

# MENTAL HEALTH LIBERIA Developer Documentation

## 1. Document Description

This Document is designed to do the following:

- Explain how to set up recommended IDE/Programs for rapid development
- Explain how to run codes in development mode
- Explain the function of given directory/file.

It is the developer's job to do the following:

- Familiarizing him/herself with programming languages such as Java, SQL, Groovy, etc.
- Installing and configuring IDE/Programs

The “short version” only contains the essential information – this applies globally, regardless of the OS. The “long version” contains step-by-step directions for developers with less experience. “Long version” assumes that you are running Windows XP/Vista/7.

## 2. Running the server code:

### 2.1 Prerequisite:

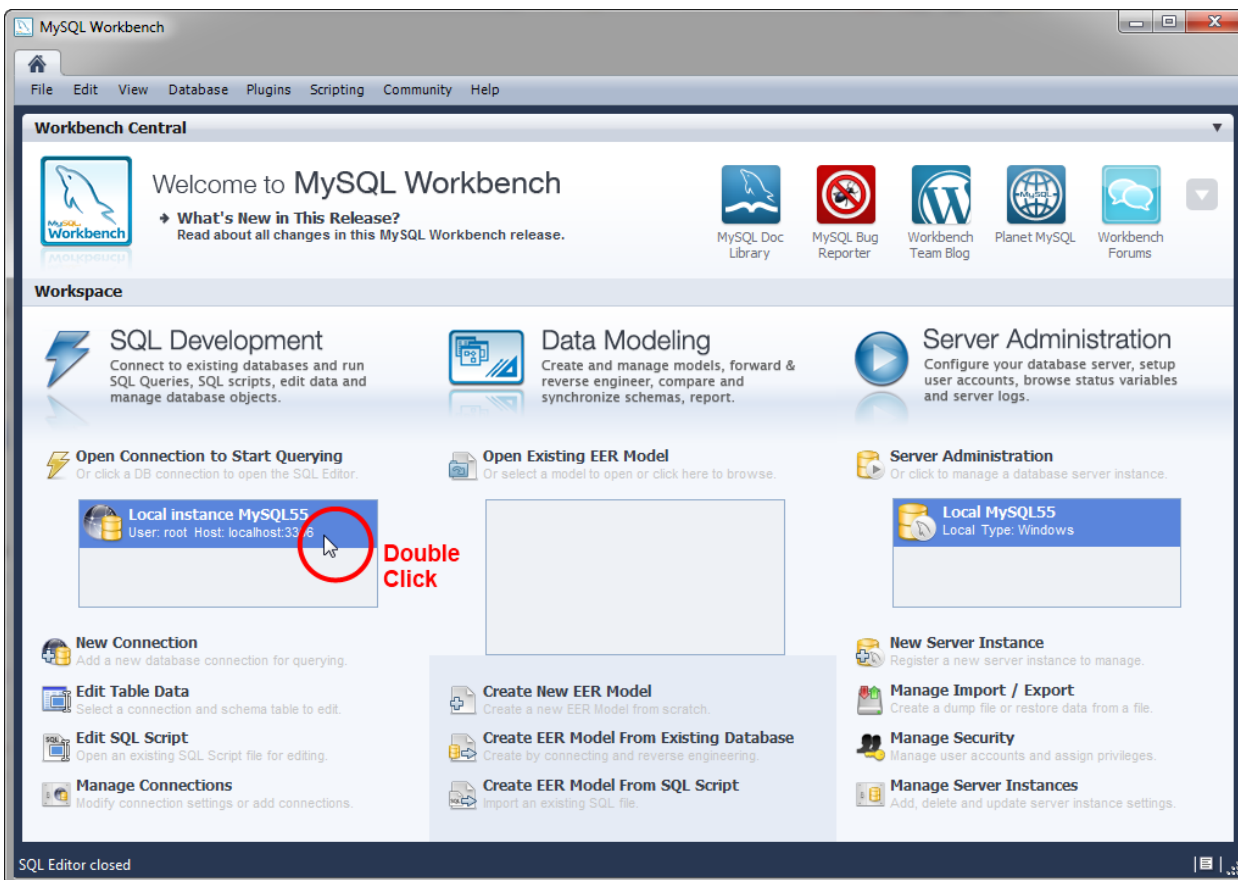
1. You must have the latest JDK (Java Development Kit) installed on your system. You can download it from <http://www.oracle.com>. If you do not have JDK on your system, please download and install it. For more information, please refer to <http://www.oracle.com>.

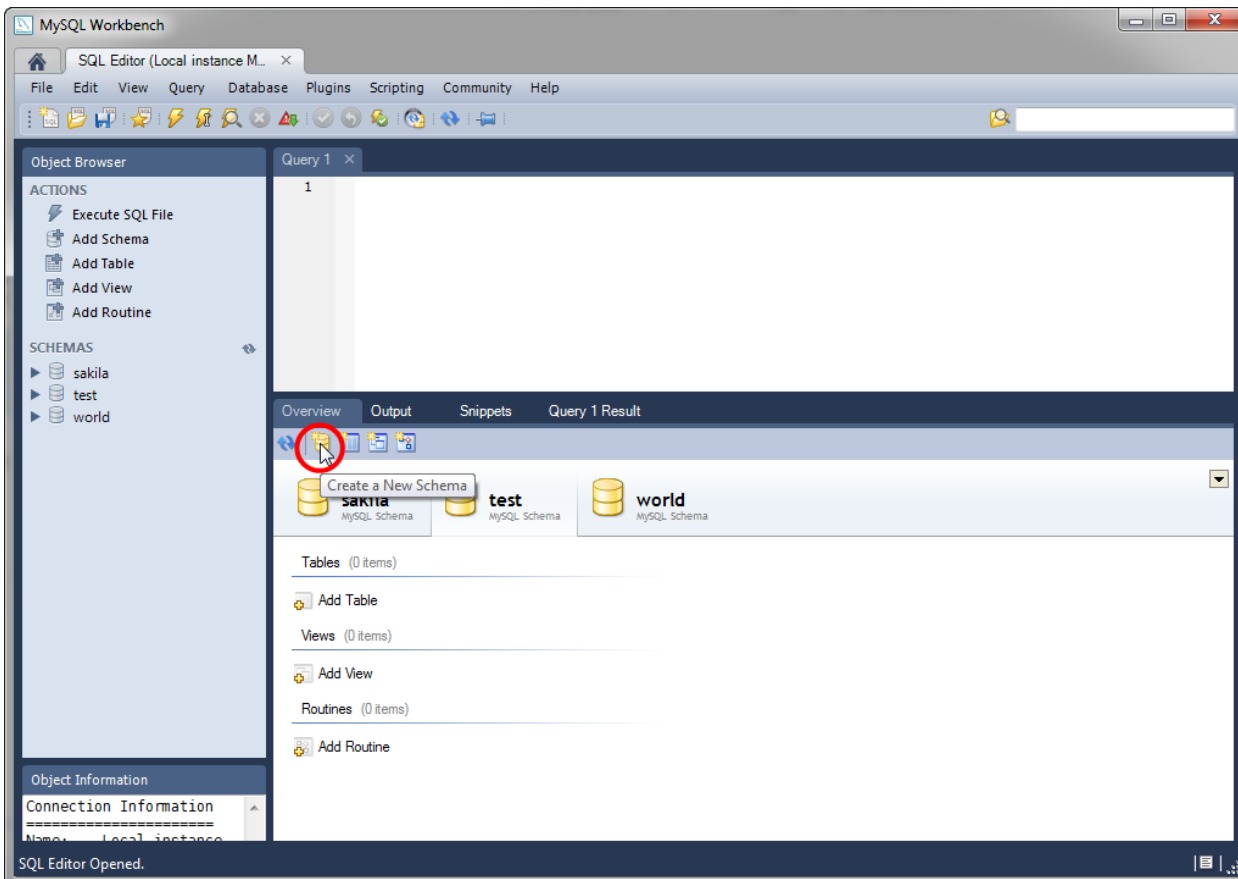
### 2.2 The short version for advanced users:

1. Download the server code and extract to a folder.
2. Download and install Grails.
  - <http://grails.org/>
3. Download and install MySQL.
  - <http://www.mysql.com/>
4. On MySQL, you must do the following:
  - Set up a database called “mentalhealthliberia”
    - create database mentalhealthliberia
  - Setup a user “mhladmin” with password “adminpass”
    - create user 'mhladmin'@'localhost' identified by 'adminpass';
  - Setup “mhladmin” with full access on DB mentalhealthliberia.
    - grant all on mentalhealthliberia.\* to 'mhladmin'@'localhost';
5. Start command line/shell and enter the folder that contains the server code and enter:
  - grails run-app

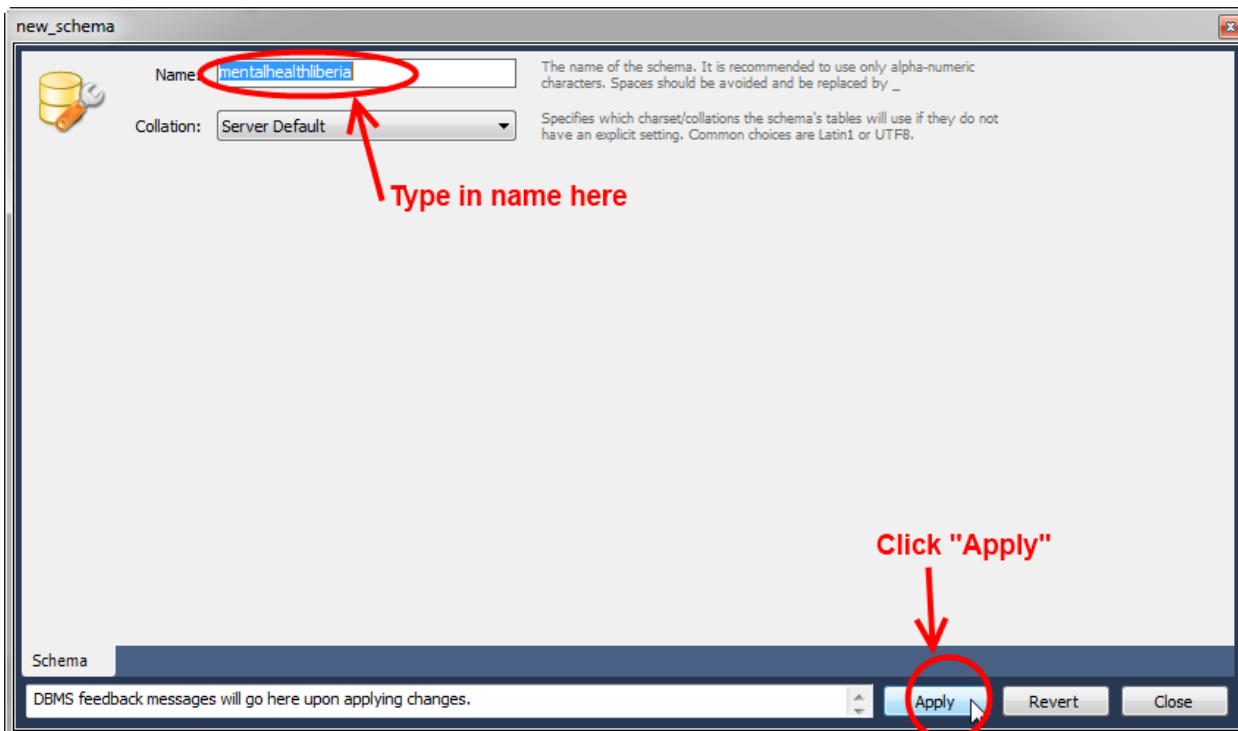
## 2.3 The long version:

1. Download the server code and extract to a folder. We suggest utilities such as 7-zip.
  - Server files are located under eclipse\_workspace folder.
2. Download and install SpringSource Tool Suite
  - <http://www.springsource.com>
  - Installer will ask you for the location of JDK. It is usually "C:\Program Files\Java\jdk1.7.0" (or something similar).
  - Install using default option.
3. Download and install MySQL.
  - <http://www.mysql.com/>
  - Install using "developer" option.
  - Use default options.
4. Configuring MySQL
  - Start MySQL Workbench, double click the local instance.
  - Note: Connection name might not match word to word.

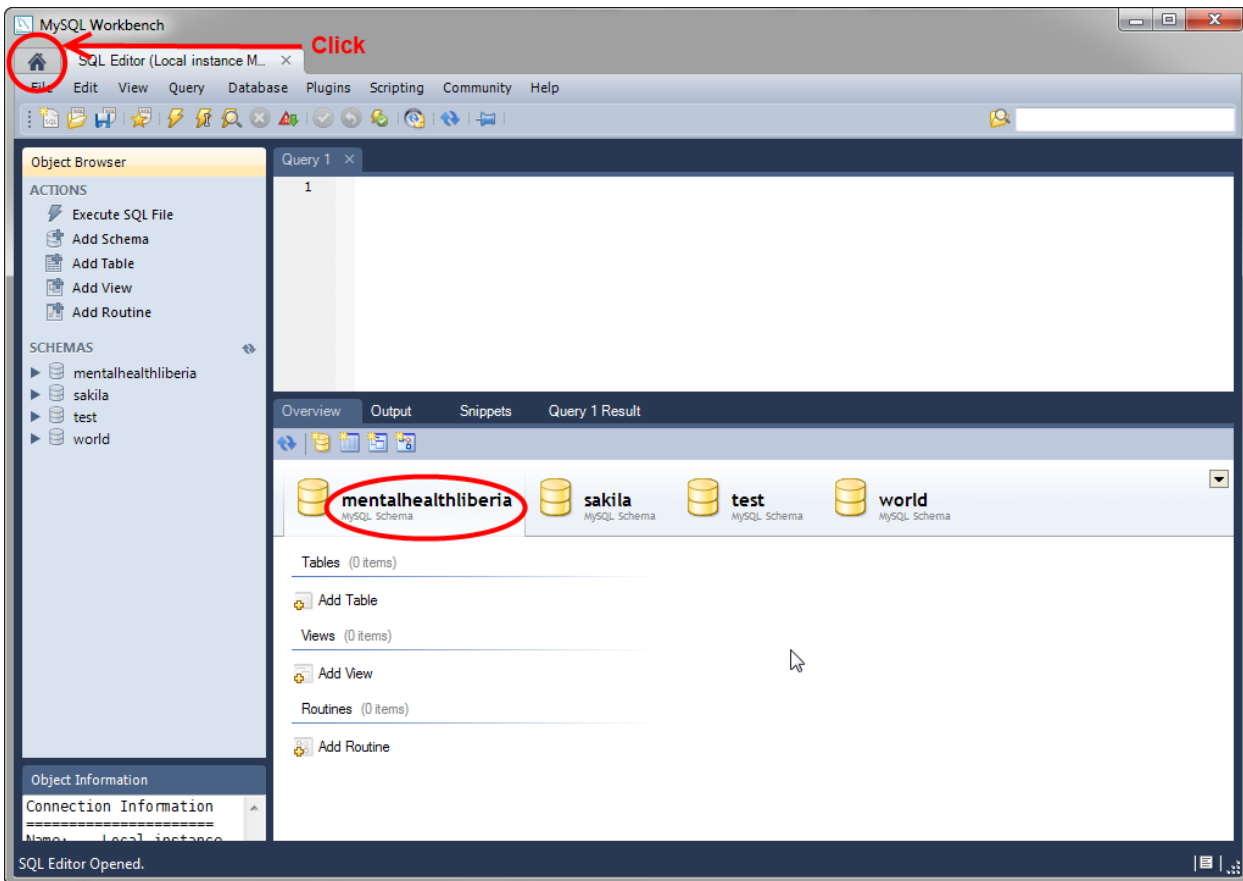




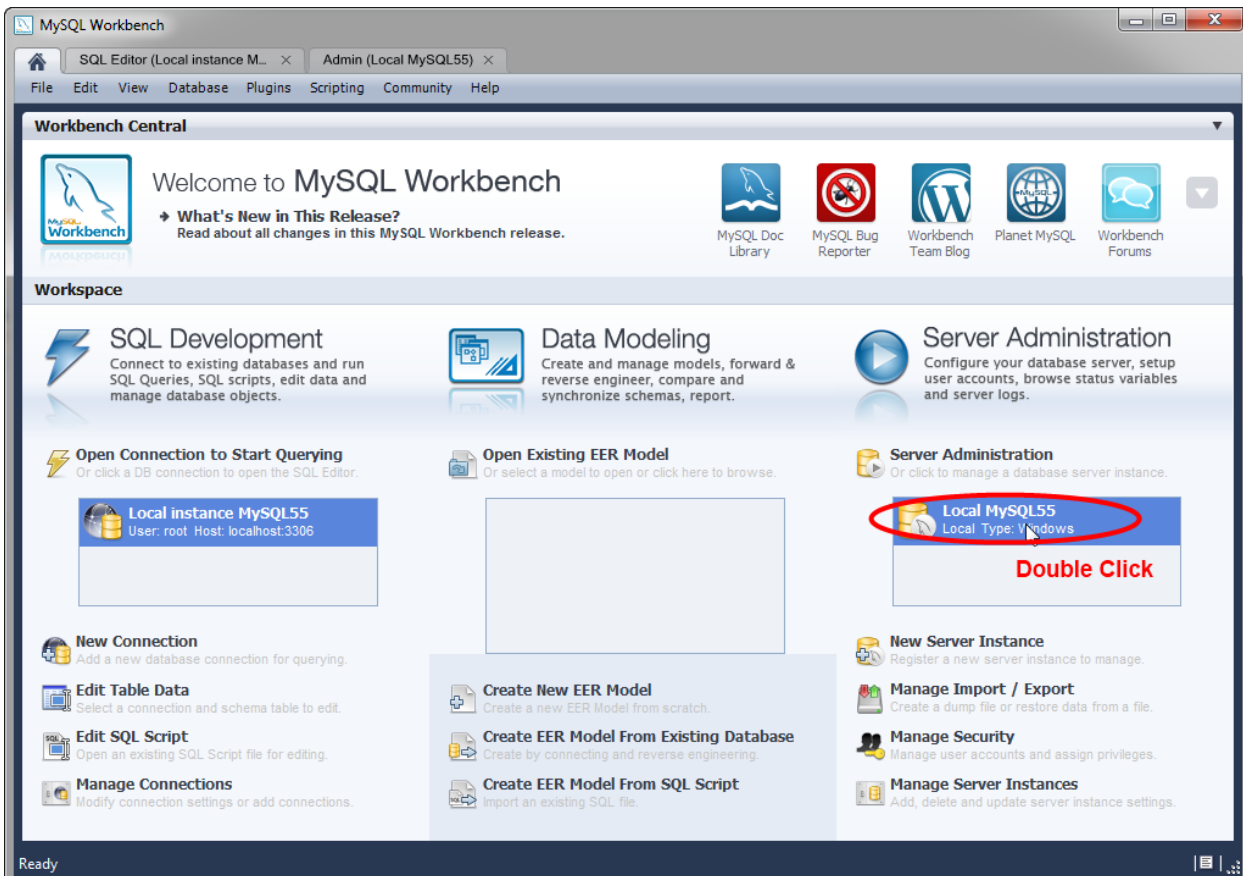
- Hit "Create a New Schema".



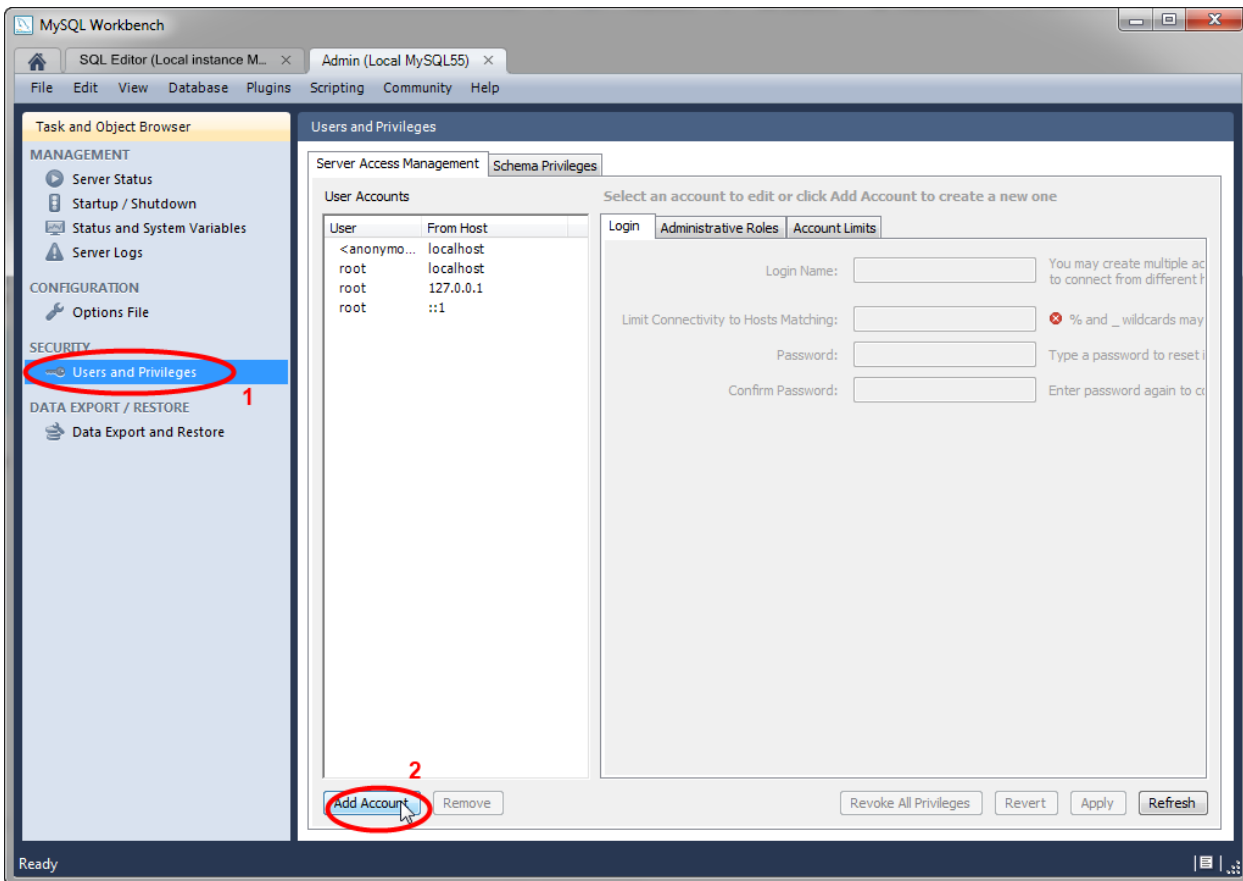
- Create "mentalhealthliberia" schema



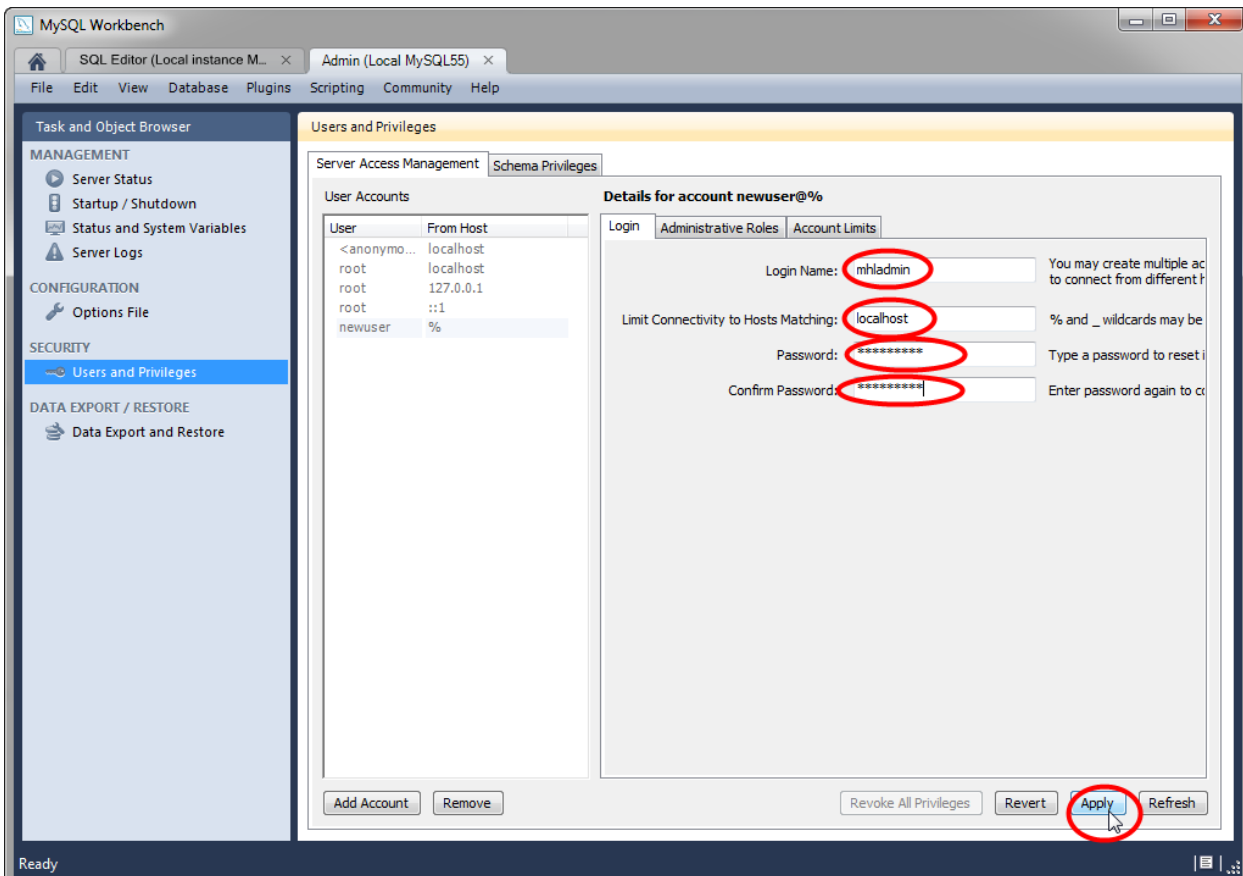
- Verify that “mentalhealthliberia” schema is created. Now go back to the main menu.



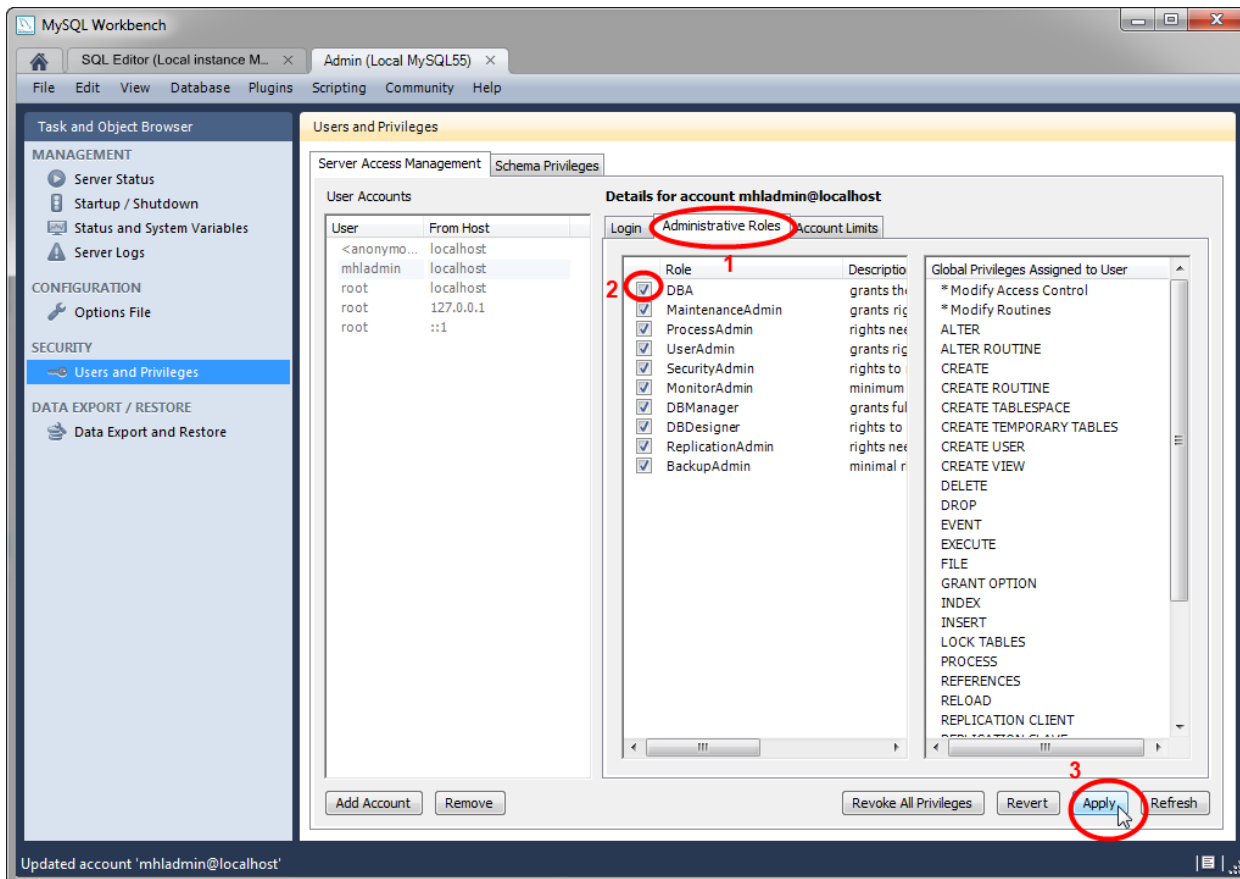
- Double click the server instance (instance name may differ).



- Click on “Users and Privileges” then click “Add Account”.



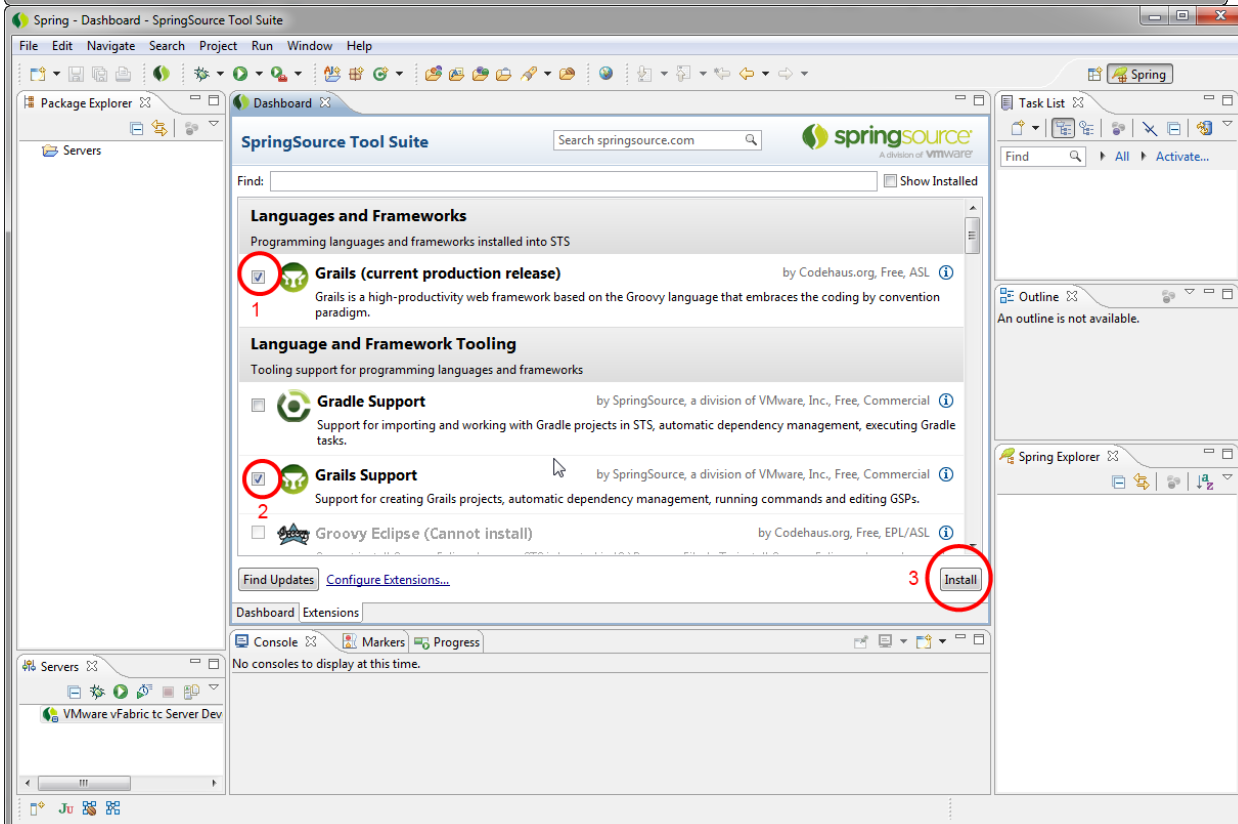
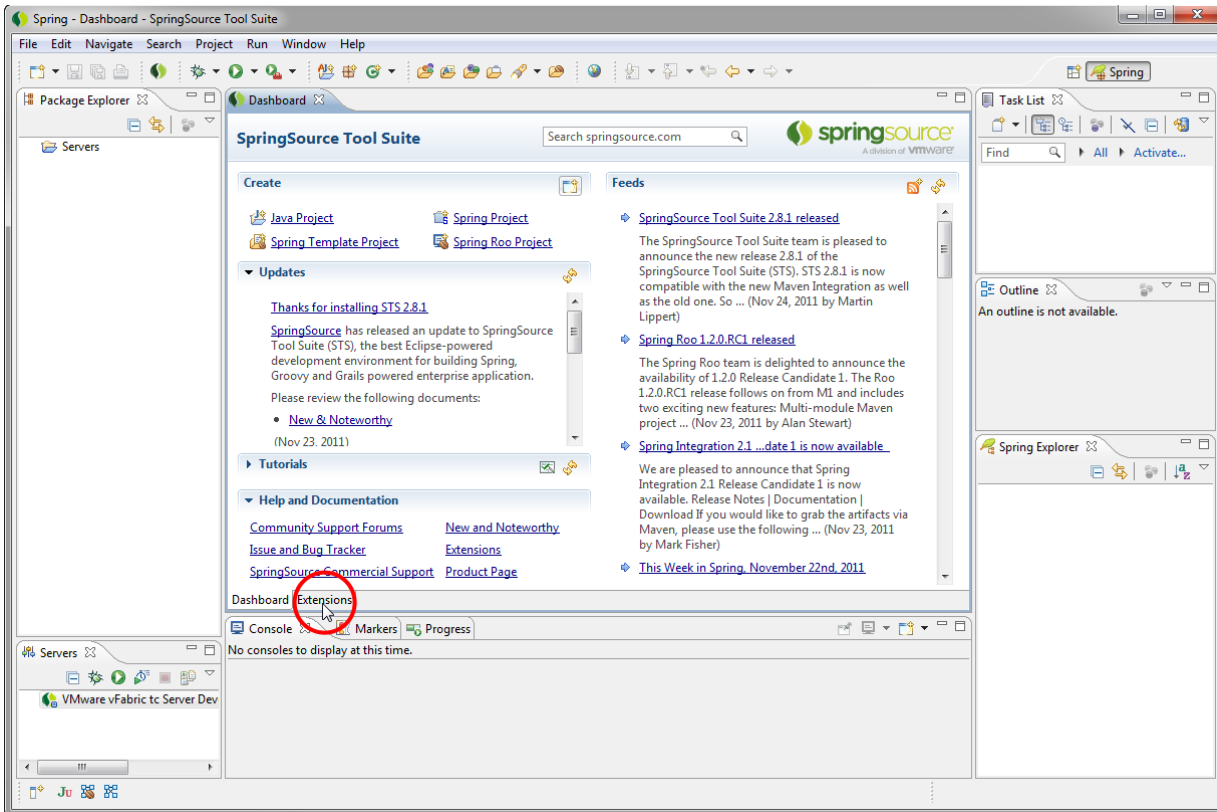
- Type: mhladmin, localhost, adminpass, adminpass on each field and press “Apply”.



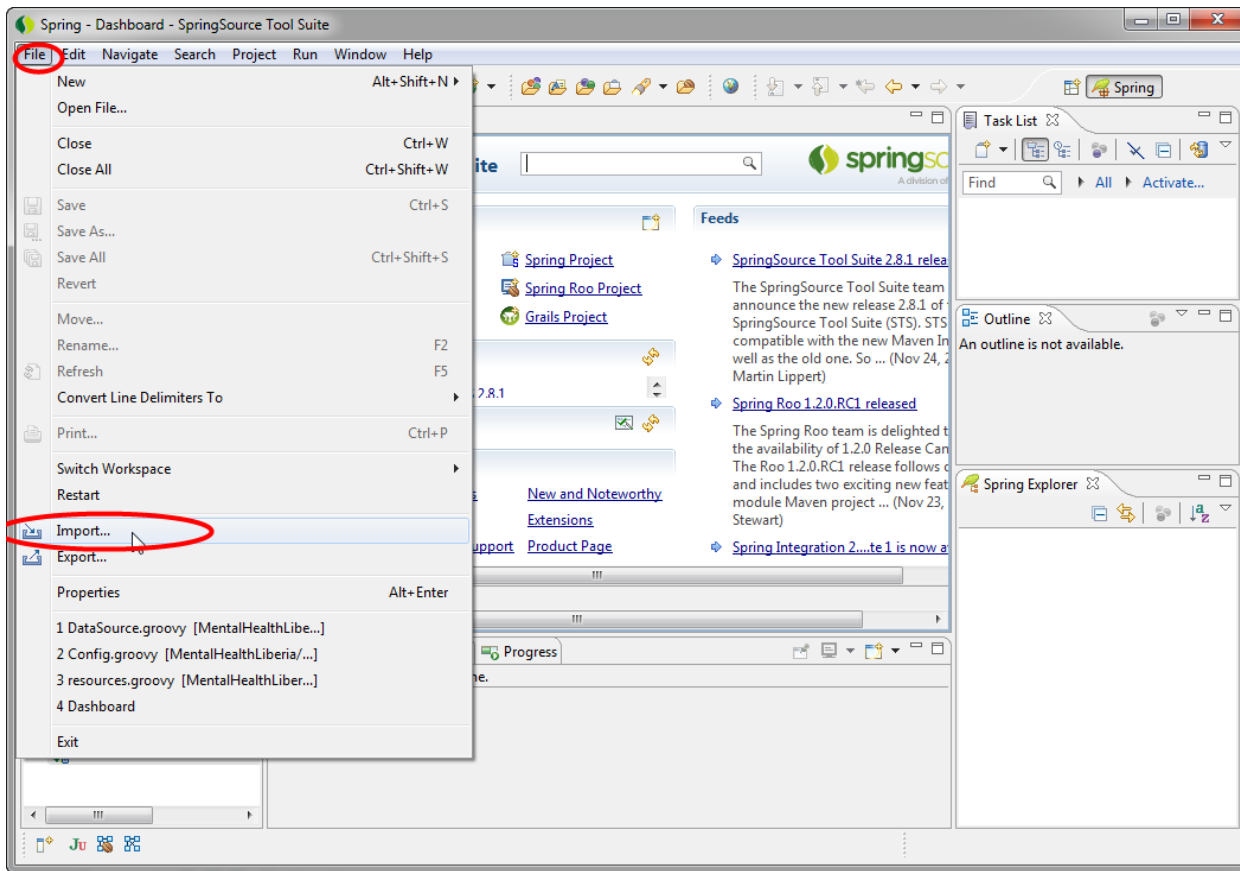
- Click “Administrative Rules”, “DBA”, then “Apply”.

## 5. SpringSource Tool (SST) Suite

- Once you start it for the first time, it will ask you for workspace location. You can use the default location. Optionally, you can select “Use this as default and do not ask again”.
- Click on extensions, then select “Grails” and “Grails Support” and install them.

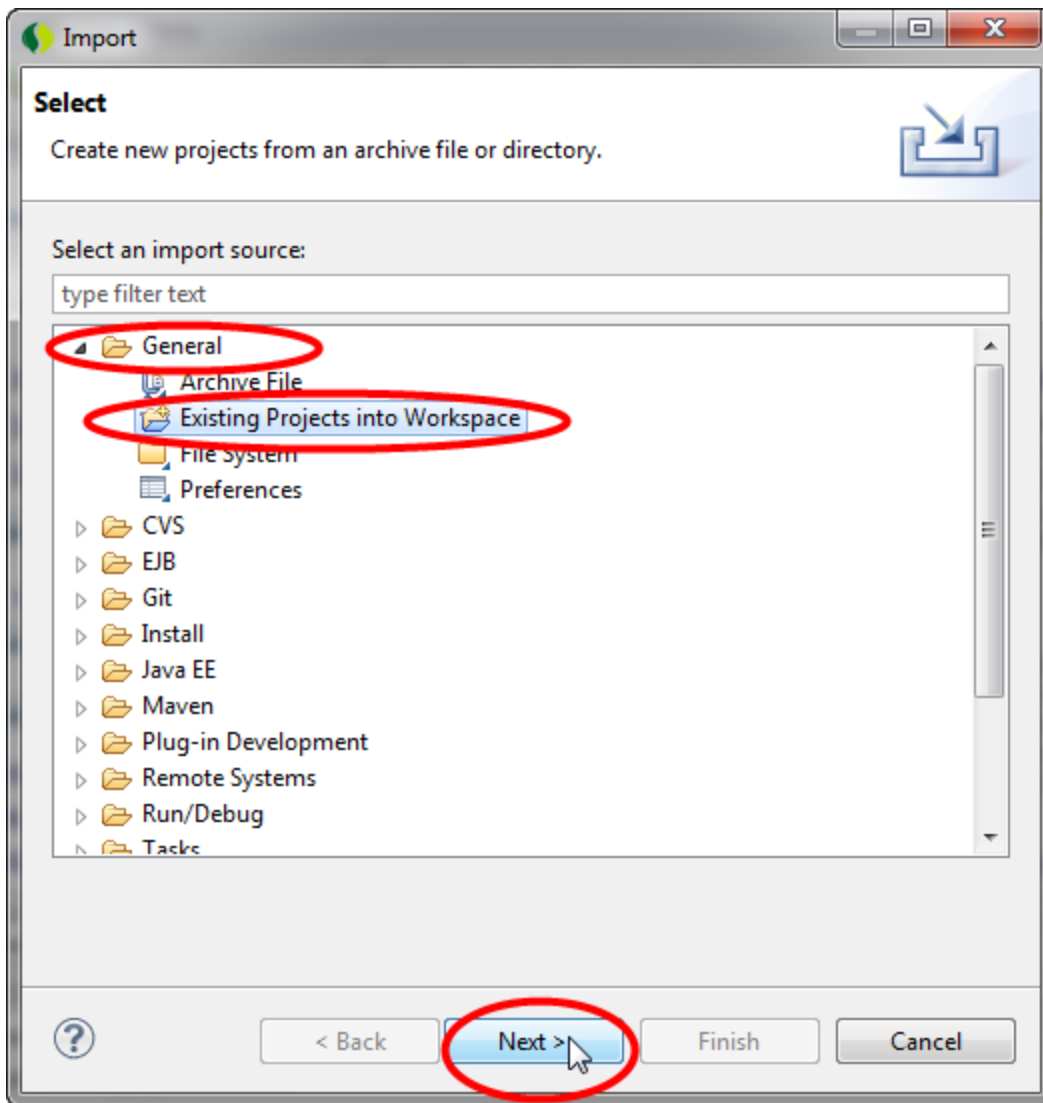


- Afterwards, restart the SST and import the project.

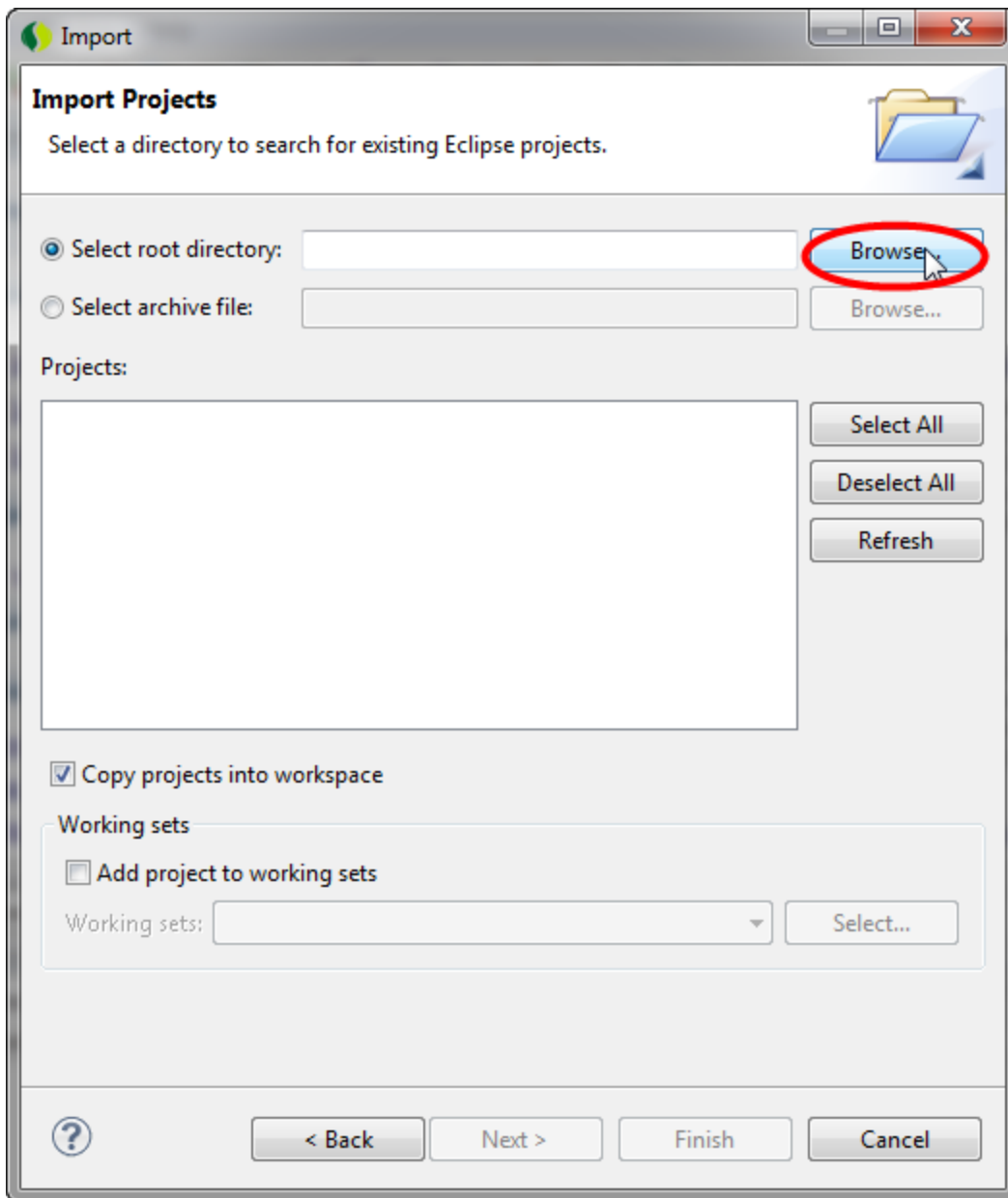


- Click File->Import.

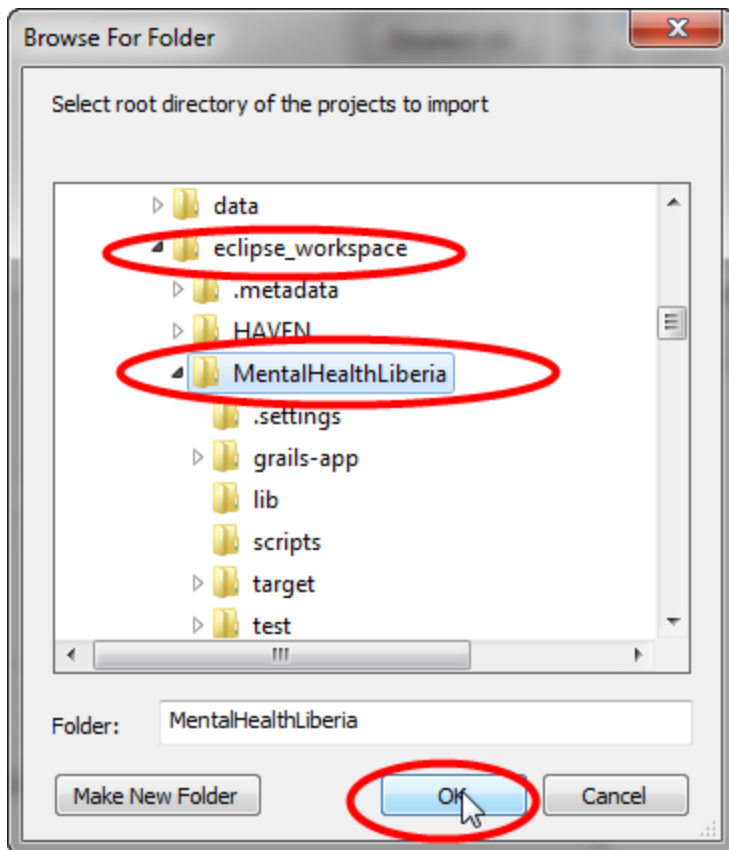




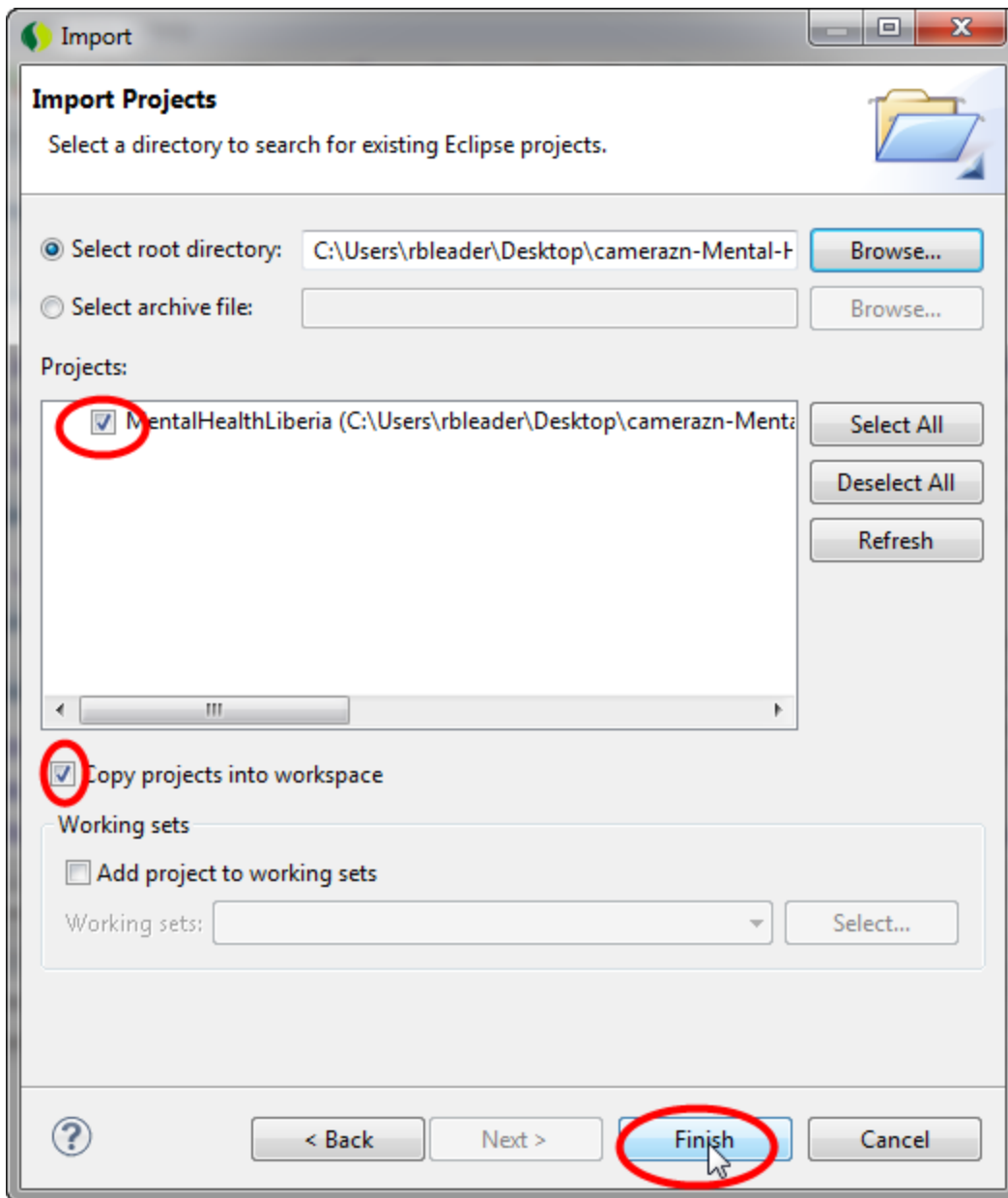
- Select General->Existing Projects into Workspace, then click "Next".



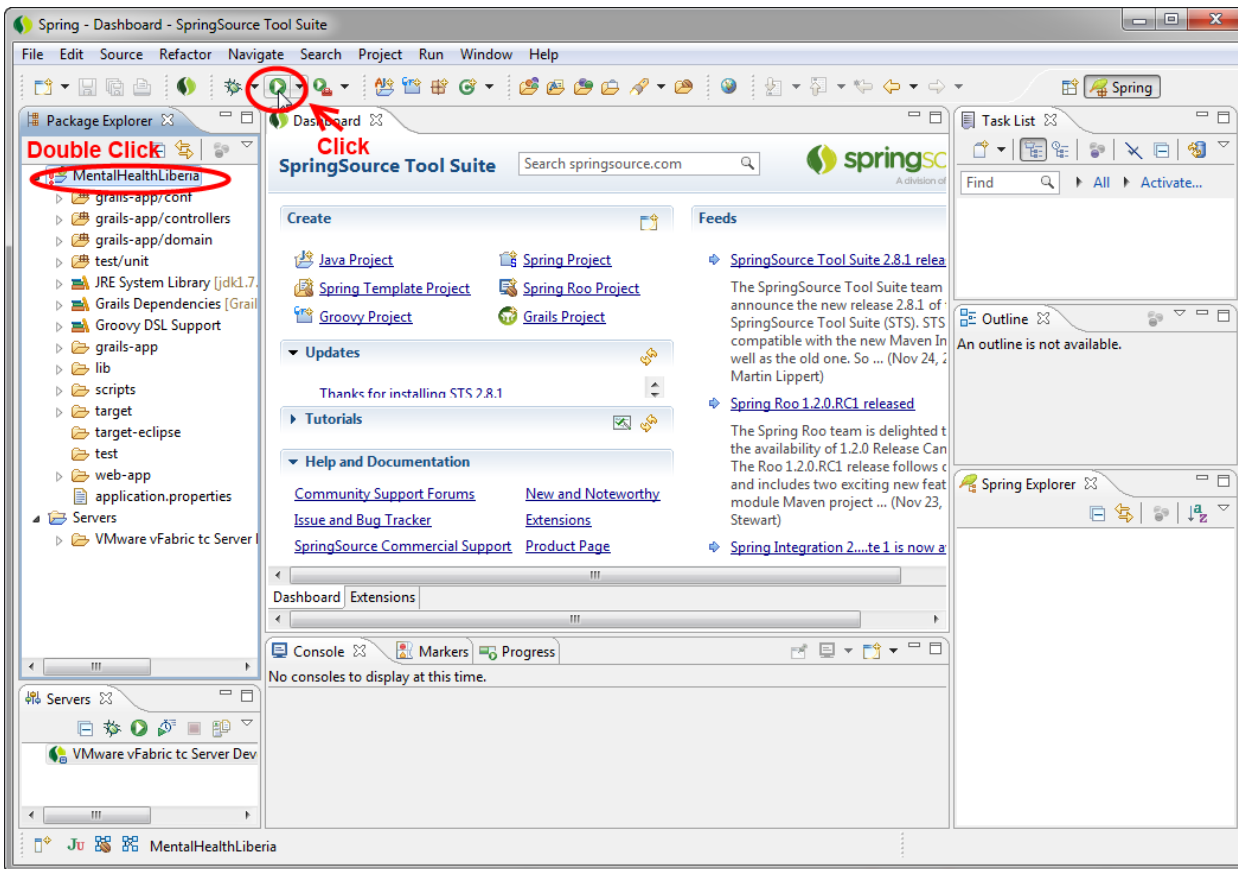
- Click "Browse..."



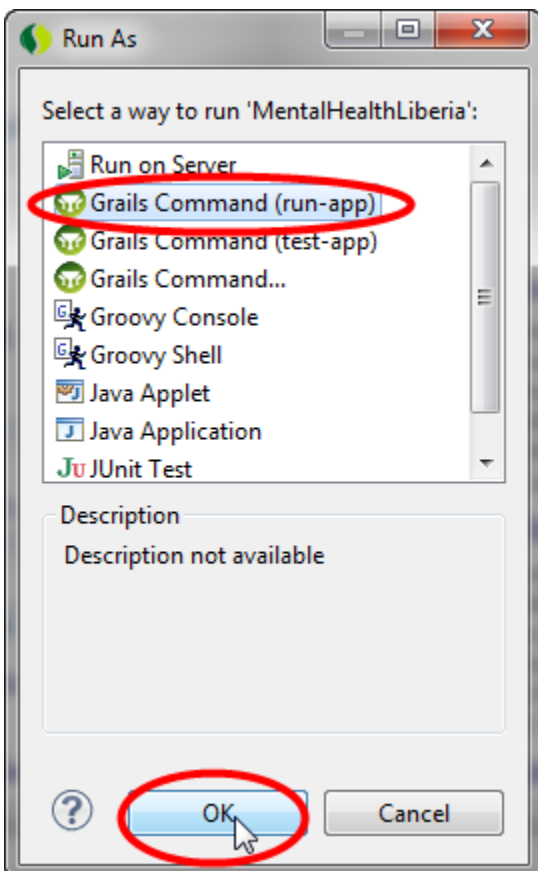
- Locate the folder you extracted the source code, select “MentalHelathLiberia” folder under “eclipse\_workspace”, then click OK.



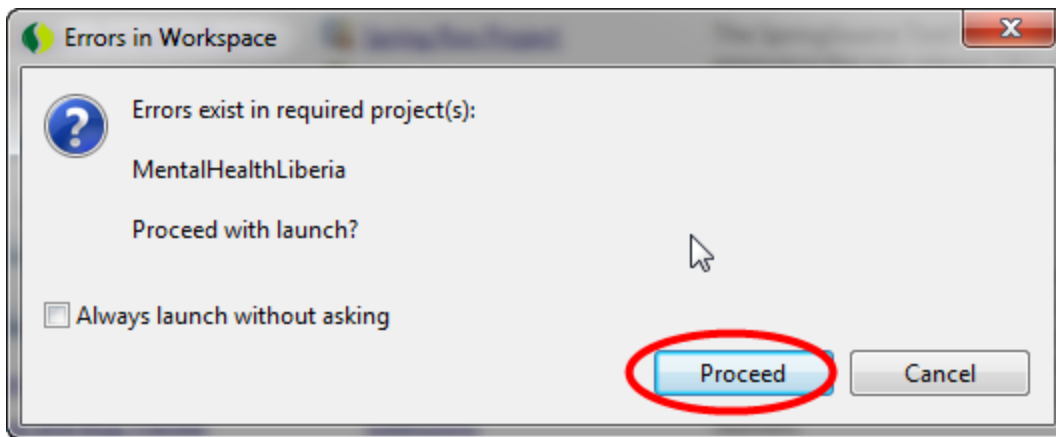
- Note that now you have a valid checkbox under [Projects:]. Click “Finish”.
- Optionally, click on “Copy projects into workspace” if you want SST to copy source files to YOUR workspace.
  - If you do so, changes made in SST are saved on your workspace, not where you unzipped the files.



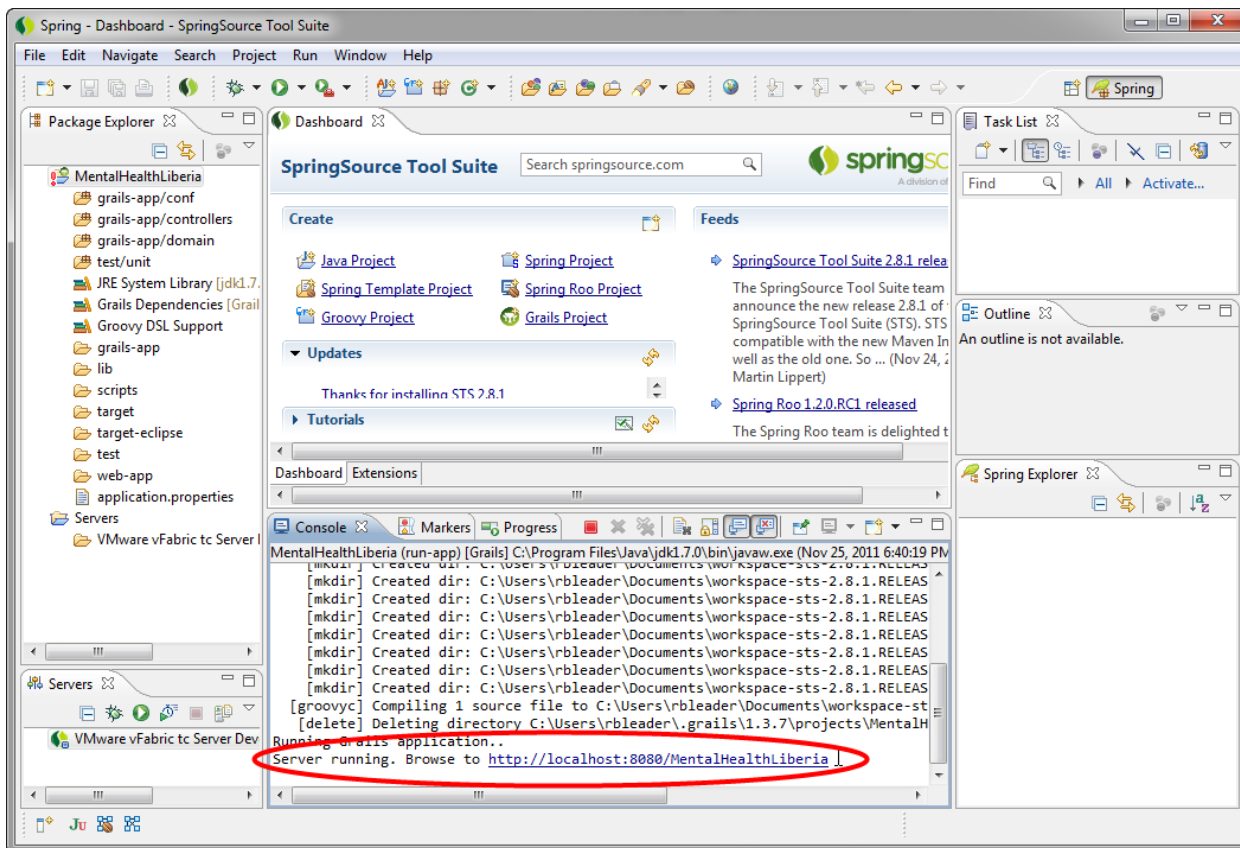
- Double click “MentalHealthLiberia” then click the “Run” icon.



- Click “Grails Command (run-app)” then OK.



- You might get this window. Click Proceed.



- If you were following this instruction to the letter, code will compile and you will see the message "Server running. Browse to http://localhost:8080/MentalHealthLiberia". Now you have a local server running. To verify, you can open up a web browser and type in the above address to the address bar.

### 3. Running the client code:

#### 2.1 Prerequisite:

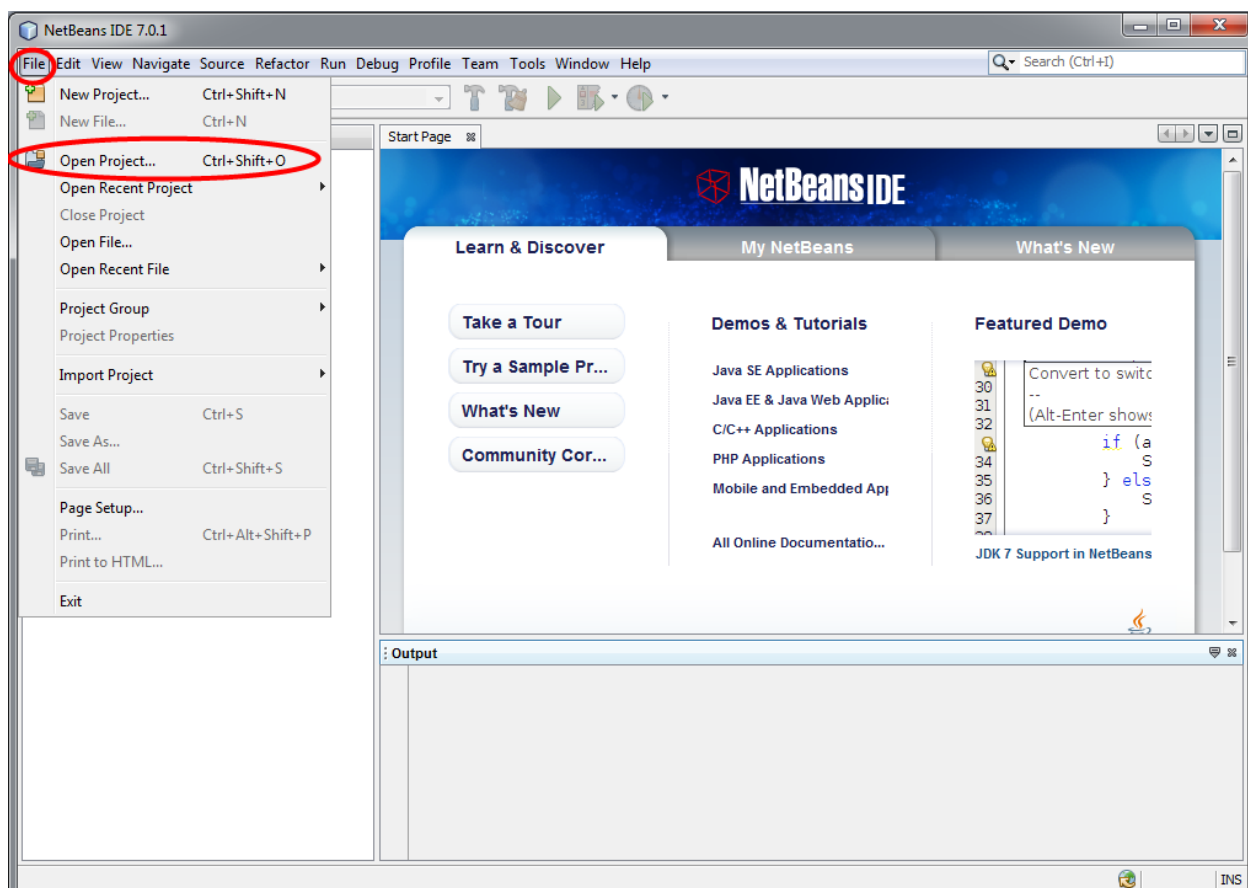
1. You must have the latest JDK (Java Development Kit) installed on your system. You can download it from <http://www.oracle.com>. If you do not have JDK on your system, please download and install it. For more information about JDK, please refer to <http://www.oracle.com>.
2. You must have the latest version of NetBeans installed on your system. You can download it from <http://netbeans.org>. If you do not have this program on your system, please download and install it. For more information about JDK, please refer to <http://netbeans.org>.

#### 2.2 The short version for advanced users:

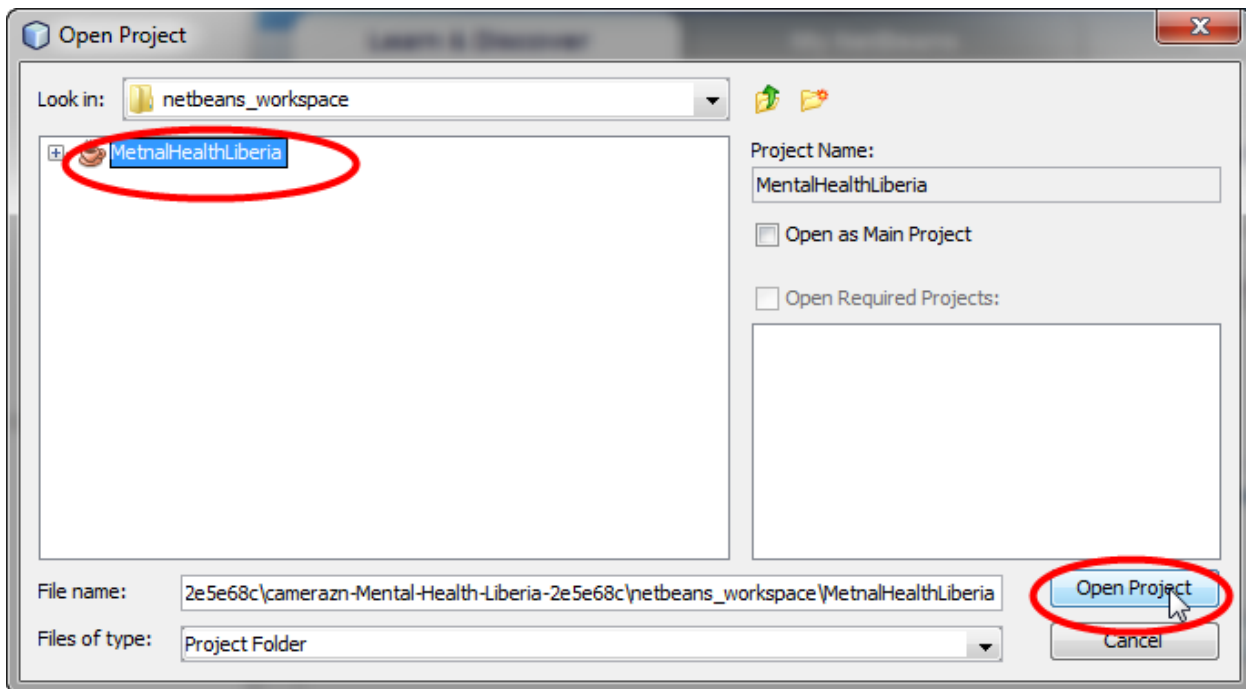
1. Start NetBeans.
2. Open the MentalHealthLiberia Project.
3. Run.

#### 2.3 The long version:

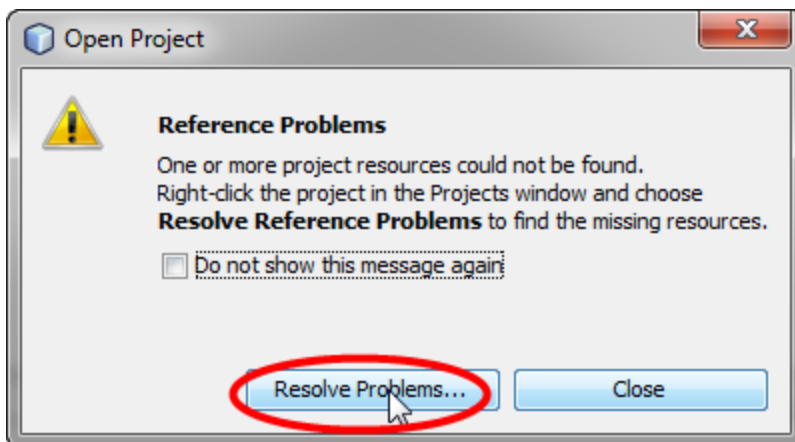
1. Start NetBeans.
2. Click on File->Open Project



3. Navigate to the client source folder (\netbeans\_workspace), then click MentalHealthLiberia project.

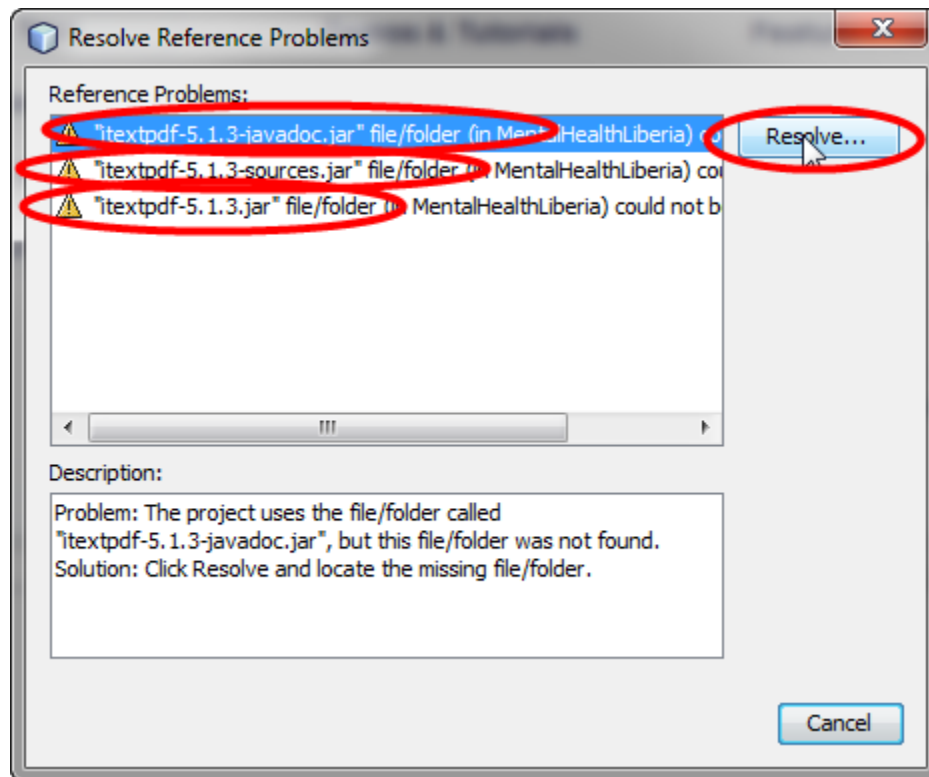


4. You might get this window. Click "Resolve Problems..."

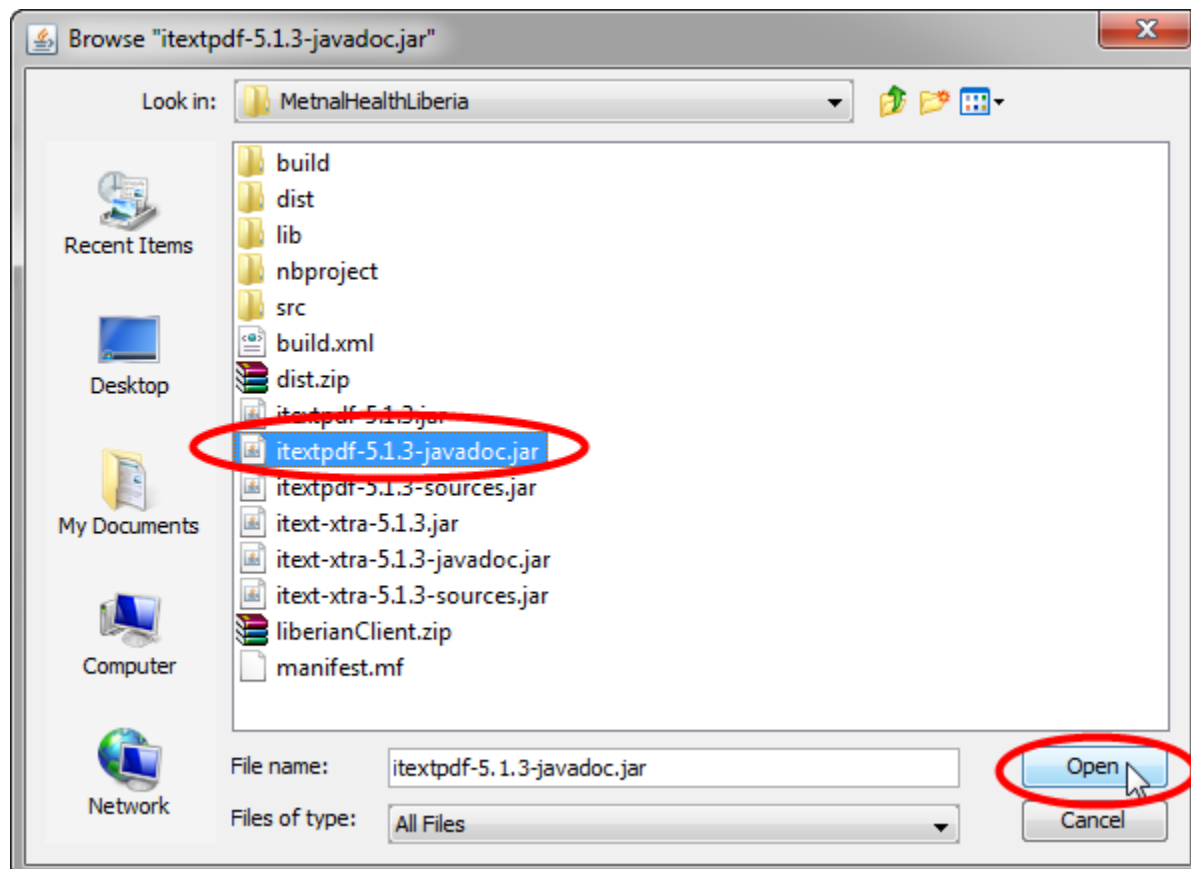




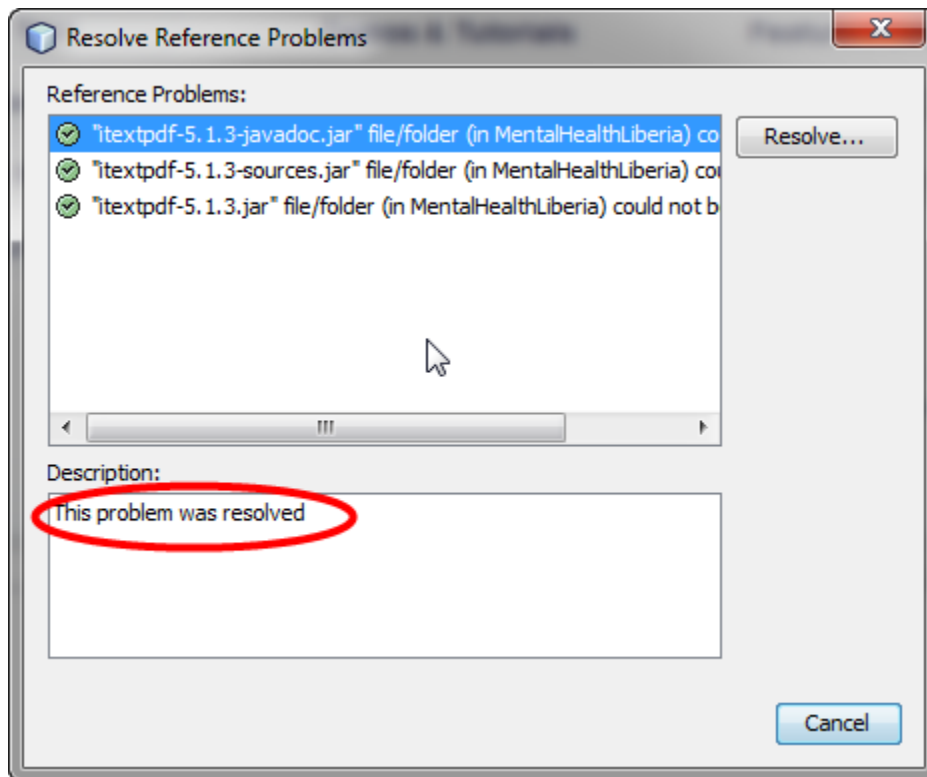
5. Download itextpdf from <http://itextpdf.com/>. Extract the files somewhere, and click resolve button.



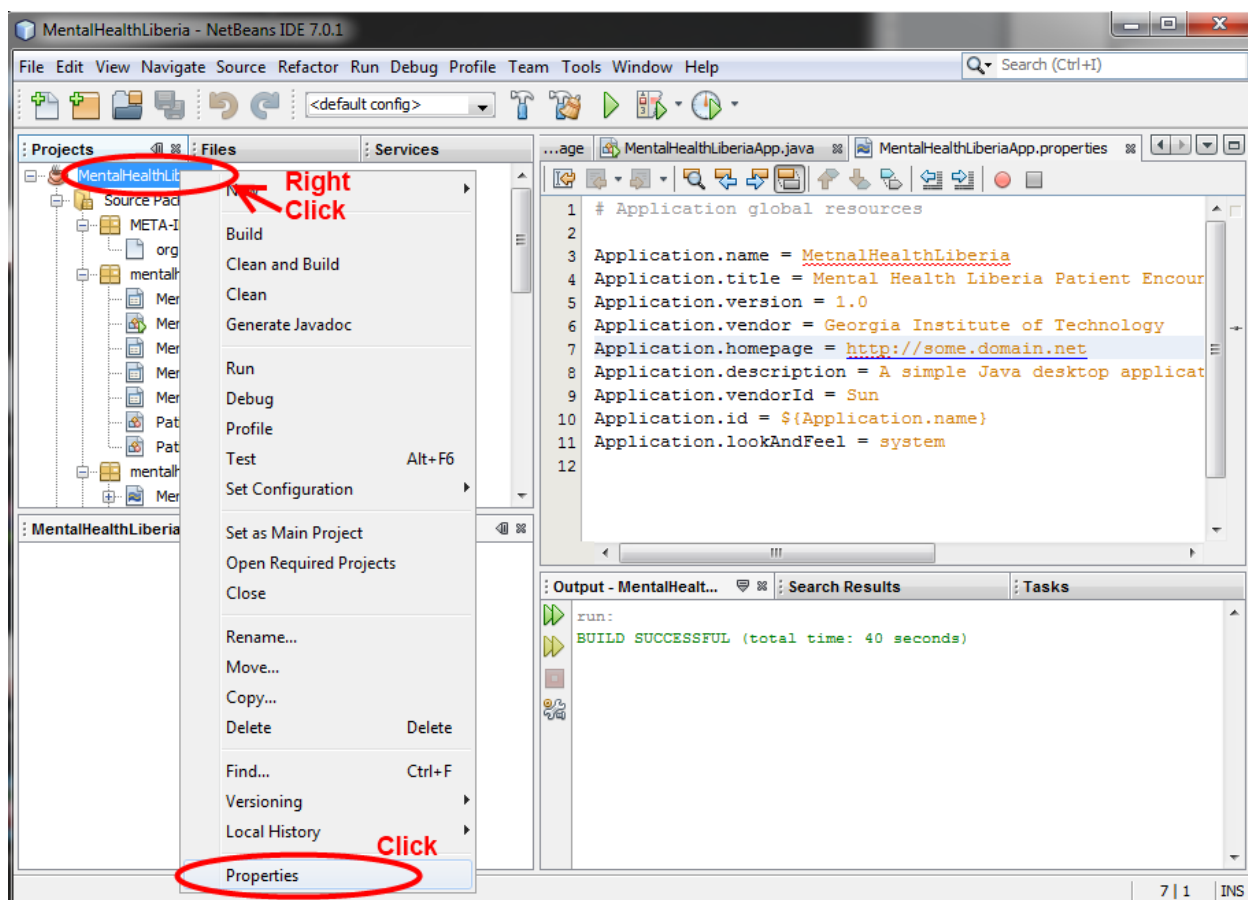
6. Locate the file with same name, and click open. Do this for all reference problems.



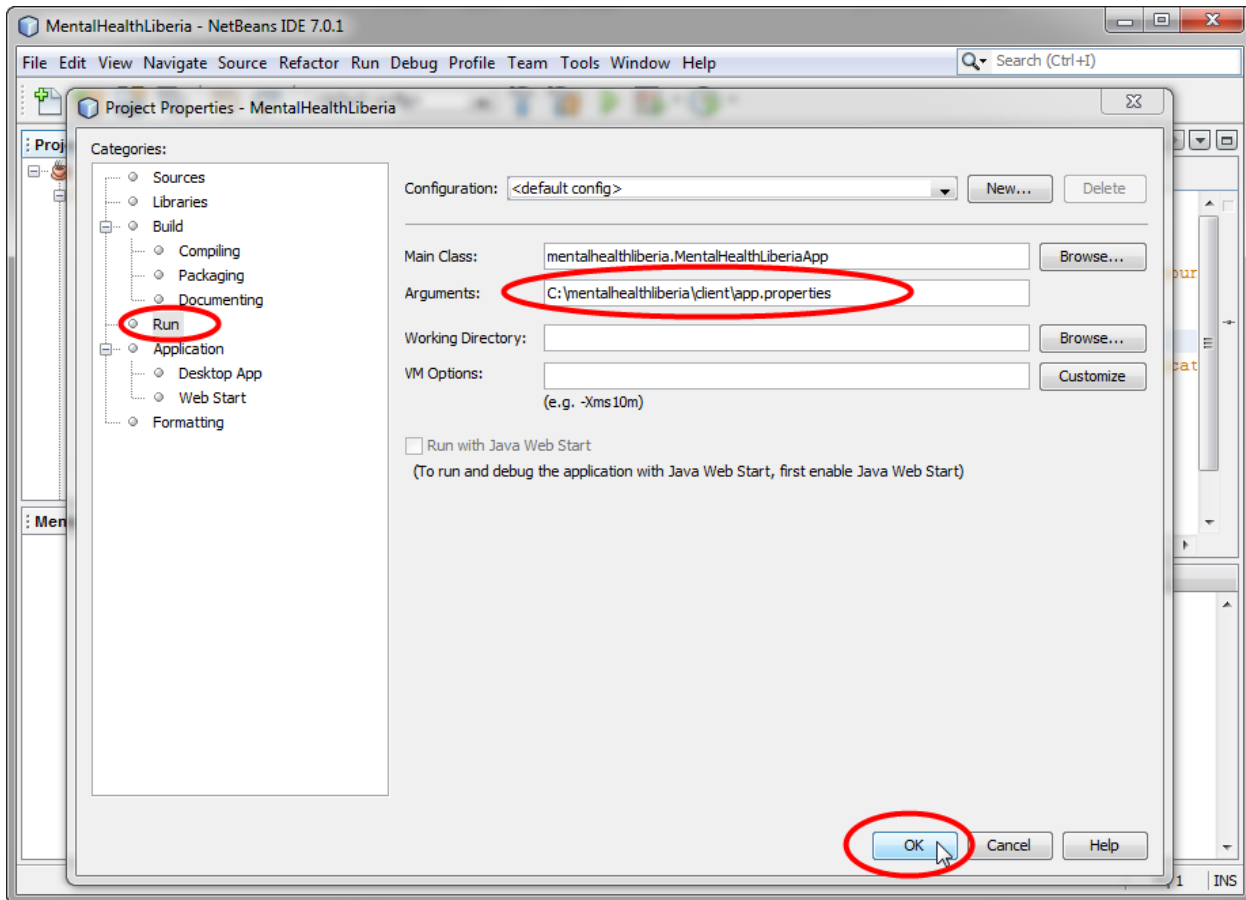
7. If you see “This problem was resolved”, then you are done. Close this window.



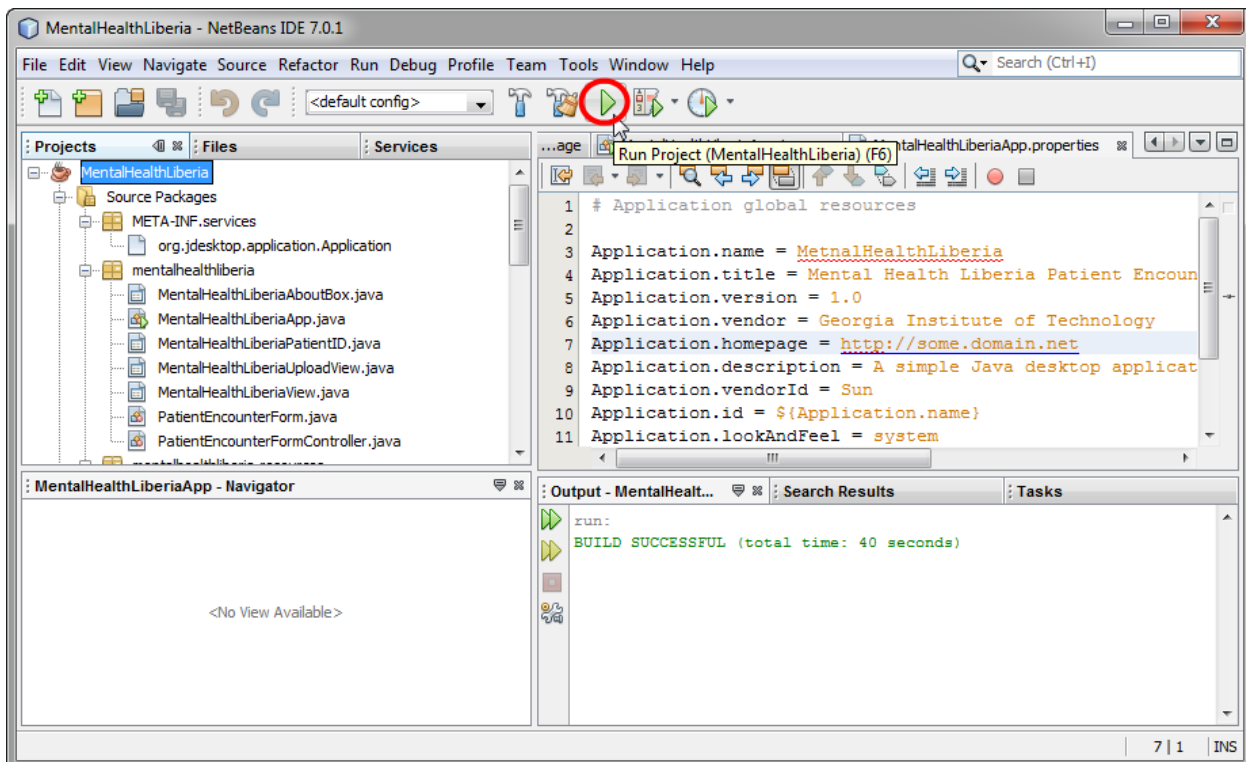
8. Find a file called “app.properties”. Note its location. Now open project properties.



9. Click “Run”, and on arguments, type in the file path for the app.properties file. (may not be same as screenshot) Click OK.



10. Click “Run Project”.



11. Everything should compile and the project will run.

The screenshot shows a web application window titled "Mental Health Liberia Patient Encounter Application". The window has a menu bar with "File" and "Help". Below the menu bar is a tabbed interface with five tabs: "Basic Information", "Patient Demographics", "Symptoms and Functioning", "Treatment", and "Discharge". The "Basic Information" tab is currently selected. The main content area of the tab contains a form with the following sections:

- Basic Information**
  - Date of Service (mm/dd/yyyy)*: A text input field containing "11/26/2011".
  - Clinician ID*: A text input field.
  - Location of Service (clinic/hospital name)*: A text input field.
  - Clinician Training Level*: A dropdown menu with the following options: "No formal training", "Community health worker training", "Physician's assistant training", "General nursing training", "Psychiatric nurse training", and "Other".
  - Reason for Visit*: A radio button group with two options: "Seeking mental health care" and "Other".
  - Patient ID*: A text input field containing "Not generated". Below it is a button labeled "Generate Patient ID".
  - Referral Source*: A dropdown menu with the following options: "Clinician", "Messages (radio, etc)", and "Other".

### 3. Deploying the server code:

#### 3.1 Prerequisite:

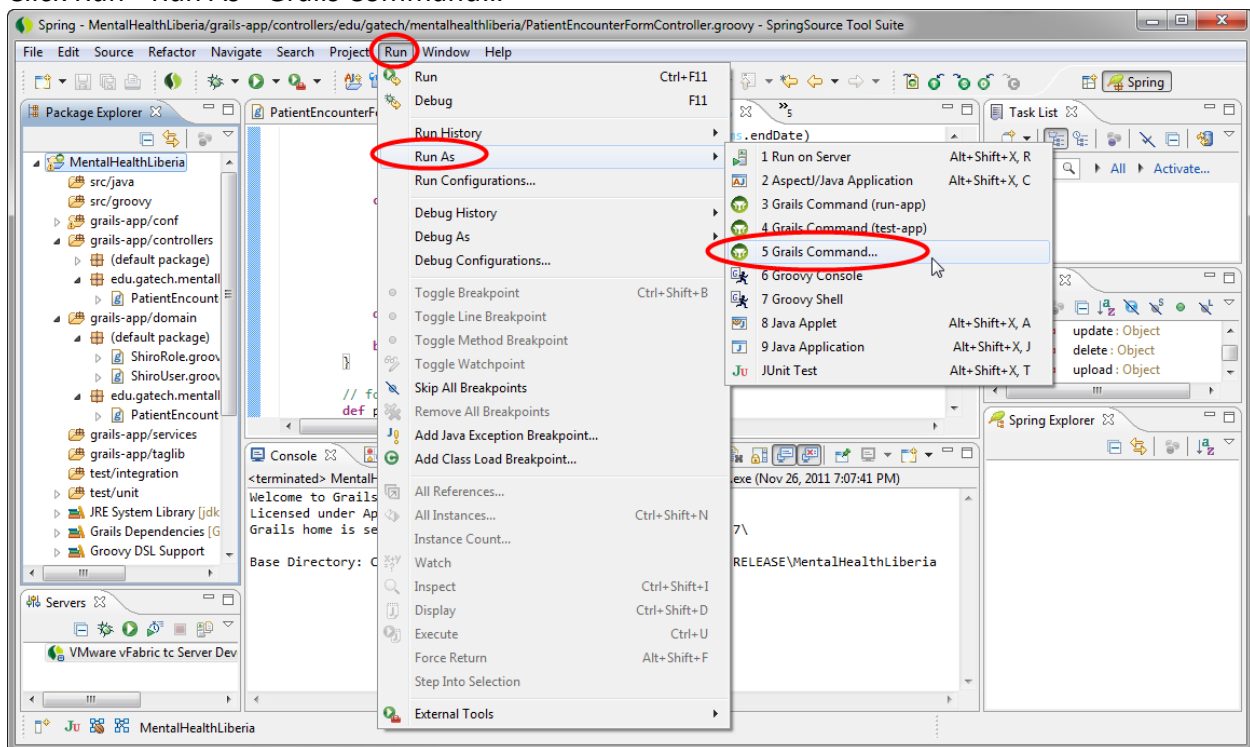
1. You must be able to compile the server code. This means you must at least have grails installed on your system.

#### 3.2 The short version for advanced users:

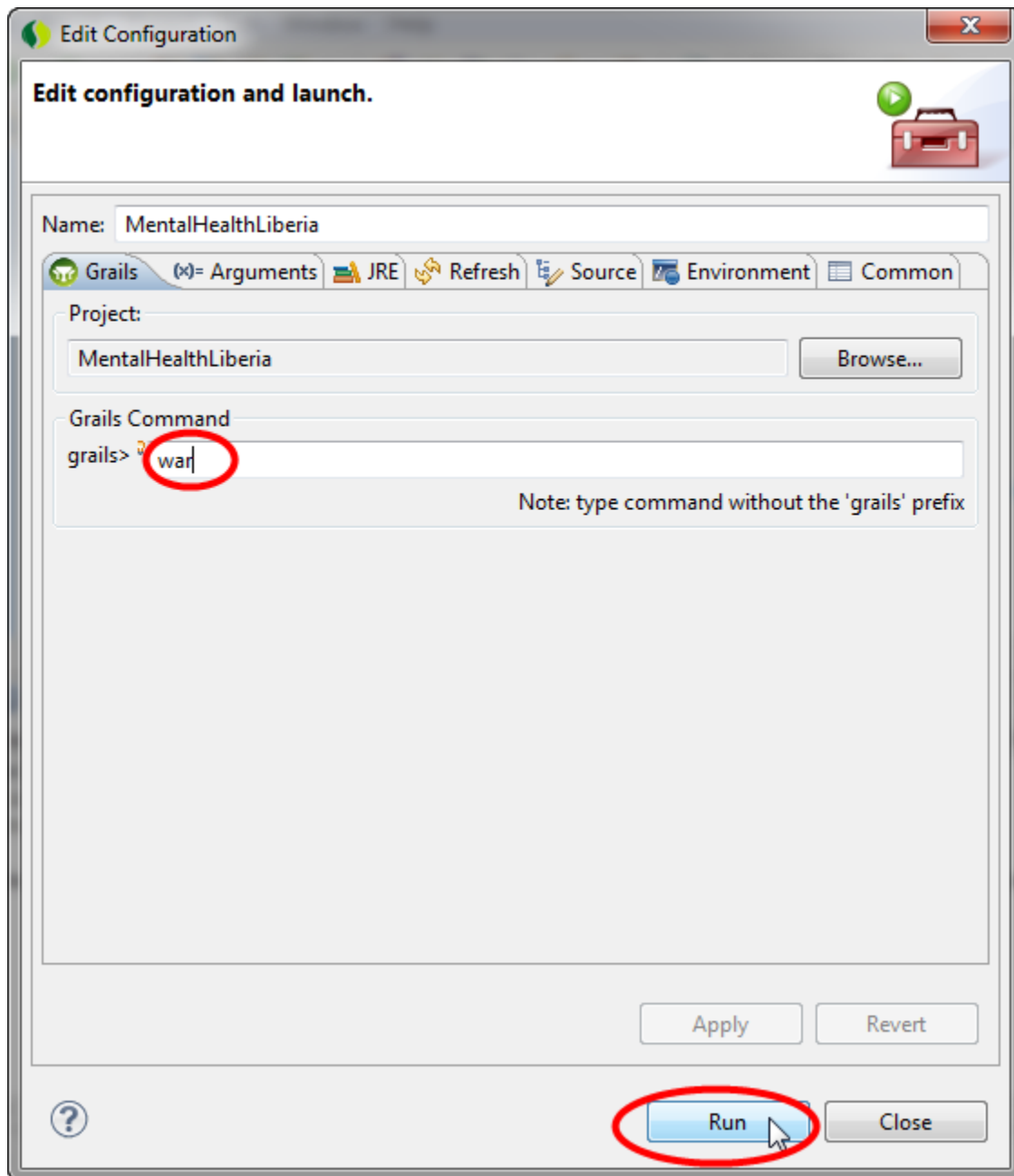
1. Start command line/shell and enter the folder that contains the server code and enter:
  - o grails war
2. Code should compile, giving you the .war file under \MentalHealthLiberia\target folder.
3. Upload the .war file to "/var/lib/tomcat6/webapps" (if you are using tomcat6. otherwise, under webapps folder. Please refer to your server document).
4. Restart the server.

#### 3.3 The long version:

1. Click Run->Run As->Grails Command...



2. Type “war” on command line and then click “Run”.



3. Code should compile, giving you the .war file under “\MentalHealthLiberia\target” folder.
4. Upload the .war file from step 3 to “/var/lib/tomcat6/webapps” (if you are using tomcat6. otherwise, under /webapps folder. Please refer to your server document).
  - Please contact the server administrator for help if necessary.
5. Restart the server.

## 4. Server files description:

- For more information, please refer to <http://www.grails.org>.
- Grails-app/conf: Contains initial configurations/bootstrap files.
  - BootStrap.groovy: This file contains commands that will be executed by the program when the server is on for the first time.
    - Useful for making default accounts, data sets, etc, for testing.
  - BuildConfig.groovy: Use this file to configure build options such as directories, dependency resolutions, runtime options, etc.
  - Config.groovy: Use this file to configure server options such as character encoding, cache size, etc.
  - DataSource.groovy: Use this file to configure database application settings (e.g., MySQL/HSQLDB).
  - Searchable.groovy: This is a script that provides Grails Searchable Plugin. Please refer to the file comments for more description.
  - ShiroSecurityFilter.groovy: Automatically generated by Shiro plugin. Used for access control.
  - UrlMappings.groovy: Maps a specific URL to a view/controller/action.
- Grails-app/controllers: Contains “controller”, which handles the user interaction
  - AuthController.groovy: Handles user log-in / log-out.
  - PatientEncounterFormController.groovy: Handles user interaction relating to the patient encounter forms – includes things like uploading a form, viewing a form, generating analysis reports using collected form data, etc.
    - If you wish to add in different functionalities, you probably have to modify this file.
- Grails-app/domain: Contains domain objects – which represent the “persistent data”.
  - ShiroRole.groovy & ShiroUser.groovy: Used by Shiro plugin for configuring user object.
  - PatientEncounterForm.groovy: This configures the DB schema for the patient encounter form.

## 5. Client files description:

- mentalhelathliberia\
  - MentalHelathLiberiaApp.java: This file contains the main class of the application.
    - For the most part, you will modify this file to add more functionality.
  - PatientEncounterForm.java: Defines a PatientEncounterForm class, which includes the data structure and data getters/setters.
  - PatientEncouterFormController.java: Contains codes for utility functions for the PatientEncounterForm. This includes saving forms to a file, outputting a saved form as a PDF file, etc.
  - Swing/AWT auto generated files: These files make up the UI portion of the application and automatically generated by NetBeans.
    - MentalHelathLiberiaAboutBox.java: About box
    - MentalHealthLiberiaPatientID.java: Generating patient ID
    - MentalHealthLiberiaUploadView.java: Log-in screen
    - MetnalHealthLiberiaView.java: Patient encounter main form
- Mentalhealthliberia.resources\
  - All .properties files in this folder are configuration files for the respective UI files above.
    - Things such as default text over the text box, font, texts on the button, etc.
  - Additionally, you can put image files in this folder for using it in your UI.