KRAD Rice build something cool.

Training Setup

STEP 1: TOOL DOWNLOADS

Download and install the following tools (if not present on your system currently)

JDK 7u11 (or 6u38) - http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download and run the JDK executable (Note: Mac clients can skip this step). If necessary, setup the environment variable JAVA_HOME that points to the JDK installation directory. Then add JAVA_HOME/bin to your PATH environment variable.

Verify

Open a command prompt and type the following:

```
java -version
```

You should see output similar to:

```
java version "1.7.0_11"
Java(TM) SE Runtime Environment (build 1.7.0_11-b06)
Java HotSpot(TM) 64-Bit Server VM (build 23.1-b03, mixed mode)
```

Intellij 12.0.2 - http://www.jetbrains.com/idea/download/index.html

After downloading run the Intellij installer. This will install Intellij onto your system. Next navigate to the location of the Intellij install, and then into the bin directory. Here you should see a file named **'idea.exe.vmoptions'**. Open this file and add the following vm options (note each option should go on a separate line).

```
-Xms128m
-Xmx1024m
-XX:MaxPermSize=512m
```

Note if you are running the 64-bit version you will need to modify the file named **'idea64.exe.vmoptions'**.

Maven 3.0.4 - http://maven.apache.org/download.cgi

This will download a zip file. Extract the contents of the zip onto your local file system.

Next setup the environment variable M2_HOME that points to the location you copied the previous folder to. Add M2_HOME/bin to your PATH environment variable. Finally add the environment variable MAVEN_OPTS with value "-Xmx1024m -XX:MaxPermSize=768m".

Verify

Open a command prompt and type the following:

```
mvn -version
```

You should see output similar to:

```
Apache Maven 3.0.4 (r1075438; 2011-02-28 10:31:09-0700)
Maven home: /usr/local/maven
```

Git 1.8.0 - http://git-scm.com/downloads

After downloading run the Git installer. You will now have Git on your system.

Verify

Open a command prompt and type the following:

```
git -version
```

You should see output similar to:

```
git version 1.8.0.msysgit.0
```

MySql 5.5.29 - http://www.mysql.com/downloads/mysql/

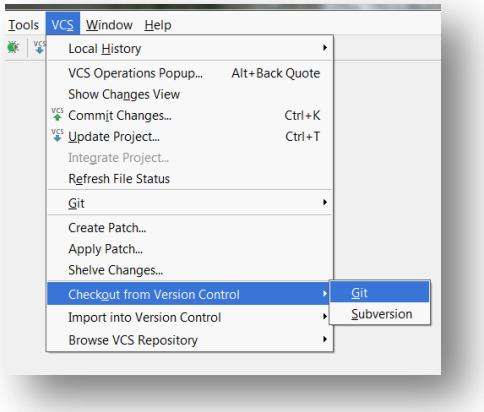
After downloading run the MySql installer. You will now have MySql on your system.

Note it is recommended to also download the MySql Workbench which can be downloaded here: http://www.mysql.com/downloads/workbench/

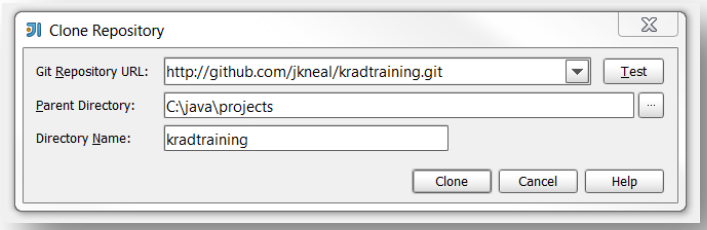
STEP 2: PROJECT CHECKOUT

The training project is maintained with a Git repository provided by GitHub (https://github.com/). You need to have the Intellij Git plugin installed to work with the repository.

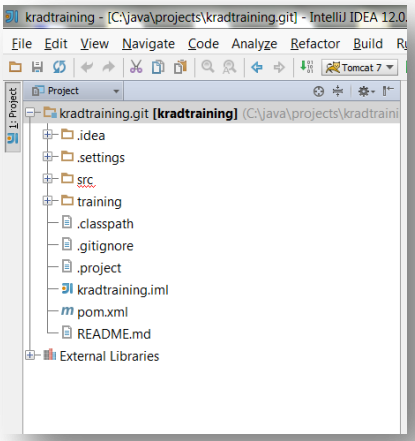
1) In Intellij, choose the menu option VCS -> Checkout from Version Control -> Git



2) In the 'Clone Repository' dialog, enter http://github.com/jkneal/kradtraining.git for the Git repository URL. For the Parent Directory, select the location you want the repository to be located in (for example /java/projects). Finally enter **'kradtraining'** for the Directory Name.



3) After the repository has been cloned, you will see a dialog asking if you would like to open the project. Click the Yes button
4) You should now have the project open in Intellij and are ready to move on. The project pane should look like the following screenshot.

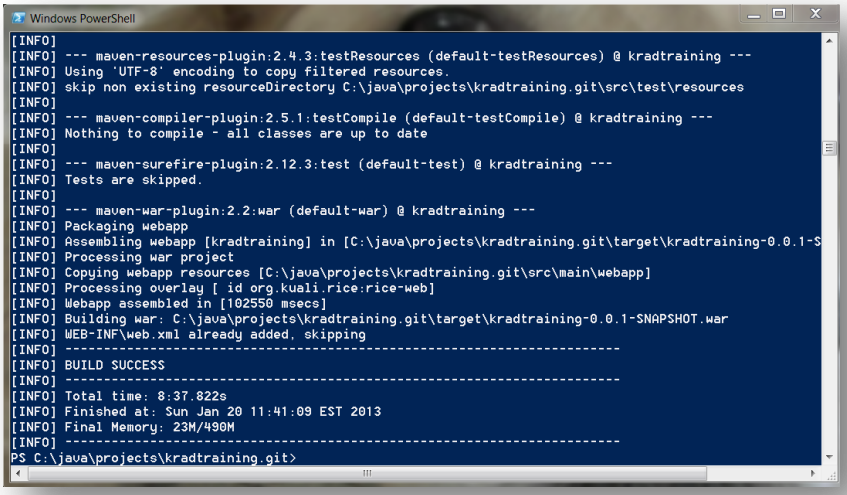


STEP 3: PROJECT BUILD

The training project is a maven project and will need to be built in order to acquire dependencies.

1) Open a command prompt
2) Change into the directory that contains the training project (example /java/projects/kradtraining)
3) Run the following command **'mvn clean package'**

4) After the process is complete you should see the message **'Build Success'**. Note this will take a couple of minutes to complete



STEP 4: TRAINING DATABASE

The database for the training project includes the Rice dataset and a few additional training tables.

1) Follow the steps documented on the confluence page https://wiki.kuali.org/display/KULRICE/Load+Impex+Data+via+Maven to create a new Rice database
2) Navigate to the location of the training project (/java/projects/kradtraining) and open the folder named **'training/db'**. Run the SQL commands in the files **'create_training_tables.sql'** and **'load_training_tables.sql'** against the training database created in step 1
3) After building the database, start up the training application (see below). Navigate to the Administration tab and select the link **'XML Ingester'** within the Workflow channel. Now browse and select the file **'BookTypeKew.xml'** located in the **'training/db'** folder of the training project. Finally click the upload button and verify the you receive a success message

STEP 5: RUNNING THE TRAINING APPLICATION

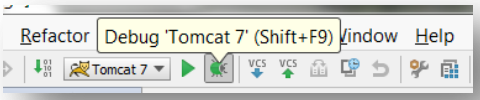
The training application can be run within Intellij. Before running the application you need to place a configuration and keystore file in your user home directory.

1) In your user home directory (for windows C:\Users\{username}), create the folders **kuali/main/dev**
2) Navigate to the location of the training project (/java/projects/kradtraining) and open the folder named **'training/resources/user'**
3) Copy the files **'krtrain-config.xml'** and **'rice.keystore'** to the **{userhome}/kuali/main/dev** folder

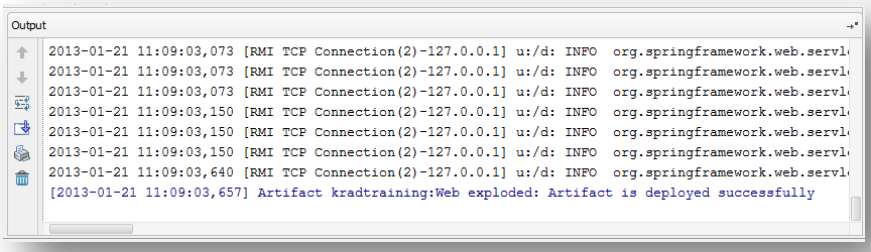
4) If necessary, open the file **'krtrain-config.xml'** and change the database connection information to match the training database you created

Now the application can be run with the following steps:

1) Open up Intellij and the training project (if not opened by default)
2) In the Intellij toolbar you should see a run configuration named **'Tomcat 7'**, click the green arrow next to the run configuration for run mode, or the next icon over for debug mode (Note it is generally recommend to run in debug mode to take advantage of reloading)



3) After the application is loaded you should see the following message in the console:



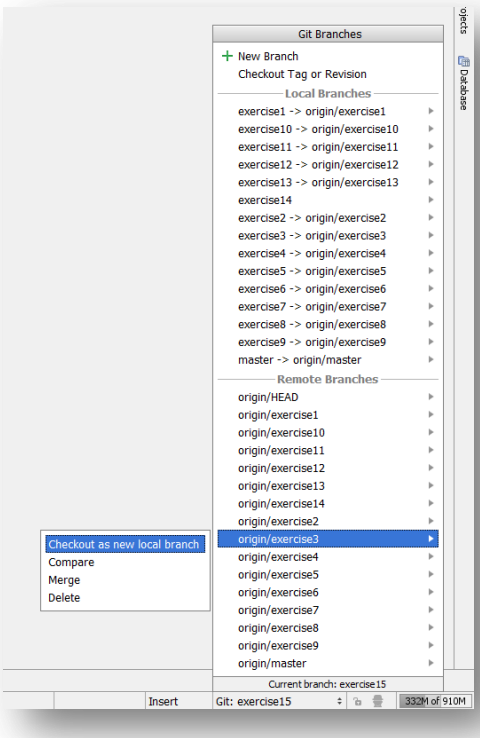
4) Open a browser and then open the URL http://localhost:8080/krtrain. At the login screen, enter the username **'admin'**. Bookmark this location for future use (Note some of the screens, like the training menu will not work yet)

STEP 6: WORKING ON THE EXERCISES

When you initially checkout the training repository you will be working on the master branch. From here you can work on exercise 1 and continue through the training program. However if you get behind, or wish to go back and work on a previous exercise you can do so by switching branches.

Switching Branches

The training repository contains a branch for each exercise. The branches are named as 'exercise' followed by the exercise number. For example, if you



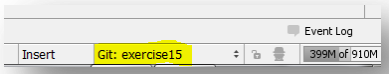
wish to work on exercise 5, you should check out the branch named ‘exercise5’.

To switch branches in Intellij, you can use the menu option **VCS -> Git -> Branches**, or the branch selector in the bottom application window (see screenshot to the right).

Note initially the branches only exist in the remote repository (with name ‘origin/exercise{n}’). Click the remote branch and select ‘Checkout as a new local branch’). This will then give you a local branch named ‘exercise{n}’ and switch your workspace to that branch.

If you have changes in your workspace and request a branch change, Intellij will move them forward if possible. If the changes present conflicts, you will be prompted to override or stash the changes. It is recommend that you commit or stash changes before changing branches. This allows you to come back to this branch and resume work later on.

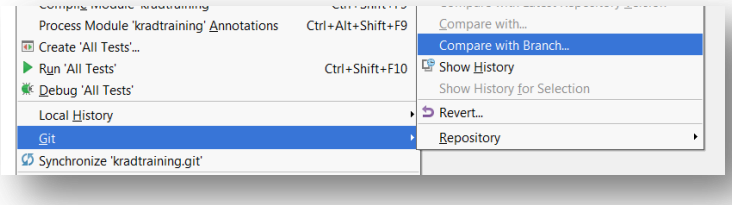
If you forget which branch you are currently working on, this is always displayed by the branch selector.



Checking Your Work

After you have worked an exercise you can check your solution by comparing your current workspace against the branch for the next exercise. For example, if you are working on exercise 1, you can check your solution by comparing against the branch origin/exercise2 (Note you should compare against the remote branches in case you have made modifications to your local branch).

You can compare your workspace against a branch by right clicking on the project (in the project pane) and selecting Git -> Compare with Branch

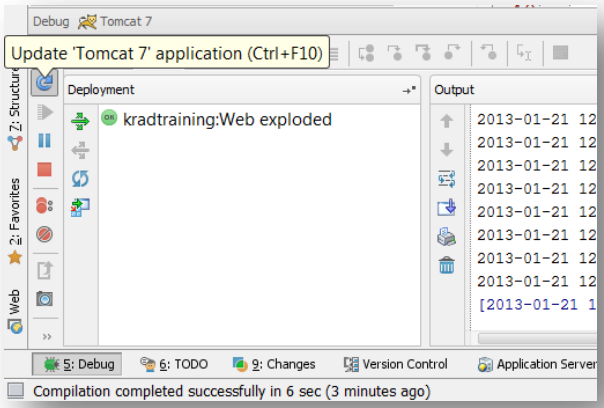


This will bring up a dialog listing all files that have differences. Right click on a file and select ‘Show Diff’ to bring up the compare editor. Note in the compare editor you might need to select ‘Ignore Whitespace’ – ‘All’.

The Reloading Dictionary

The training project by default has the reloading data dictionary enabled. This means when you make changes to a dictionary file you do not need to restart the server. Instead you just need to update resources on the server which will cause a reload of the dictionary beans.

To update resources, select the blue arrow in the Debug pane (see screenshot to the right).



This will bring up an update select dialog (if you don't see this dialog check settings). For dictionary only changes select **'Update resources'**. If you have also made changes to a class select **'Update classes and resources'** (not all class changes are reloadable, if the class cannot be reloaded you will see a failure notification).

Once the dictionary has been reloaded you should see the following output in the console (note if there are problems in the updated configuration you will see error messages instead).

```
2013-01-21 12:19:46,978 [Timer-2] u:/d: INFO
org.kuali.rice.krad.datadictionary.DataDictionary - reloading dictionary
configuration for Book.xml

…

2013-01-21 12:19:47,902 [Thread-23] u:/d: INFO
org.kuali.rice.krad.datadictionary.uif.UifDictionaryIndex - Starting View Index
Building

2013-01-21 12:19:48,487 [Thread-23] u:/d: INFO
org.kuali.rice.krad.datadictionary.uif.UifDictionaryIndex - Completed View
Index Building

2013-01-21 12:19:48,597 [Thread-22] u:/d: INFO
org.kuali.rice.krad.datadictionary.DataDictionaryIndex - Completed DD Index
Building

2013-01-21 12:19:48,598 [Thread-22] u:/d: INFO
org.kuali.rice.krad.datadictionary.DataDictionaryIndex - Started DD
Inactivation Blocking Index Building

2013-01-21 12:19:48,668 [Thread-22] u:/d: INFO
org.kuali.rice.krad.datadictionary.DataDictionaryIndex - Completed DD
Inactivation Blocking Index Building
```