Install and config of development environment

1. Development Environment

There are several commonly known development environments available for the Java language which also support Java EE features. Their functionality can usually be extended with the installation of plugins.

- *NetBeans* - open source and free, but with receding support from the Java EE community. There are less available plugins for NetBeans than for the following two environments.
- *IntelliJ IDEA* - very popular nowadays in the Java EE community. The full IntelliJ pack can only be used with a commercial licence or with a free student licence. There is also a Community version available but without a lot of essential features needed for Java EE development (eg server support).
- *Eclipse IDE for Java EE Developers (with Oxygen as the current version)* - it is also freely downloadable from the internet with a wide support and lots of available plugins.

1.1 Installation at home

- Download from the following site:

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/oxygen/R/eclipse-jee-oxygen-R-win32-x86_64.zip

- Eclipse does not need installation, simply unzip it to a place of your choice and run eclipse.exe.

1.2 Classroom usage

- Eclipse has been preinstalled for you in the folder c:/javaee
- Find the folder and run eclipse.exe

After starting Eclipse it will ask you for a *workspace* location. A workspace is a unit of work which collects projects and config settings (eg which jdk to use). For development at home we advise you to use a separate workspace for each project. You can create a new workspace by copying an old one and deleting the projects in it keeping only the settings.

2. Installation of the application server

There are several application servers available which support Java EE 7. Some of them only support a subset of the Java EE specification, the web part (for example Apache Tomcat or Jetty). We will also be working with *Enterprise JavaBeans (EJB)* during the course so we will be needing a server which supports the full stack. Glassfish is the official Oracle implementation of the server specification and is useful for development purposes but there are some problems with the Eclipse integration so our application server of choice will be

Wildfly (formerly JBoss). This application server integrates well with Eclipse. It is also widely used in production as well as development.

2.1 Installation at home

- Start Eclipse with the workspace you want to work with
- Open *Help -> Eclipse Marketplace…* menu and search for JBoss Tools plugin (in our case it is **JBoss Tools 4.5.0.Final**)
- Install the plugin and restart Eclipse
- Choose the Servers tab in the lower part of the screen and click on *„No servers are available. Click this link to create a new server…"*
- Choose *Wildfly 10.0 Runtime* and provide localhost as hostname, server name can be chosen freely
- In the next tab leave the default values (server local, file management)
- On the next page click on *„Download and install runtime…"* and click through the dialogues and download the server to a place of your choice
- If in *Execution Environment* only JRE is listed you also have to add a JDK: click on the *Installed JREs…* bottom and find the JDK installed on your machine and add it

- Click *Finish* and you will see a new Server on the Servers tab
- Right click on your new server and choose start
- In Launch Configurations add -Dfile.encoding=UTF-8 to VM arguments. This will set the file encoding to UTF-8
- After starting your server choose http://localhost:9990/console to check whether it has really started

2.2 Classroom installation

- You will need to follow the steps above with one difference: in step *„Download and install runtime…"* do not download the server but find it on the local machine (c:/javaee)
- You will also need to change the default 8080 port the server listens on as this is used by another application
- On the Servers tab please find the server config xml (standalone.xml)
- *Wildfly 10.0 szerver -> Filesets -> Configuration File -> standalone.xml.*
- In the config file: *server->socket-binding-group->port-offset ${jboss.socket.binding.port-offset:0}* change it to *${jboss.socket.binding.port-offset:5}*
- You will find all the ports offsetted by 5, so instead of port 8080 you will have to access port 8085

3. DB config

3.1 Accessing the DB through Eclipse

We want to access a remote DB through Eclipse:

- Clicking on the *Data Source Explorer* tab below we will find the *Database Connections* part empty
- Right click on the label and select *New…*
- Database type is *PostgreSQL*, name can be anything
- On the next page, we will find that there is no driver specified yet
- Click on the bottom next to the list:
    - On the *Name/Type* tab choose *PostgreSQL JDBC Driver*

- In *JAR List* delete the entry and provide the driver you received from the course instructor (*postgresql-9.4-1201.jdbc41.jar*)
- On the properties tab fill in the following (leave the other entries)
  - Connection URL: jdbc:postgresql://hj14.modit.hu:25432/hj14
  - Password: Rooh8oophiej
  - User name: hj14

- On clicking on *Test Connection* we can check our connection
- If everything is OK click on *Finish*

3.2 Configuring DB Access in the application Server

To be able to access our DB through the application server we have to configure a DataSource in the application server.

- Copy the *postgresql-9.4-1201.jdbc41.jar* driver file and a modules.xml file given to you by the course instructor to *modules\system\layers\base\org\postgresql\main* in the application server subfolder.
- Edit the standalone.xml file: *Wildfly 10.0 szerver -> Filesets -> Configuration File -> standalone.xml*
- Under the drivers tag add the org.postgresql driver module added above (datasources subsystem)

```
<drivers>
        <driver name="h2" module="com.h2database.h2">
         <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
         </driver>
         <driver name="postgres" module="org.postgresql">
              <driver-class>org.postgresql.Driver</driver-class>
         </driver>
</drivers>
```

- You also have to add the dataosurce in the standalone.xml file
  Servers tab -> Wildfly 10.0 szerver -> Filesets -> Configuration File -> standalone.xml

```
<subsystem xmlns="urn:jboss:domain:datasources:3.0">

    <datasources>

<datasource jndi-name="java:/laborDS" pool-name="laborDS" enabled="true" use-java-context="true">

      <connection-url>jdbc:postgresql://hj14.modit.hu:25432/hj14</connection-url>

            <driver>postgres</driver>

            <security>

                <user-name>hj14</user-name>

                <password>Rooh8oophiej</password>

            </security>

        </datasource>

        <drivers>

            <driver name="postgres" module="org.postgresql">

                <driver-class>org.postgresql.Driver</driver-class>
```

```
            </driver>
        </drivers>
    </datasources>
</subsystem>
```