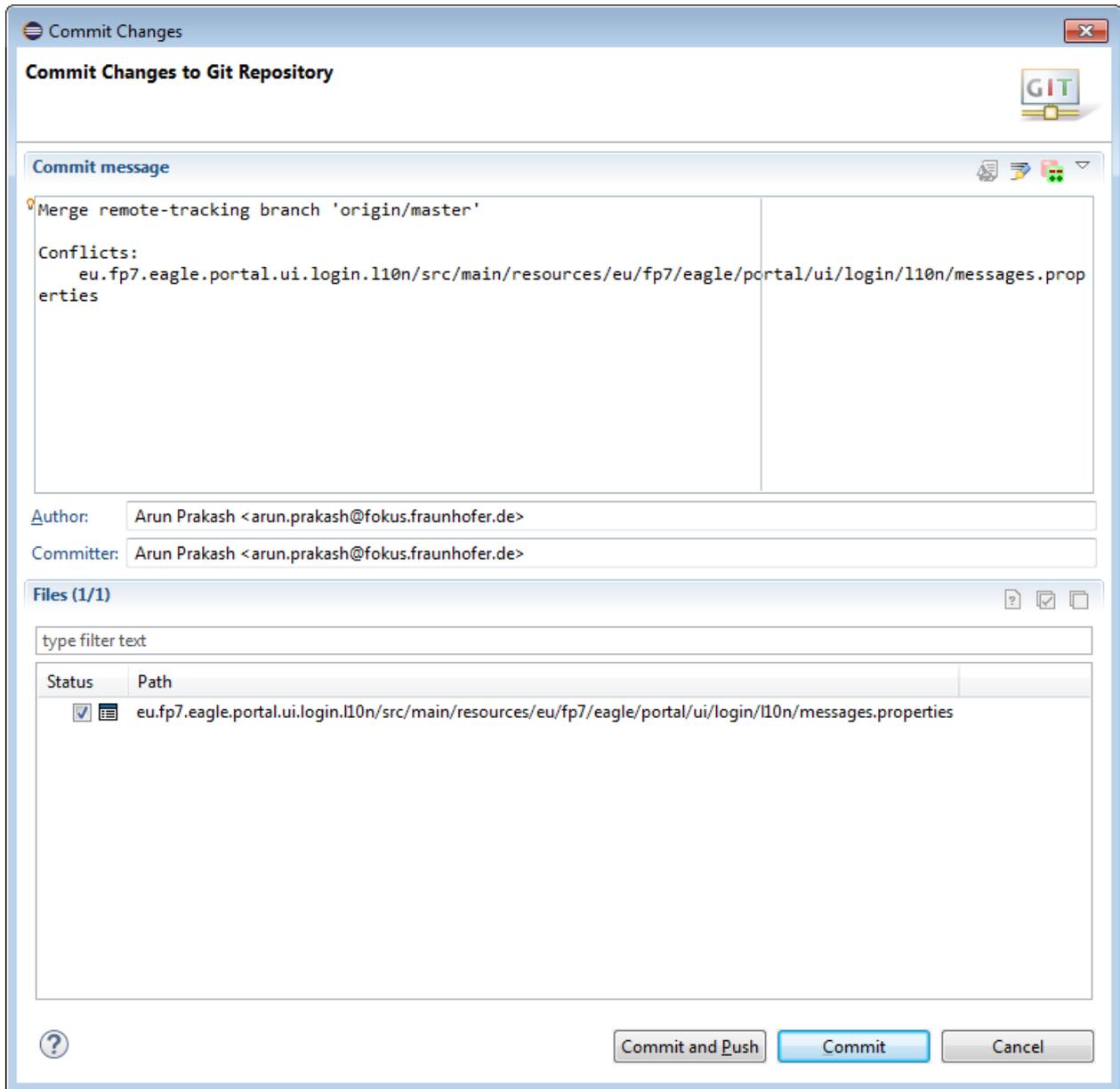


FIGURE 74: COMMIT AND PUSH TO REMOTE REPOSITORY



Contract Number
619347

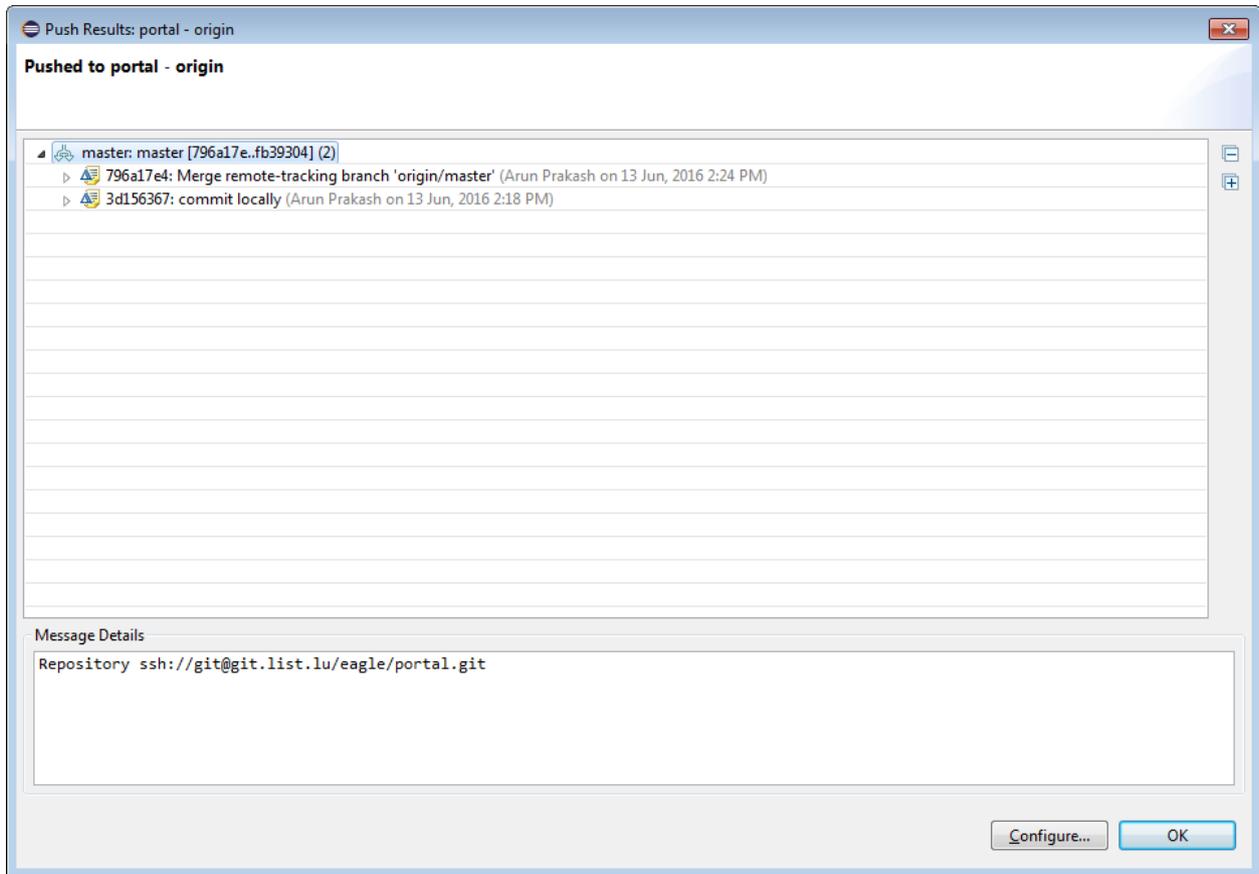


FIGURE 75: CONFLICT RESOLUTION END RESULT