KB-DEV-001: Opening a NetBeans project in Eclipse from Git

Introduction

When users collaborate on code, they often use Git as versioning system. The problem, however, is that users use their preferred IDE. This can cause many difficulties, especially if the IDE does not support the import of the other IDE.
This article focusses on one case: opening a simple JSP NetBeans project in Eclipse as a Dynamic Web Project with Git as versioning system. We suppose the Git repository is already cloned into a folder.

Contents

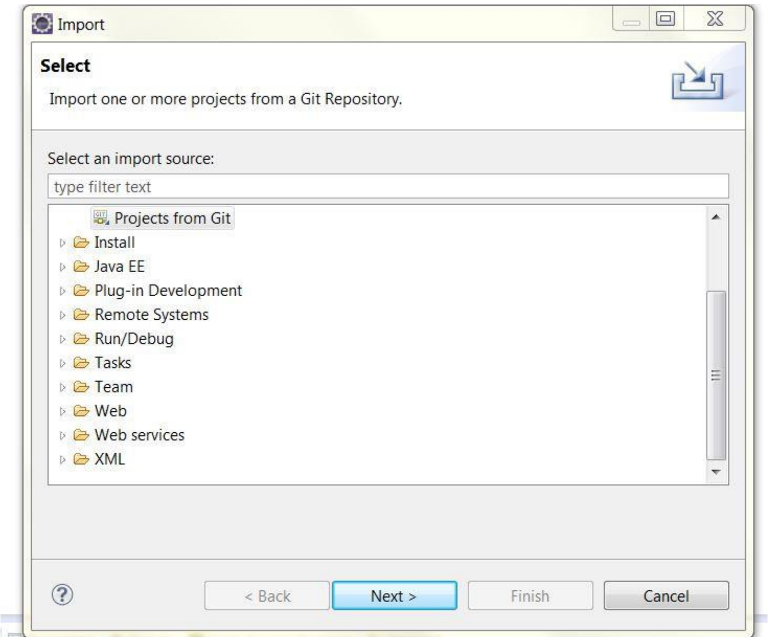
Prerequisites

- Eclipse IDE
- EGit plugin for Eclipse

Importing the project

The project can be imported using the default import option. To do so, right-click in the **Navigation panel**. Then click **Import...** and again **Import...** Afterwards search for the Git option, an example of the import window can be found in Figure 1. The **Git** option can be found within the **Git** folder.

Figure 1 The import window



Import

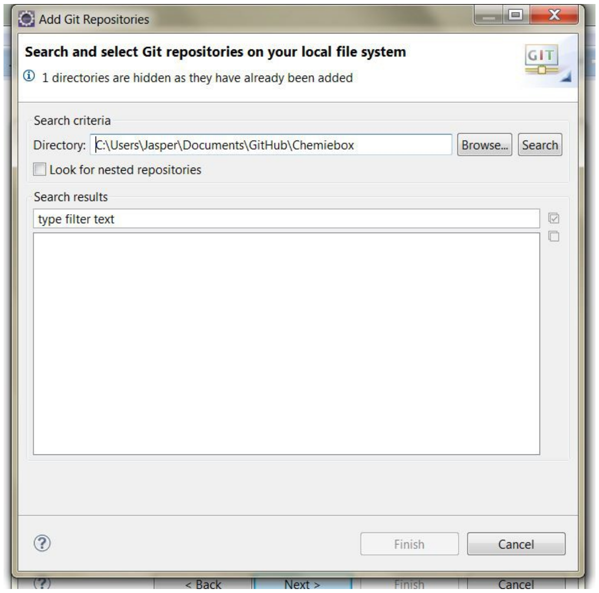
Select

Import one or more projects from a Git Repository.

Select an import source:

Next, the IDE asks to select a repository. As this article supposes the repository is already cloned, choose the **Existing local repository** option. Afterwards the **Search and select Git repositories on your file system** window lets you locate the Git repository in your file system. If everything went well , select the repository you want to use. In the case of this article this is the repository "Chemiebox", as seen in Figure 3.

Figure 2 Search and select Git repositories on your local file system



Add Git Repositories

Search and select Git repositories on your local file system

1 directories are hidden as they have already been added

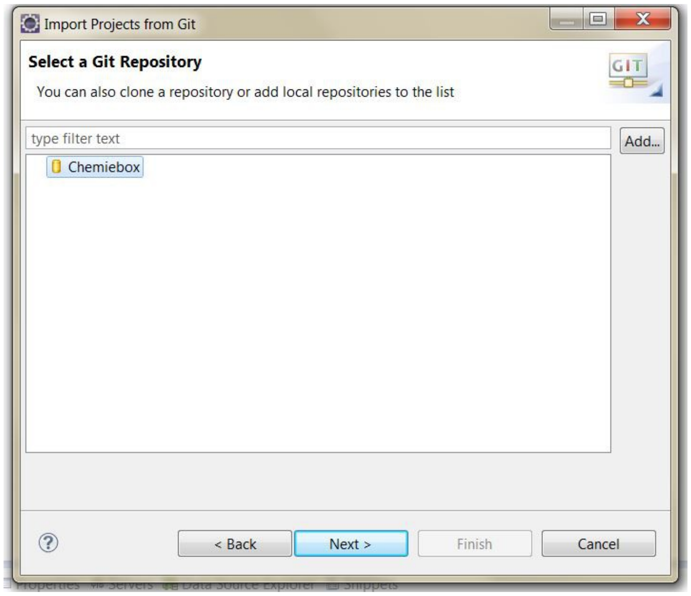
Search criteria

Directory: C:\Users\Jasper\Documents\GitHub\Chemiebox

□ Look for nested repositories

Search results

Figure 3 The local repository


Import Projects from Git

Select a Git Repository

You can also clone a repository or add local repositories to the list

type filter text

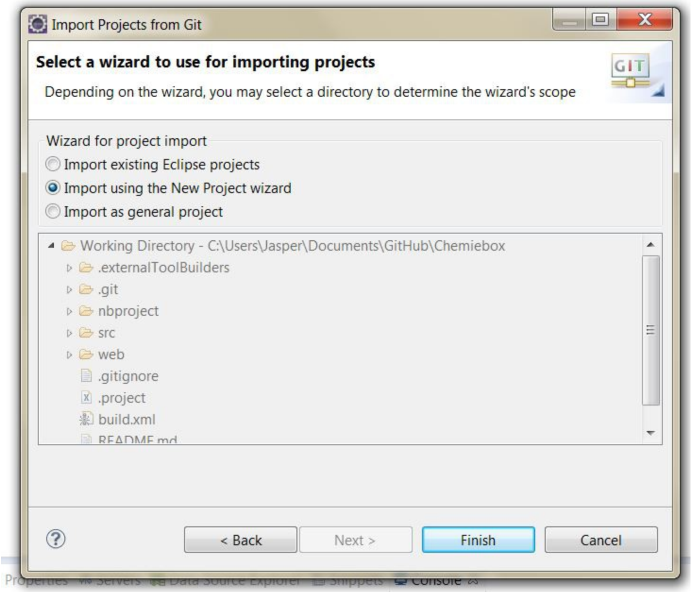
Chemiebox

Again, click **Next** to go to the next screen, where you will create the project.

Create the project from the repository

After selecting the project, a new project needs to be created. Choose the **Import using the New Project wizard** option before clicking on Finish. You need this option, see Figure 4, as the first option is invalid because it is a NetBeans project and the third option will not recognize it as a Dynamic Web Project.

Figure 4 Creating the project from the source


Import Projects from Git

Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope

Wizard for project import

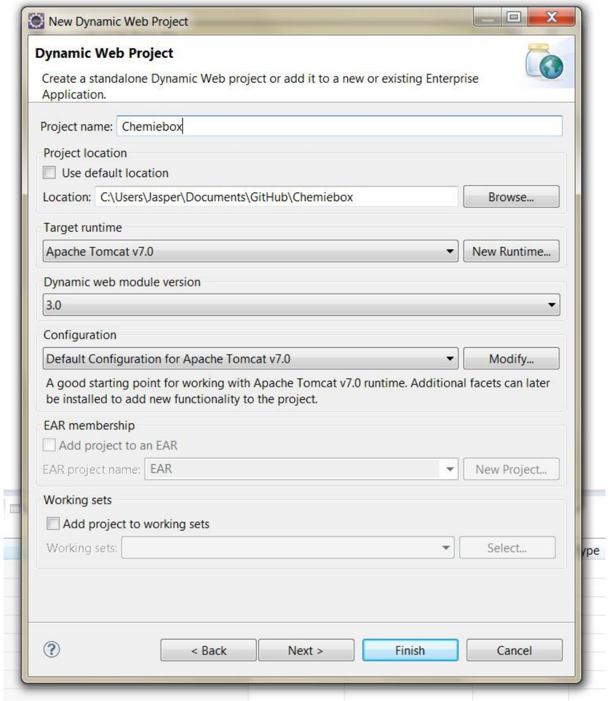
○ Import existing Eclipse projects
● Import using the New Project wizard
○ Import as general project

```
Working Directory - C:\Users\Jasper\Documents\GitHub\Chemiebox
    .externalToolBuilders
    .git
    nbproject
    src
    web
    .gitignore
    .project
    build.xml
    README.md
```

After clicking **Finish**, the **New Project** wizard will open. Choose the **Dynamic Web Project**, which is located in the **Web** folder. Click **Next** and give the project a name.

Important: to use the git functionality, uncheck the **Use default location** option in the **Project location** section. The location of the project should be within the repository itself. This is illustrated in Figure 5.

Figure 5 Project location



New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: Chemiebox

Project location

□ Use default location

Location: C:\Users\Jasper\Documents\GitHub\Chemiebox

Target runtime

Apache Tomcat v7.0

Dynamic web module version

3.0

Configuration

Default Configuration for Apache Tomcat v7.0

A good starting point for working with Apache Tomcat v7.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

□ Add project to an EAR

EAR project name: EAR

Working sets

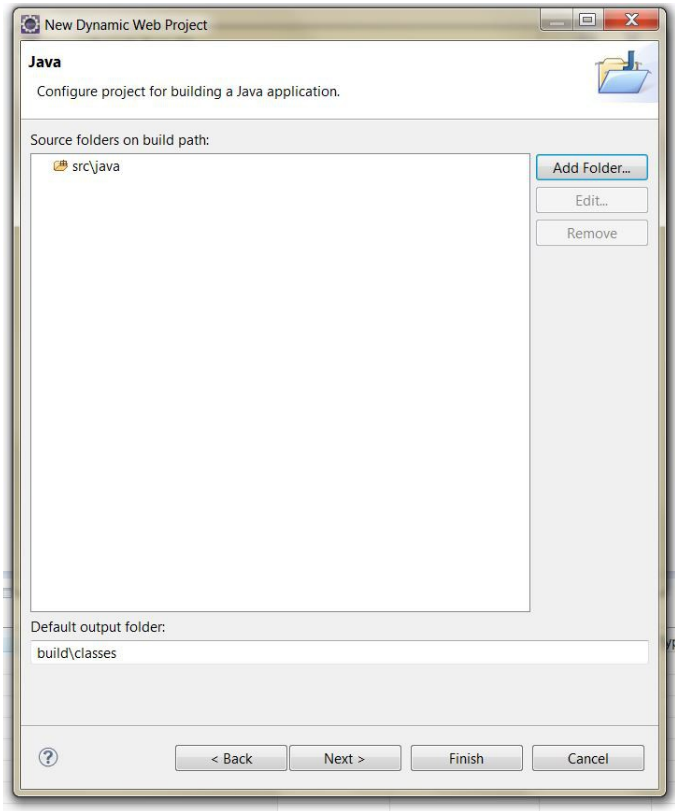
□ Add project to working sets

Working sets:

Select the desired **Target runtime** and click **Next**. If you do not have a valid **Target runtime** and do not know how to create it, consider reading **KB-DEV-002**.

The next step is configuring the build path. By default, this is the *src* folder. If we would continue using the default setting, we would run into many errors to solve. A better approach is to change the build path to *src/java*, as NetBeans uses this as well. Leave the default output folder to its default, *build\classes*. This configuration is also showed in Figure 6.

Figure 6 Configure build path



New Dynamic Web Project

Java

Configure project for building a Java application.

Source folders on build path:

src\java

Default output folder:

build\classes

The last step before clicking **Finish** is to adapt the Content directory, click **Next** to do this. Change the **Content directory** to *web*. Now click **Finish**.

Adding the missing libraries

After the creation of the project you will might run into many errors. This is due to missing libraries. As it is bad practice to add libraries to your Git repositories, you will have to re-add them manually in Eclipse.$

To do so, simply add the in the **WEB-INF/lib** folder within the **web** folder. Within some seconds you will see them appear under **Web App Libraries**, this is visualized in Figure 7 and 8.

Figure 7 The lib folder

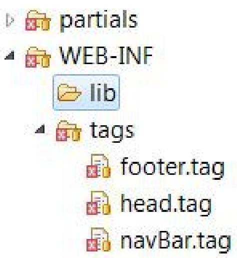
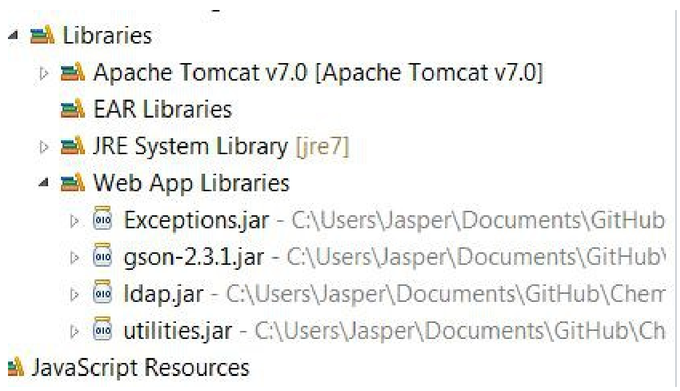


Figure 8 Web App Libraries



```
◢ Libraries
    ▷ Apache Tomcat v7.0 [Apache Tomcat v7.0]
      EAR Libraries
    ▷ JRE System Library [jre7]
    ◢ Web App Libraries
        ▷ Exceptions.jar - C:\Users\Jasper\Documents\GitHub
        ▷ gson-2.3.1.jar - C:\Users\Jasper\Documents\GitHub\
        ▷ ldap.jar - C:\Users\Jasper\Documents\GitHub\Chem
        ▷ utilities.jar - C:\Users\Jasper\Documents\GitHub\Ch
JavaScript Resources
```