Tutorial 1.1: Clojure Environment Setup with IntellijIDEA

1. *What is clojure ?*

Using the definition taken from the site:
"Clojure is a dialect of Lisp, and shares with Lisp the code-as-data philosophy and a powerful macro system. Clojure is predominantly a functional programming language, and features a rich set of immutable, persistent data structures. When mutable state is needed, Clojure offers a software transactional memory system and reactive Agent system that ensure clean, correct, multithreaded designs."

Clojure features:
http://clojure.org/features

2. *What is code-as-data ?*

Internet in plenty of definitions for this. Basically it means the program structure is similar to its syntax, and therefore the program's internal representation can be inferred by reading the text's layout.

3. Why should I use clojure ?

a. If you are familiar with Lisp, it will be easy for you to get started with Clojure.
b. Allows huge functionality in a few lines.
c. Java interoperability.
d. Runs on the JVM, so you get benefits from all its features including memory management, security and stability.
e. For sure you will find more advantages after you work with it!

4. Setting your Clojure development environment:

You might use Emacs to develop your Clojure projects or even any text editor. However, on this tutorial we will use intellijIDEA to assist you in your clojure projects.

In this tutorial, we will install and configure:

a. JDK 1.7.0_80
b. IntellijIDEA 14.0.4 Community Edition
c. Leiningen

d. Cursive Clojure plugin
e. Leiningen immutant plugin
f. WildFly 8.2.1
g. H2 database 1.3.176

a. Installing JDK 1.7.0_80

Download and Install the Java SE Development Kit 7u80 from here.

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

b. Installing IntelljIDEA

We need to download IntellijIDEA, download version 14.0.4 of the community edition.

https://confluence.jetbrains.com/display/IntelliJIDEA/Previous+IntelliJ+IDEA+Releases

c. Install Leiningen (https://github.com/technomancy/leiningen).

Leiningen is a tool for automating/managing your projects in clojure, it comes with a series of template projects.

For Unix/Linux users :

1. Download the lein script from the link below.
   https://raw.githubusercontent.com/technomancy/leiningen/stable/bin/lein
2. Place it on your $PATH. (~/bin is a good choice if it is on your path)
3. Set it to be executable. (chmod 755 ~/bin/lein)

For Windows users:

1. You just need to download and install leiningen from this link.

http://leiningen-win-installer.djpowell.net/

After the steps above, for All users:
To check if the Leiningen installation worked, just type on your console : *lein help* you should see something like this.

```
cnb@NeXT:~ % lein help
Leiningen is a tool for working with Clojure projects.

Several tasks are available:
change              Rewrite project.clj by applying a function.
```

```
check               Check syntax and warn on reflection.
classpath           Print the classpath of the current project.
clean               Remove all files from project's target-path.
compile             Compile Clojure source into .class files.
deploy              Build and deploy jar to remote repository.
deps                Download all dependencies.
do                  Higher-order task to perform other tasks in succession.
help                Display a list of tasks or help for a given task.
immutant            Tasks for managing Immutant 2.x projects in a WildFly container.
install             Install the current project to the local repository.
jar                 Package up all the project's files into a jar file.
javac               Compile Java source files.
new                 Generate project scaffolding based on a template.
plugin              DEPRECATED. Please use the :user profile instead.
pom                 Write a pom.xml file to disk for Maven interoperability.
release             Perform :release-tasks.
repl                Start a repl session either with the current project or
standalone.
retest              Run only the test namespaces which failed last time around.
run                 Run a -main function with optional command-line arguments.
search              Search remote maven repositories for matching jars.
show-profiles       List all available profiles or display one if given an argument.
test                Run the project's tests.
trampoline          Run a task without nesting the project's JVM inside Leiningen's.
uberjar             Package up the project files and dependencies into a jar file.
update-in           Perform arbitrary transformations on your project map.
upgrade             Upgrade Leiningen to specified version or latest stable.
vcs                 Interact with the version control system.
....
```

The immutant task is the one we will use later on the workshop and also the new.

d. Cursive Clojure Plugin (https://cursiveclojure.com/)

Cursive is currently an IntellijIDEA plugin so it leverages all the functionalities available in the jetbrain's IDE.

Cursive is built on IntelliJ, the most sophisticated Java IDE. Cursive contains all the functionality you've come to expect from JetBrains products, from project management to version control integrations across all platforms. Building on IntelliJ also provides in-editor inspections and seamless Java integration.

Built in Clojure

Cursive is written (almost) entirely in Clojure, allowing us to easily integrate all the fantastic tooling in the Clojure ecosystem (for example, Leiningen and nREPL). Cursive is developed with Cursive, so we want it to be the best Clojure development environment around!

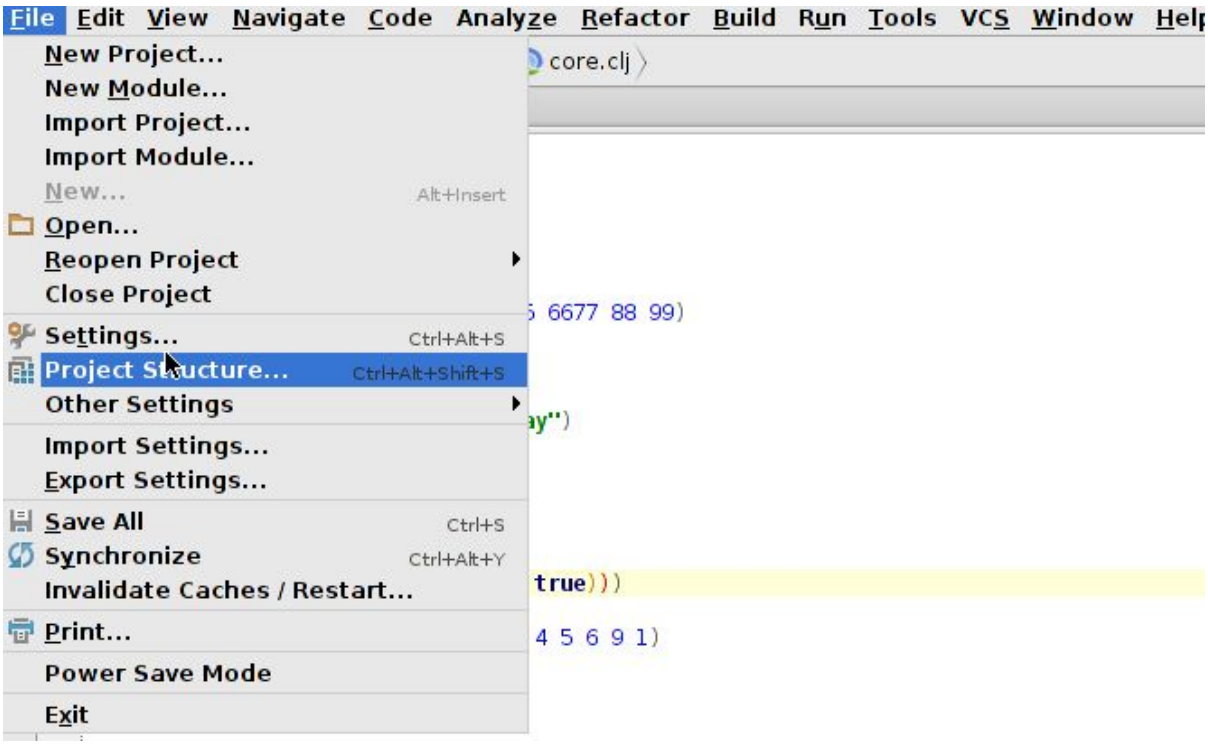
Cursive provides...

- Syntax Highlighting
- Navigation (Jump to symbol, Find Usages)
- Symbol renaming
- nREPL based REPL
- Leiningen support
- Paredit-style structural editing
- Code formatting
- A symbolic debugger
- Solid integration with Java for mixed projects
- All standard IntelliJ features (project management, VCS etc)

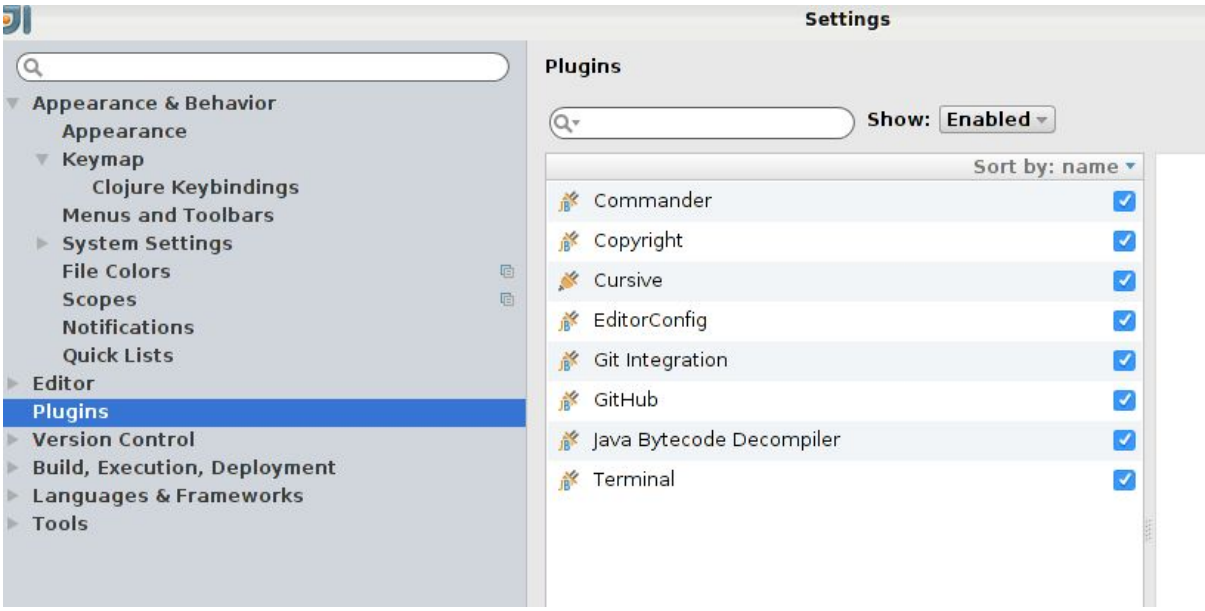
Configure IntelljIDEA with Cursive Clojure plugin

Now that you have downloaded and installed I*ntellijIDEA,* let's start it and configure it for using the cursive plugin.

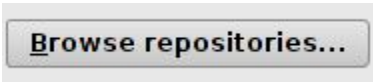
i. I suggest to disable unneeded plugins as follows. Click on File then on settings as described in the image.



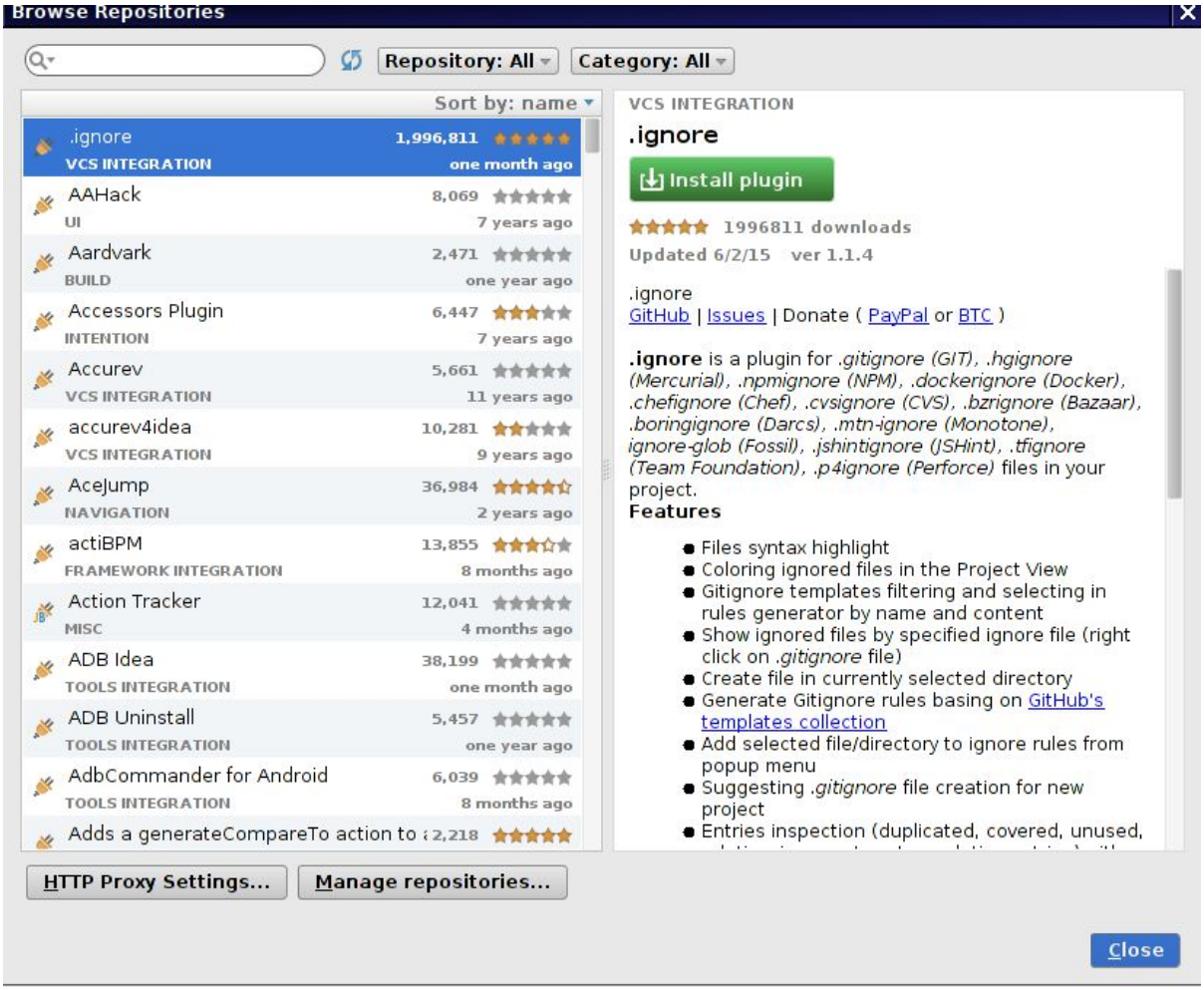
Now go to plugins, and disable all except the following ones :

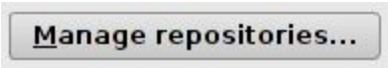


ii. Then, Install Cursive Clojure plugin:

Again go to plugins and click on the 

button that is located at the bottom right. That will take you to this screen.

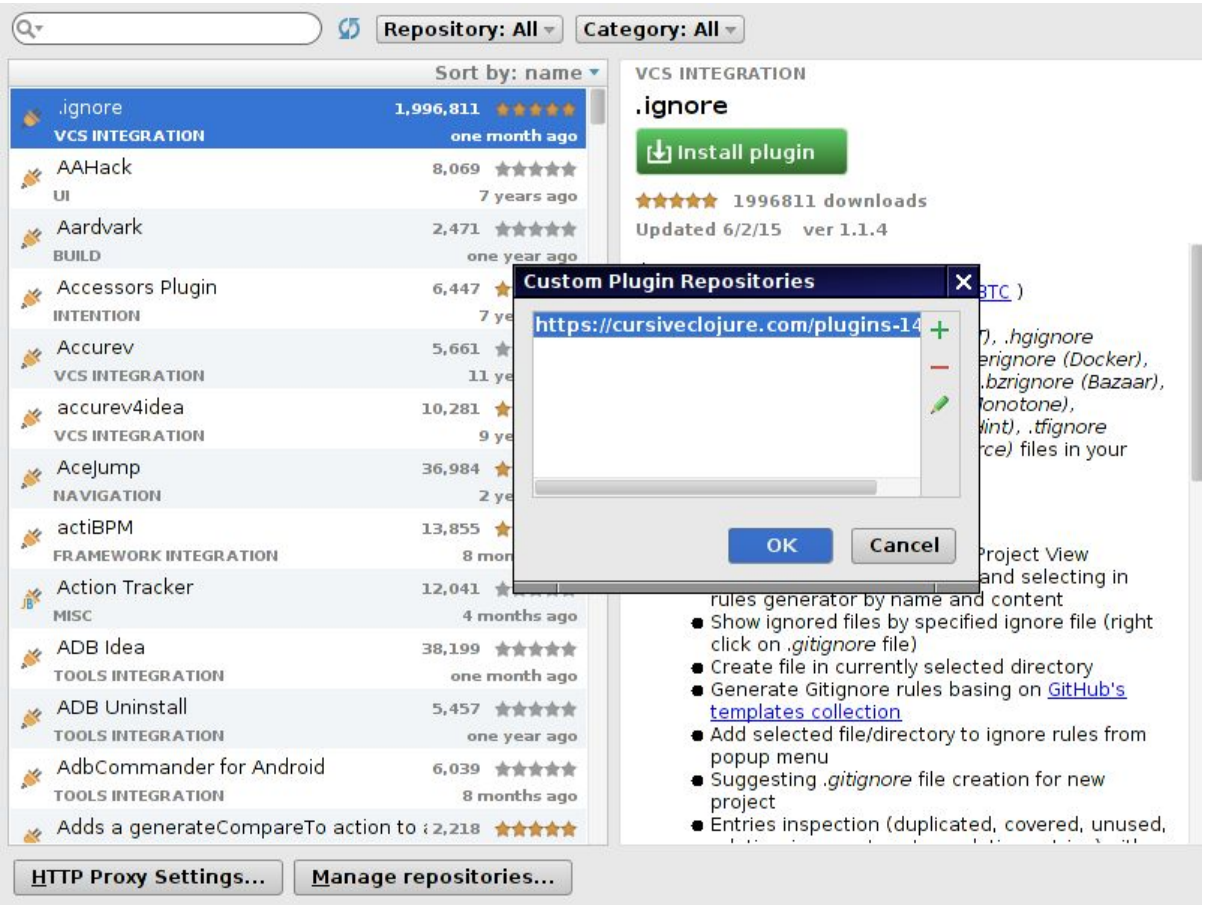


Then Click on 

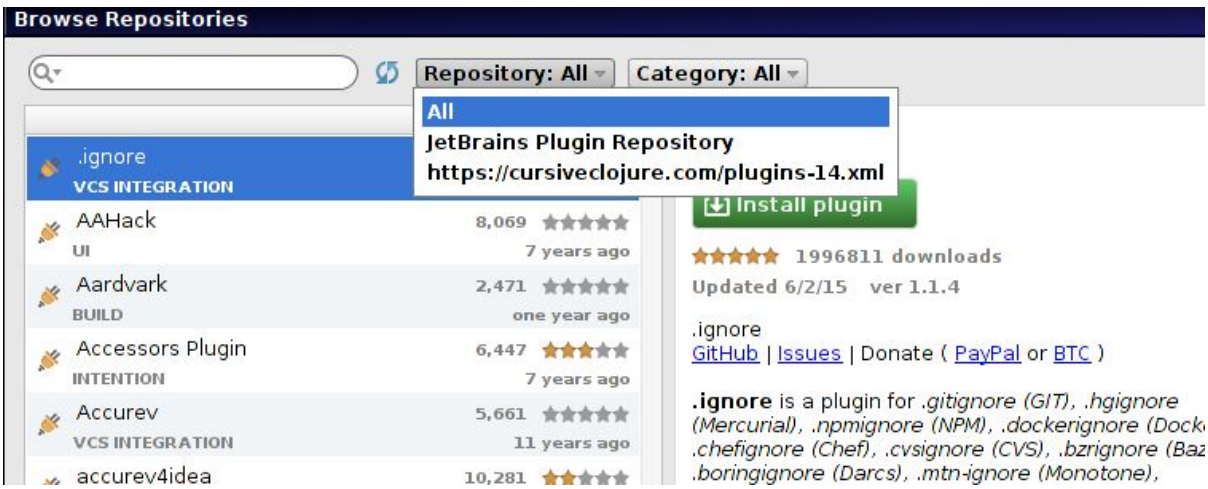
and add the following URL

https://cursiveclojure.com/plugins-14.xml

clicking on the  button, as it is presented in the following screen.



Now choose the new repository we have added (https://cursiveclojure.com/plugins-14.xml)



After that the cursive plugin will appear as the only one in the list of available plugins.

Click on the 

button located at the right panel, this will start the installation then *intellijIDEA* will ask you to restart the IDE after that the plugin will be available.

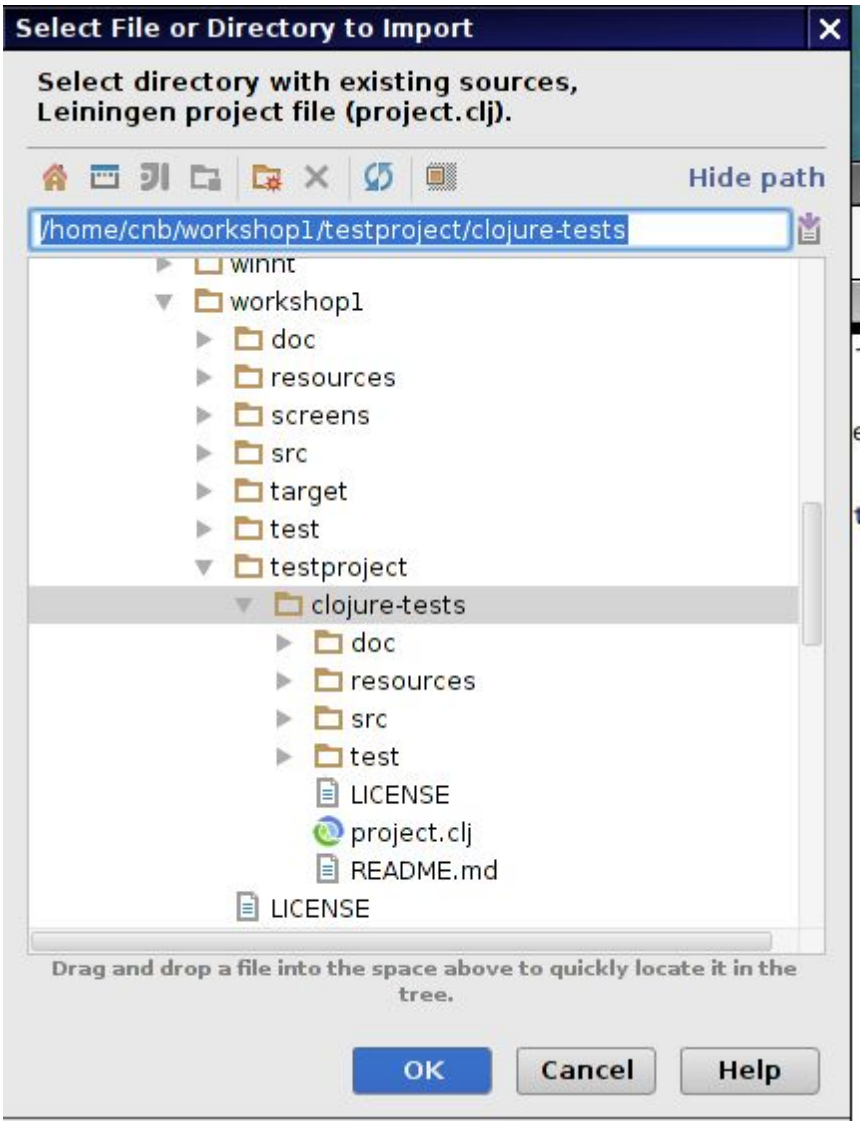
iii. Testing the plugin:

Let's create a sample clojure project using Leiningen, this is done using the new task and selecting one of the available templates for this in our case we will use the app template.
I will create a new clojure app project named clojure-tests:

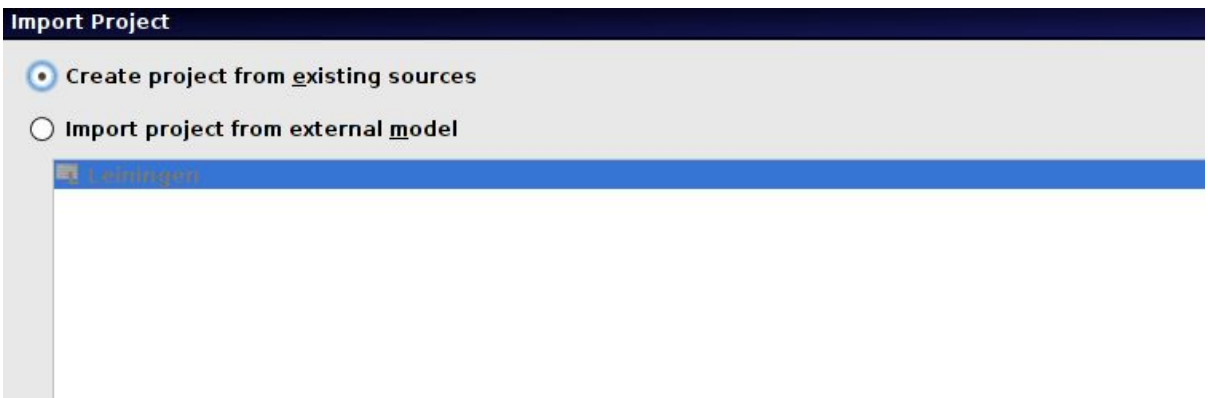
```
cnb@NeXT:~/workshop1/testproject % lein new app clojure-tests
Generating a project called clojure-tests based on the 'app' template.
```

Now we are ready to import this project using intellijIDEA. Let's click on **Import Project**

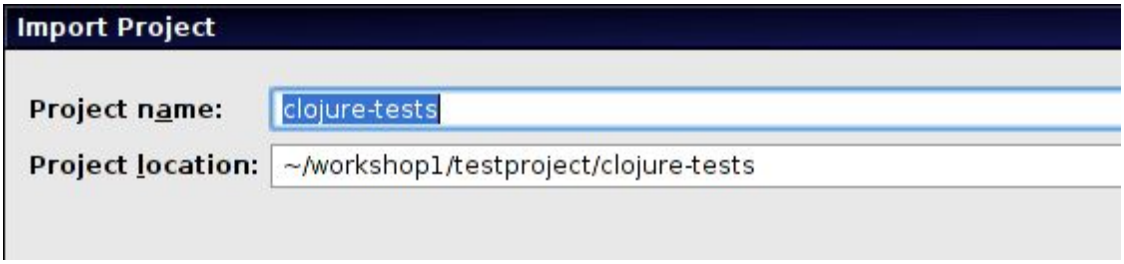
And then simply look where we had saved the leiningen created project in our case clojure-tests.



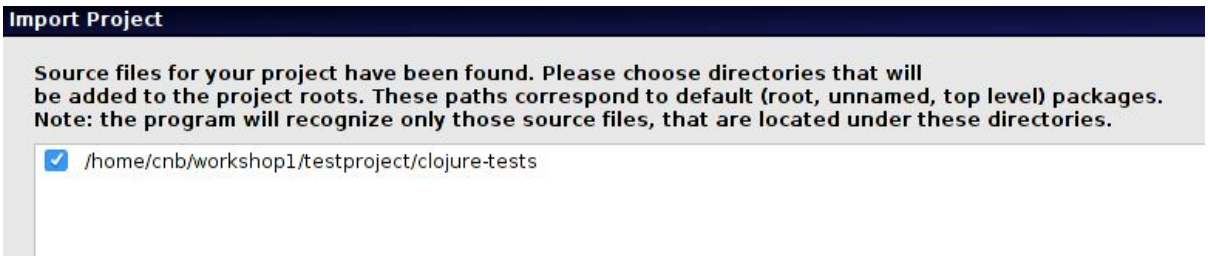
Choose ***Create Project from existing sources***



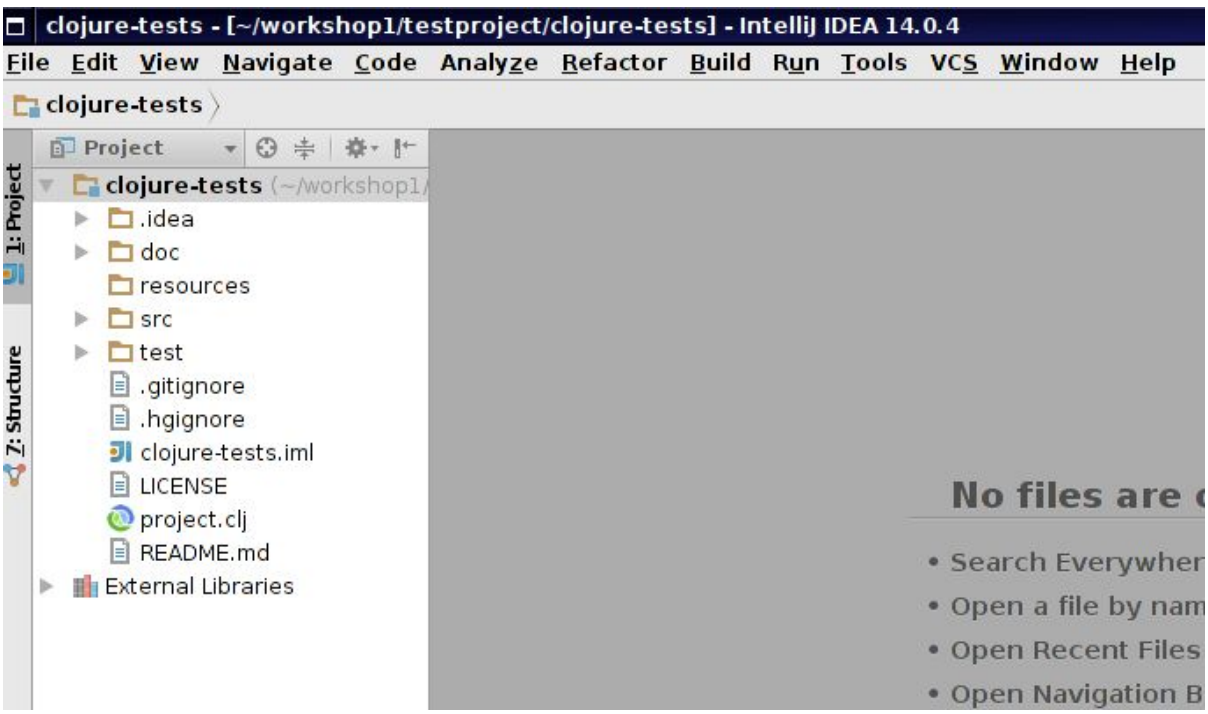
Choose a name for your project I have named mine using the same name as the folder.



By default it is selected leave it as it is. This will add to our project all the sources found on the location we specified when importing a project.

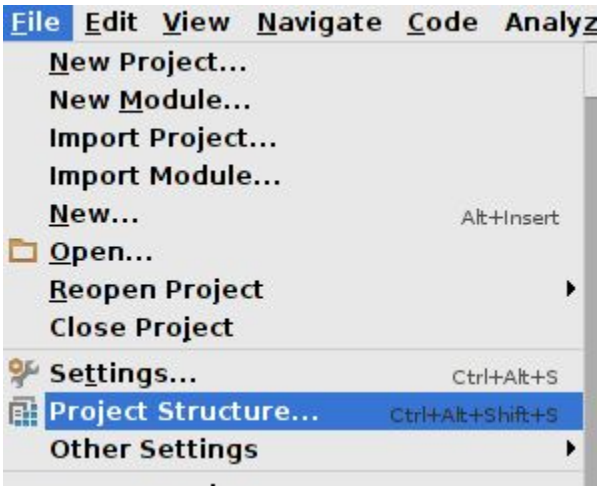


Finally your project will be imported into *IntellijIDEA*, if you see the clojure logo  on the project.clj file created by leiningen it means the cursive clojure has been loaded.

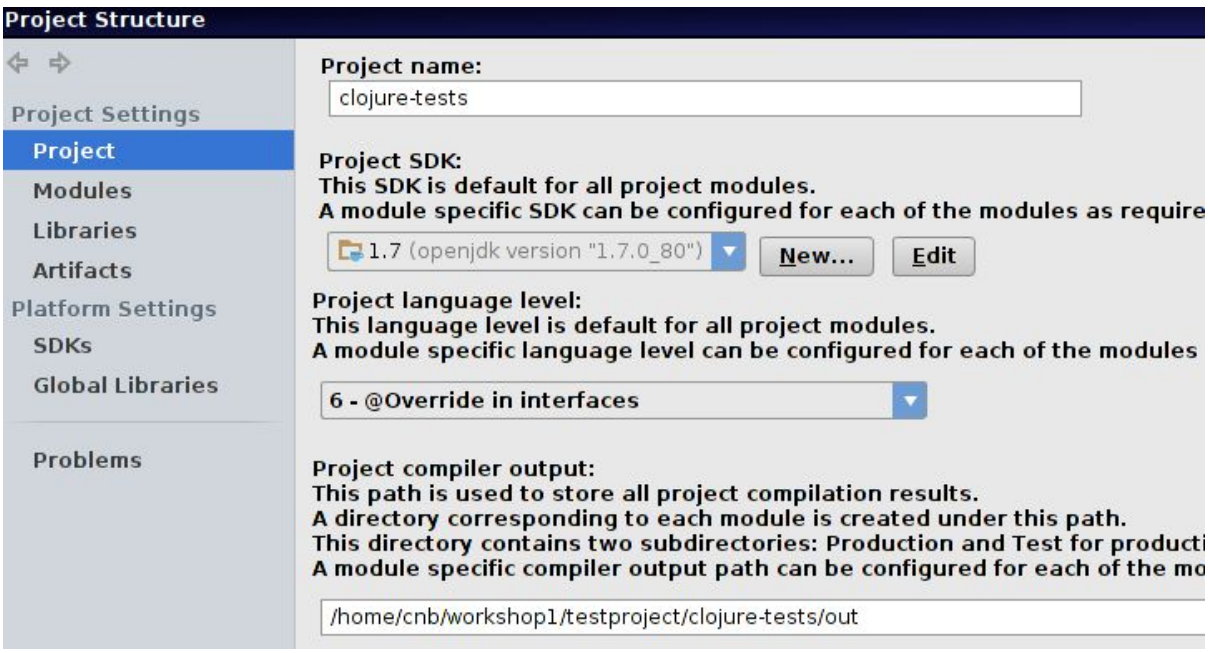


iv. Now we need to configure our project in order to use cursive.

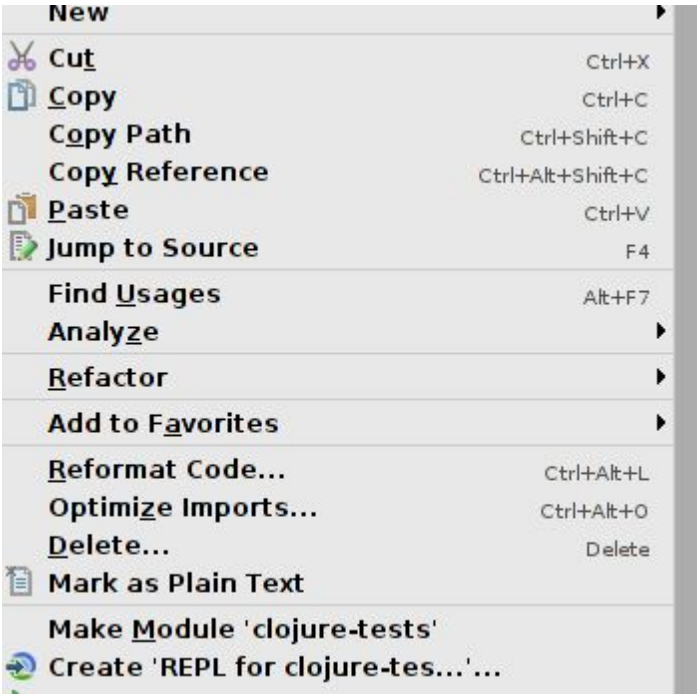
First, let's go to Project structure



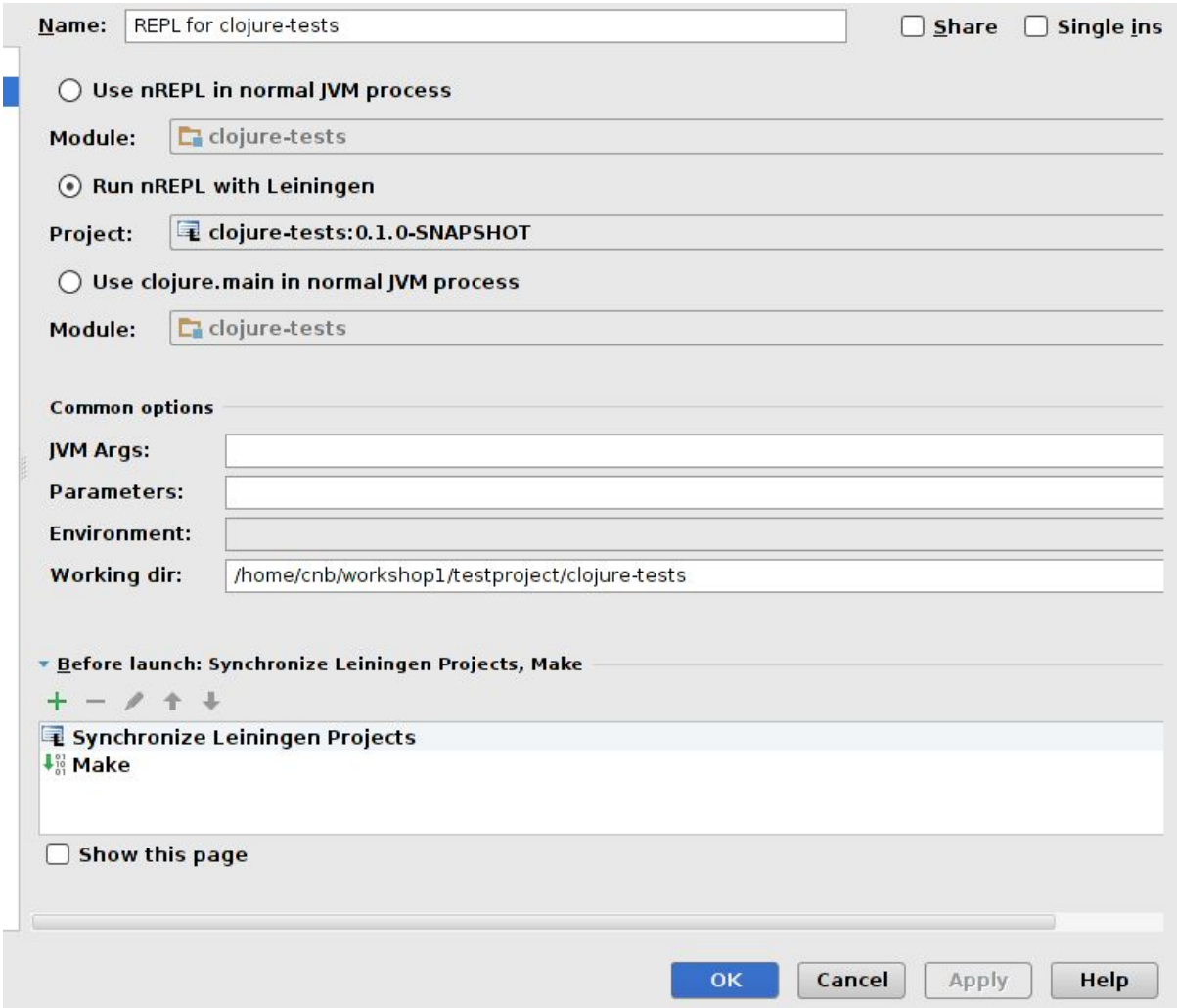
Make sure the Project SDK points to the java version we have installed.



After this is setup you can create/configurate your repl. this is done following the next steps.
Right Click on the project.clj file on your project and select Create 'REPL for clojure-test..'



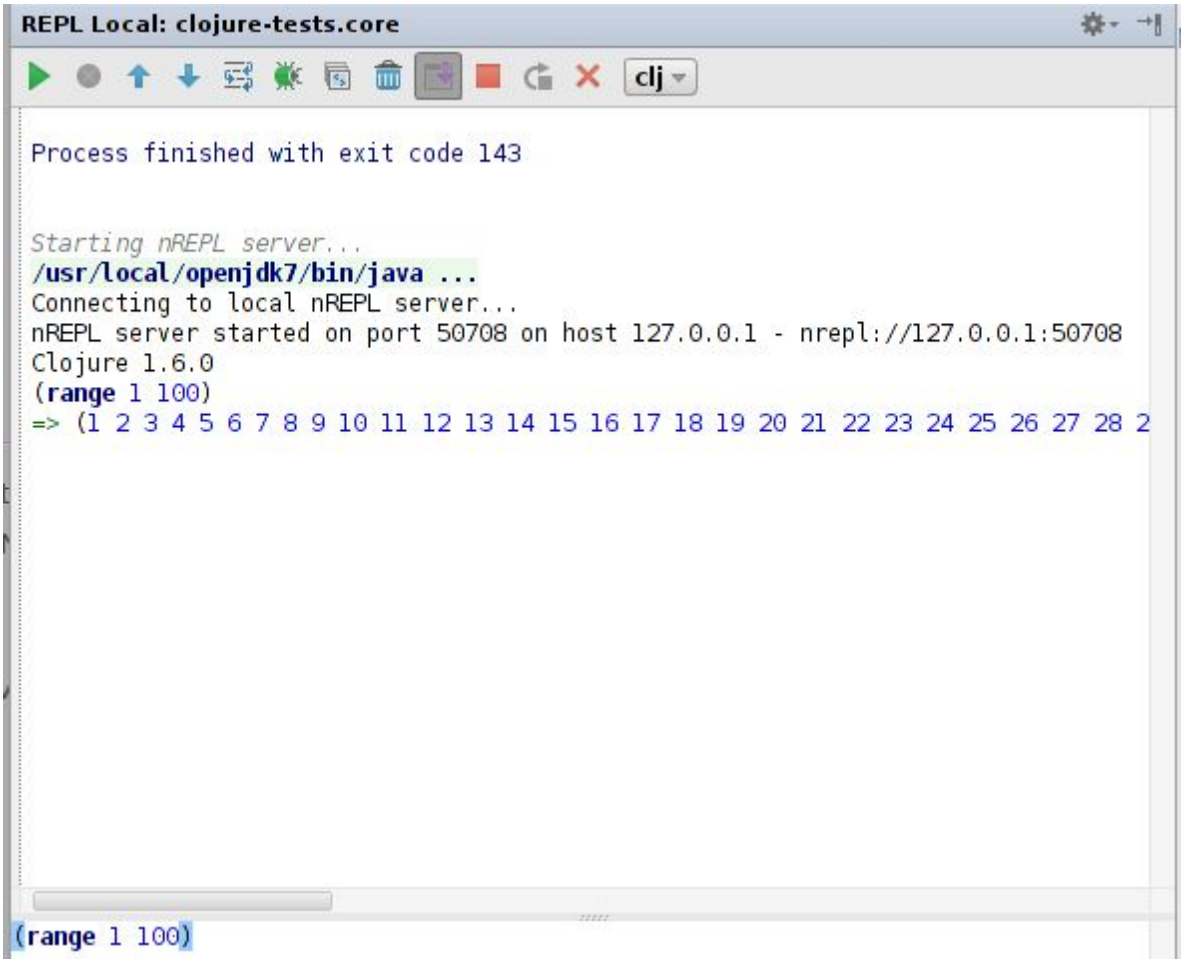
In the next screen choose ***Run nREPL with Leiningen***:



Now this button 

should be visible clicking on it will start our repl.

Now we can start interacting with our repl in this window.



We have finished installing cursive in intelliJIDEA now we are ready to start coding our app.

e. Installing Inmutant Leiningen plugin

This link below includes everything you need.
https://github.com/immutant/lein-immutant

Basically, just need to add to your .lein/profiles.clj file the following plugin, if you do not have a profiles.clj file just created under .lein/profiles.clj.

```
{:user {:plugins [[lein-immutant "2.0.0"]]}}
```

This will allow us to create war files and deploy them into a WildFly container.

f. Installing Wildfly

Download version 8.2.1 Final from http://wildfly.org/downloads/
Uncompress the archive and go to the folder bin in the wildfly folder and execute the add-user script answer the following questions with the values chosen as this example:

```
cnb@NeXT:~/wildfly-8.2.1.Final/bin % ./add-user.sh

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : cnb
Password recommendations are listed below. To modify these restrictions edit the
 add-user.properties configuration file.
 - The password should not be one of the following restricted values {root, admi
n, administrator}
 - The password should contain at least 8 characters, 1 alphabetic character(s),
 1 digit(s), 1 non-alphanumeric symbol(s)
 - The password should be different from the username
Password :
JBAS015269: Password must have at least 8 characters!
Are you sure you want to use the password entered yes/no? yes
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated
list, or leave blank for none)[  ]:
About to add user 'cnb' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'cnb' to file '/usr/home/cnb/wildfly-8.2.1.Final/standalone/configura
tion/mgmt-users.properties'
Added user 'cnb' to file '/usr/home/cnb/wildfly-8.2.1.Final/domain/configuration
/mgmt-users.properties'
Added user 'cnb' with groups  to file '/usr/home/cnb/wildfly-8.2.1.Final/standal
one/configuration/mgmt-groups.properties'
Added user 'cnb' with groups  to file '/usr/home/cnb/wildfly-8.2.1.Final/domain/
configuration/mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS pr
ocess?
e.g. for a slave host controller connecting to the master or for a Remoting conn
ection for server to server EJB calls.
yes/no? yes
```

Now let's start wildfly, let's go to the bin folder again and execute it with the following parameter -c standalone-full.xml, the -c flag specifies a config file that wildfly will use to start. You should see something like this :

```
cnb@NeXT:~/wildfly-8.2.1.Final/bin % ./standalone.sh  -c standalone-full.xml
=========================================================================

  JBoss Bootstrap Environment

  JBOSS_HOME: /home/cnb/wildfly-8.2.1.Final
```

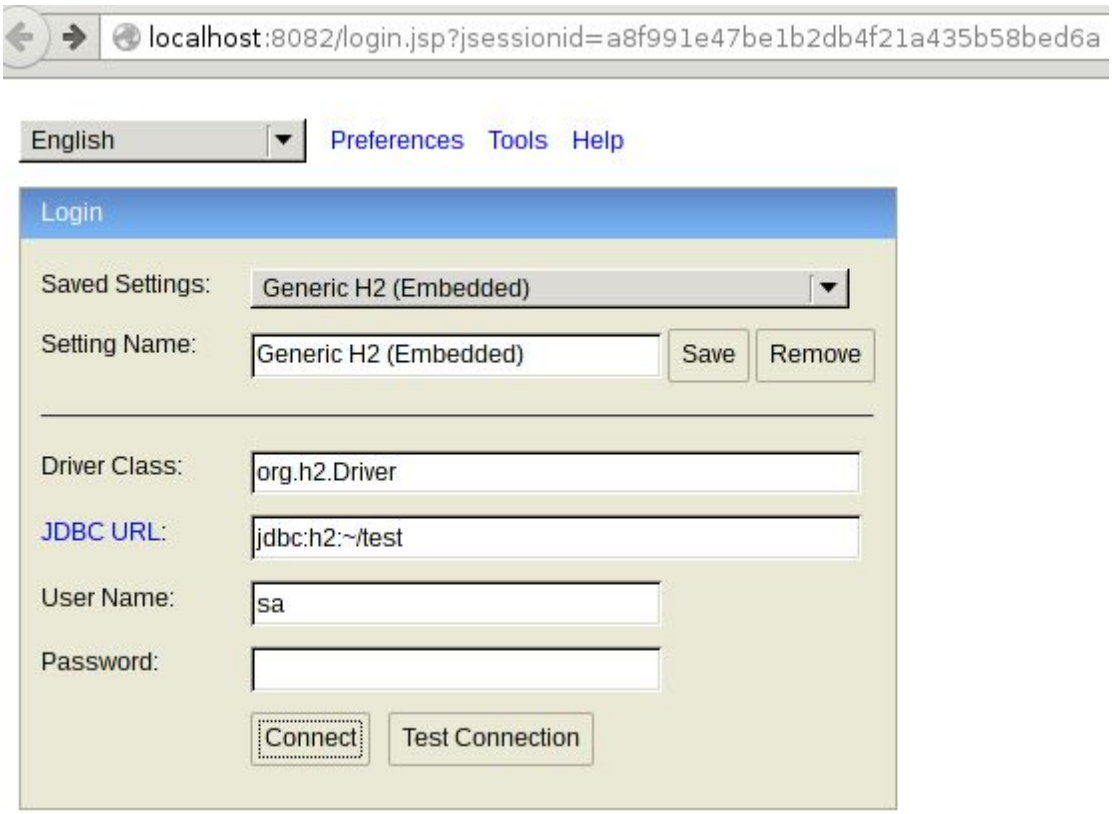
```
  JAVA: java

  JAVA_OPTS:  -server  -server -Xms64m -Xmx512m -XX:MaxPermSize=256m
-Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman
-Djava.awt.headless=true
```

g. H2 Database

We need version 1.3.176 which is available from here:
http://www.h2database.com/html/download.html

Just unzip, and go to the bin folder and execute the h2 script to make sure all is working. Without options, it will open your default browser and show the console.



We have installed everything we need to start our development using clojure and wildfly!

This material is property of codemissions.com
Author: Carlos Antonio Neira Bustos
Email : cneirabustos@gmail.com