BDD in a Day

A Hands-on Workshop for Programmers

Introduction

Welcome. These instructions should help you get set up for the BDD in a Day workshop. They cover three things:

1) Installing and configuring the necessary software
2) Loading the project into Eclipse
3) Verifying that you are set up and ready to go

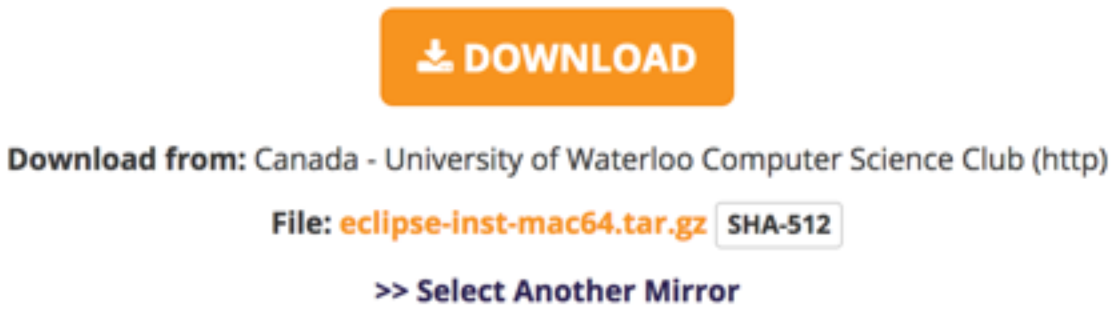
Software Set Up

Install NodeJS and npm

Use the online instructions at https://nodejs.org/en/download/ to install NodeJS and npm on your computer.

Install Spring - Oxygen

1) Open your browser, navigate to https://www.eclipse.org/oxygen/, and click on the "Download" button
2) The next page will let you choose the source (mirror) from which you want to download Eclipse. If you don't care then click the orange "DOWNLOAD" button and the download will begin. If you want to see if there's a mirror closer to you, then click on ">>Select another mirror" and a list will be displayed. After you've chosen one, the popup will disappear and you'll be back here with a different location displayed after the "Download from:" label. Click the "DOWNLOAD" button and wait for your file to download.



Download from: Canada - University of Waterloo Computer Science Club (http)

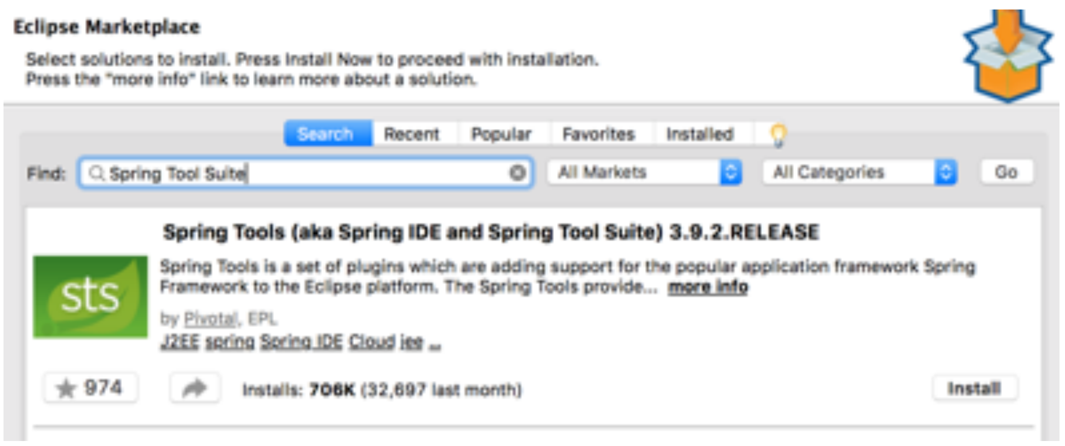
File: eclipse-inst-mac64.tar.gz SHA-512

3) After the file has downloaded, follow the normal installation instructions for your operating system then run the program, which should be the Eclipse Installer.
4) When the Installer finishes loading, select "Eclipse IDE for Java Developers" then click the "INSTALL" button. This start the actual download of the actual Eclipse program.

5) Once Eclipse has finished downloading, follow your normal installation instructions to install it, then run it so you can get on to the next set of tasks.

Install the Spring Tool Suite plugins:

1) From the menu bar, select Help / Eclipse Marketplace…
2) Enter “Spring Tool Suite” in the “Find” input field and hit the “Enter/Return” key.


Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: Spring Tool Suite

Spring Tools (aka Spring IDE and Spring Tool Suite) 3.9.2.RELEASE

Spring Tools is a set of plugins which are adding support for the popular application framework Spring Framework to the Eclipse platform. The Spring Tools provide... more info

by Pivotal, EPL

J2EE spring Spring IDE Cloud iee ...

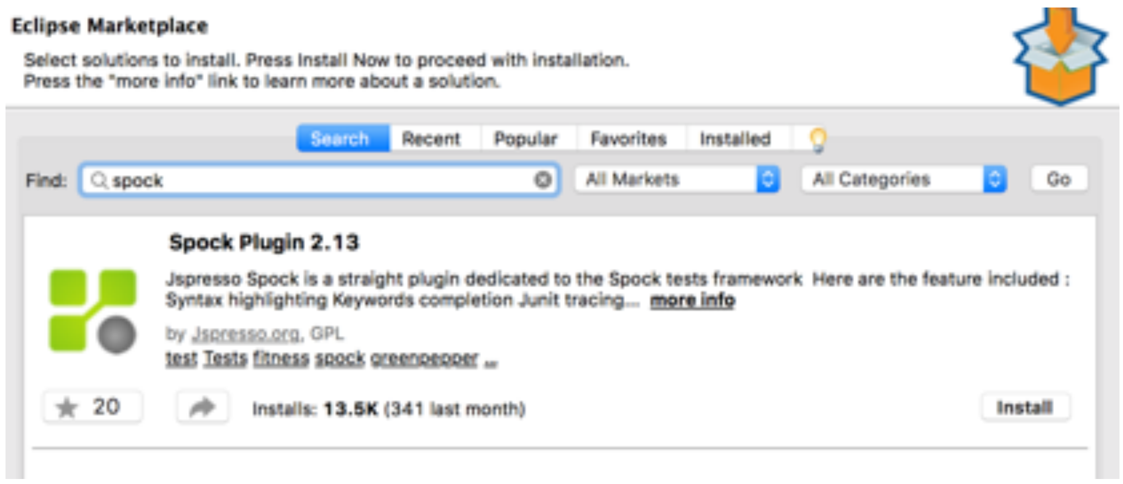
974

Installs: 706K (32,697 last month)

3) Click the “Install” button then accept the license agreement. You’ll get a warning that the plugin contains unsigned content. Click “Install anyway” to continue.
4) Note: After the dialog box disappears, you’ll see the message “Installing Software” at the bottom right of the Eclipse window followed by the current completion percentage. Once the installation completes, Eclipse will display a popup telling you it needs to restart for the changes to take effect. Click “Restart Now” to continue.

Install the Jspresso Spock Plugin

1) From the menu bar, select Help / Eclipse Marketplace…
2) Type “spock” into the “Find” input field and hit the “Enter/Return” key


Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: spock

Spock Plugin 2.13

Jspresso Spock is a straight plugin dedicated to the Spock tests framework Here are the feature included : Syntax highlighting Keywords completion Junit tracing... more info

by Jspresso.org, GPL

test Tests fitness spock greenpepper ...

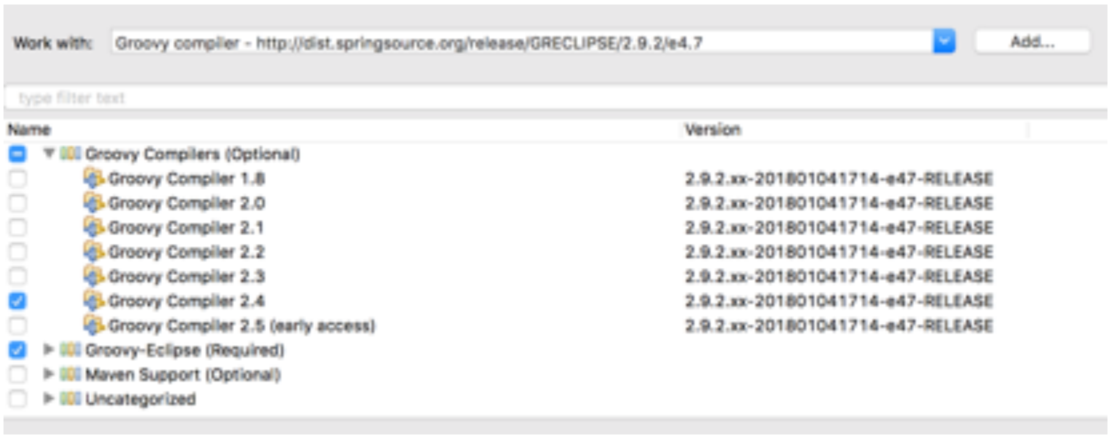
20

Installs: 13.5K (341 last month)

3) Click the “Install” button for the Spock Plugin by espresso.org, and accept the license agreement. You’ll get a warning that the plugin contains unsigned content. Click “Install anyway” to continue.
4) Repeat Eclipse’s “Installing Software / Restart” process.

Install the Groovy 2.4 compiler

1) From the menu bar, select Help / Install new software… then, in the “Work with” text field enter the following URL: http://dist.springsource.org/release/GRECLIPSE/2.9.2/e4.7 and click the “Add” button.
2) Click on the triangle to the left of the top item “Groovy compilers (Optional)” then click on the check box for “Groovy Compiler 2.4” and the check box for Groovy-Eclipse (Required).



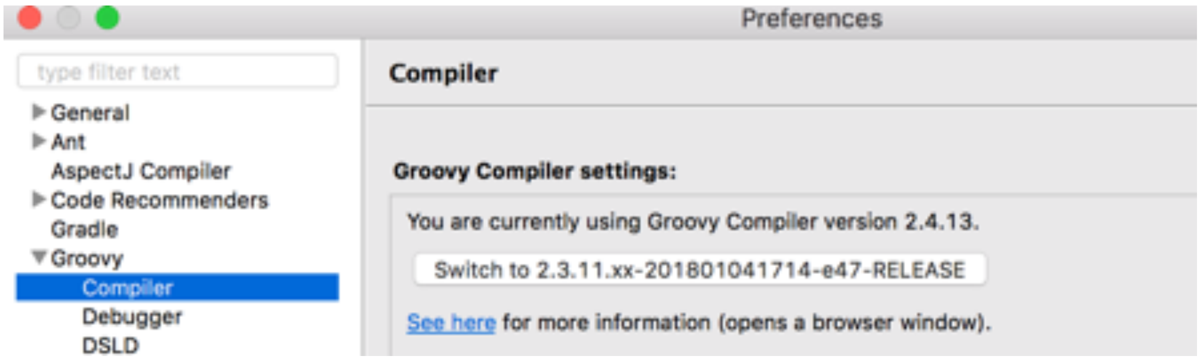
Work with: Groovy compiler - http://dist.springsource.org/release/GRECLIPSE/2.9.2/e4.7

	Name	Version
▣	▼ Groovy Compilers (Optional)	
☐	Groovy Compiler 1.8	2.9.2.xx-201801041714-e47-RELEASE
☐	Groovy Compiler 2.0	2.9.2.xx-201801041714-e47-RELEASE
☐	Groovy Compiler 2.1	2.9.2.xx-201801041714-e47-RELEASE
☐	Groovy Compiler 2.2	2.9.2.xx-201801041714-e47-RELEASE
☐	Groovy Compiler 2.3	2.9.2.xx-201801041714-e47-RELEASE
✓	Groovy Compiler 2.4	2.9.2.xx-201801041714-e47-RELEASE
☐	Groovy Compiler 2.5 (early access)	2.9.2.xx-201801041714-e47-RELEASE
✓	▶ Groovy-Eclipse (Required)	
☐	▶ Maven Support (Optional)	
☐	▶ Uncategorized	

3) Click on the “Next >” button at the bottom, then follow the instructions to complete the installation.

Software Configuration

After Eclipse reboots, set the default Groovy compiler to version 2.4:
1) From the menubar, select Eclipse / Preferences…
2) On the Preferences dialog box, click the triangle next to “Groovy”, then click on “Compiler”. at the top of the righthand section, under “Groovy Compiler Settings”, you should see something like “You are currently using Groovy Compiler version 2.4.13”



Compiler

Groovy Compiler settings:

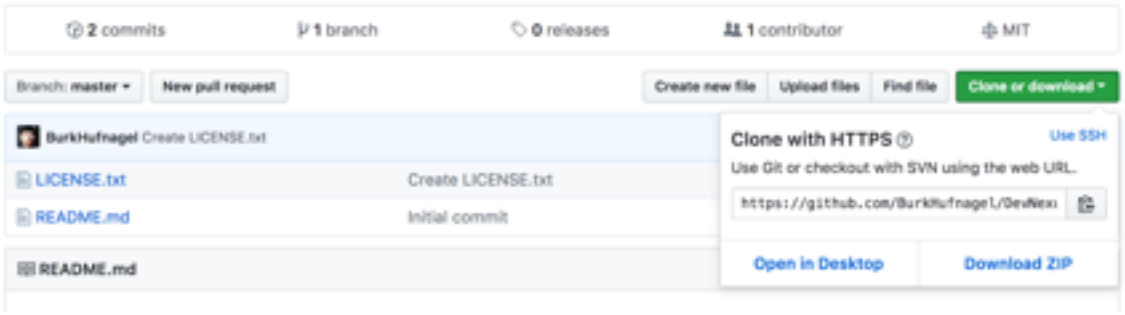
You are currently using Groovy Compiler version 2.4.13.

See here for more information (opens a browser window).

3) Click the “Cancel” button at the bottom of the dialog box to close it.

Download the project from GitHub

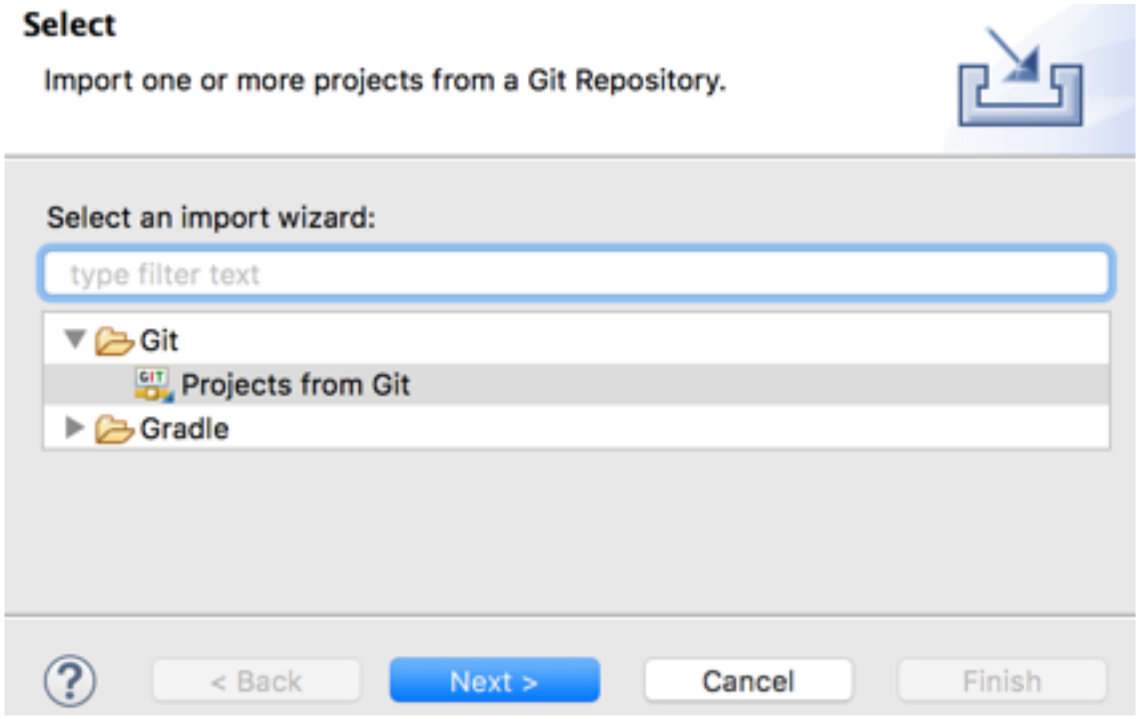
1) Open your browser and navigate to https://github.com/BurkHufnagel/DevNexus-2018-BDD-in-a-Day. Click on the green “Clone or Download button” to display the URL, then click on the clipboard icon to copy it to your clipboard.



Use Git or checkout with SVN using the web URL.

https://github.com/BurkHufnagel/DevNex

2) Open Eclipse and select File / Import to bring up the Import dialog box. From the list select “Git” then “Projects from Git” and click the “Next >” button to continue.



Import one or more projects from a Git Repository.

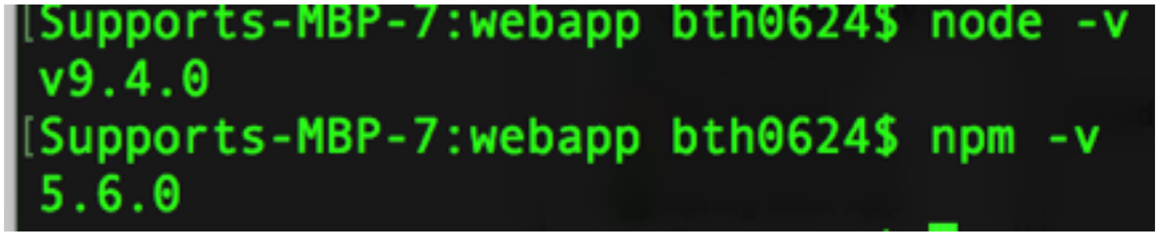
Select an import wizard:

3) Select “Clone URI” and click “Next >” and it should display the“Import Projects from Git” page with all of the information needed to download a repository. If the information is blank, or for a different repository, then use the keyboard shortcut to paste the information copied to the clipboard back in step 1 into the fields. Note: If you’ve never done this before you”ll need to enter your GitHub User and Password information.

4) Clicking the “Next >” button will take you to the “Branch Selection” page. Click the “Next >” button, change the “Destination Directory” if you don’t like the default, then click “Finish” and Eclipse will create a local copy of the repository in that directory.
5) If it’s not already open, select Window / Show View / Other… to bring up the “Show View” dialog box. Click the triangle to the left of Git then select “Git Repositories” and click the “Open” button.
6) From the “Git Repositories” window, right click the “DevNexus-2018-BDD-in-a-Day” entry and click on the “Import Projects…” entry of the context menu. After Eclipse finishes going through the directory and finds a folder to import, click on the “Finish” button and Eclipse will load the project which should be visible in the “Package Explorer” or “Navigator” window.

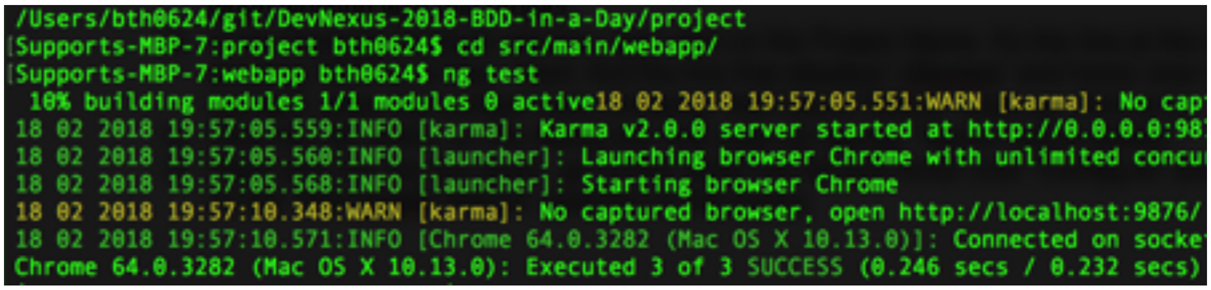
Run the unit test for the Angular 5 client

1) From the command line, execute the “node -v” and “npm -v” commands to verify that both of them are installed and the version numbers are the same or higher than the ones in the screenshot.



```
[Supports-MBP-7:webapp bth0624$ node -v
v9.4.0
[Supports-MBP-7:webapp bth0624$ npm -v
5.6.0
```

2) Now navigate to the directory into which you download the project. From there, navigate to the “src/webapp” directory and execute the command “ng test” which invokes the karma test runner and runs unit tests against the Angular client. You should see something similar to the following screenshot.



```
/Users/bth0624/git/DevNexus-2018-BDD-in-a-Day/project
[Supports-MBP-7:project bth0624$ cd src/main/webapp/
[Supports-MBP-7:webapp bth0624$ ng test
 10% building modules 1/1 modules 0 active18 02 2018 19:57:05.551:WARN [karma]: No cap
18 02 2018 19:57:05.559:INFO [karma]: Karma v2.0.0 server started at http://0.0.0.0:98
18 02 2018 19:57:05.560:INFO [launcher]: Launching browser Chrome with unlimited concu
18 02 2018 19:57:05.568:INFO [launcher]: Starting browser Chrome
18 02 2018 19:57:10.348:WARN [karma]: No captured browser, open http://localhost:9876/
18 02 2018 19:57:10.571:INFO [Chrome 64.0.3282 (Mac OS X 10.13.0)]: Connected on socke
Chrome 64.0.3282 (Mac OS X 10.13.0): Executed 3 of 3 SUCCESS (0.246 secs / 0.232 secs)
```

3) The important part is on the last line of the screenshot, where it says, “Executed 3 of 3 SUCCESS” indicating that all the tests passed.

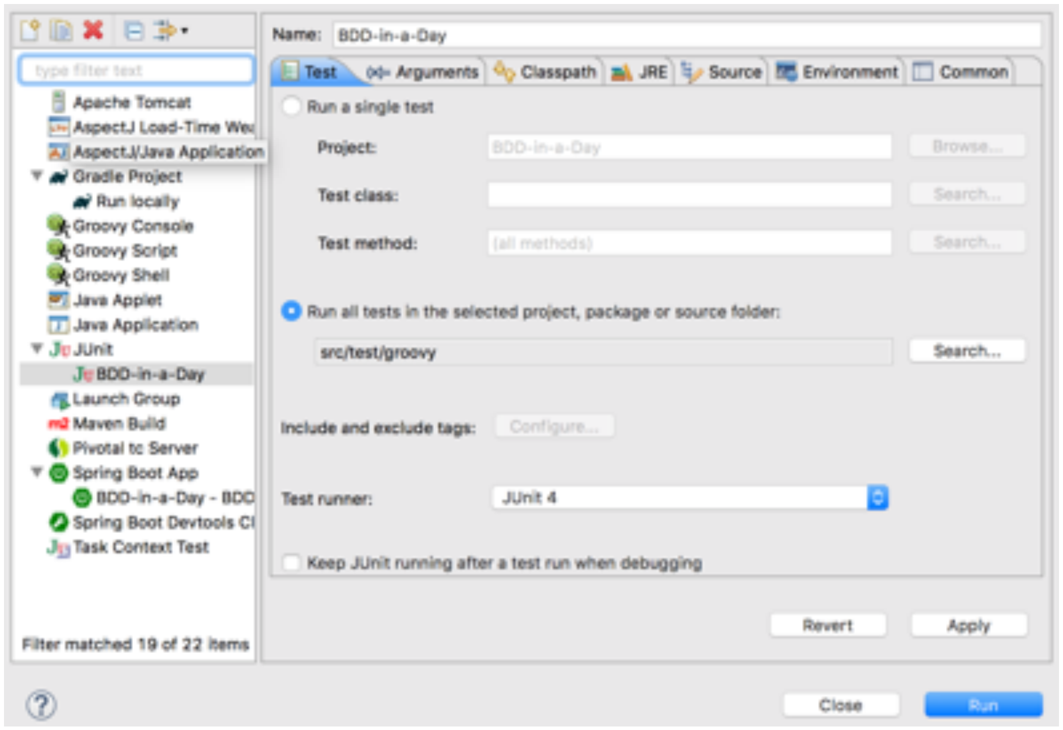
Adding Gradle and Spock nature to the Eclipse project

1) Adding the Spock and Cradle nature tells Eclipse that we want to use Gradle and Spock when working with this project.
2) In the Package Explorer window, right click on the Project Name. It’s the line at the top showing “BDD-in-a-Day”. Next, find the line that displays “JSpreso” and hover over it for a moment, when the context menu showing “Add Spock Nature” pops up, click on it.

3) Click the OK button on the dialog box that pops up saying, “Spock nature added.”
4) Right click the top line again, go further down the list and hover over “Configure” and click on the “Add Gradle Nature” selection in the popup.

Configure unit, integration, and functional tests

1) Select Run / Run Configurations… from the menu bar and Eclipse will display the “Run Configurations” dialog box.



Name: BDD-in-a-Day

Run a single test

Project: BDD-in-a-Day

Test class:

Test method: (all methods)

Run all tests in the selected project, package or source folder:

src/test/groovy

Include and exclude tags:

Test runner: JUnit 4

Keep JUnit running after a test run when debugging

2) Click on the JUnit entry on the left side of the dialog box then, at the top left of that area, click on the icon showing a white rectangle with a yellow plus sign on its upper right corner and it will create an entry under JUnit with the name “BDD-in-a-Day”
3) In the name field at the top of the right-hand pane, change the name to “BDD-in-a-Day Unit Tests”, then click on the white “Search…” button and, from the popup “Folder Selection” dialog box select the “src/test/groovy” and click OK, then change the Test Runner from JUnit 3 to JUnit 4.
4) You dialog box should look similar to the one above. Click “Apply” button to save the changes you made to the “BDD-in-a-Day” configuration.
5) Repeat steps two, three, and four to create a “BDD-in-a-Day IntegrationTest” and “BDD-in-a-Day Functional” configuration, selecting the appropriate directory for each of them in the “Folder Selection” dialog box.
6) Click the white “Close” button to close the “Run Configurations” dialog box.

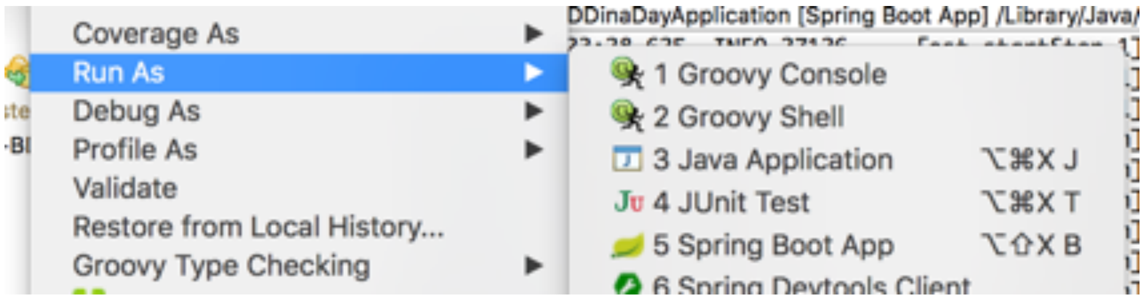
Run the unit and integration tests

1) From the Package Explorer window, click on the triangle next to the project name to display the project files and folders then click on “src” to display the files and folders under it.

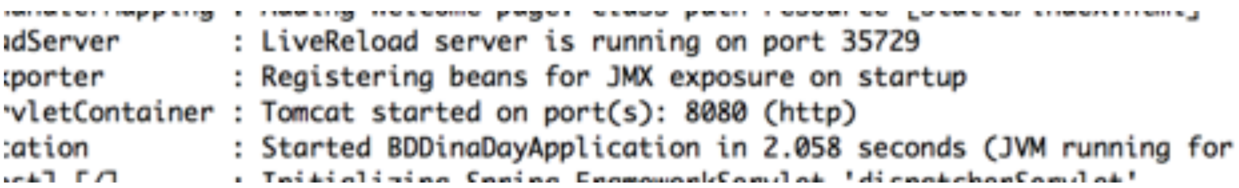
2) Right click on “test” and select “BDD-in-a-Day Unit Test” if it’s displayed. If you get the single choice “Run Configurations…” then select it and when the “Run Configurations” dialog box is displayed, select “BDD-in-a-Day Unit Test” and click the blue “Run” button at the bottom right.
3) Eclipse should show you the “Console” view and things should be scrolling up and out of sight. When it stops, look for a tab labeled “JUnit” — in the tab, there should be a green check box with a white check in it to the left of the word JUnit. That’s a quick way to know that all the tests you last ran passed. If any of them failed, you would see a red box with a white ‘x’ in it instead.
4) Run the Integration and Functional tests the same way. The Integration tests should pass, but the Functional test should fail because it depends on a running instance of the application and we haven’t started one yet.

Run the application

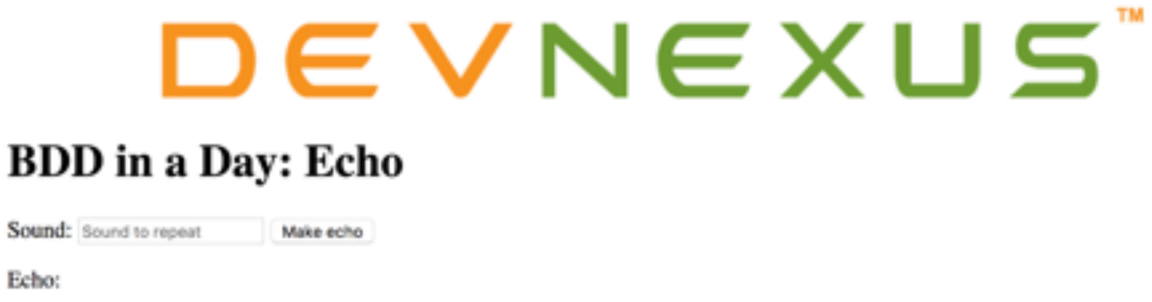
1) Right click on the Project name, hover over “Run as” then click “Spring Boot App”



2) There should be a lot of activity in the “Console” tab that ends with something like this:


```
adServer         : LiveReload server is running on port 35729
kporter          : Registering beans for JMX exposure on startup
vletContainer    : Tomcat started on port(s): 8080 (http)
cation           : Started BDDinaDayApplication in 2.058 seconds (JVM running for
```

3) The important part here is the line that says “Started BDDinaDayApplication in 2.058 seconds” since it’s telling us that the application actually started. The line above it tells us the port number to use when communicating with the server.
4) Now run the Functional Test again. This time it should pass because the program is running.
5) Open your browser and enter “localhost:8080” in the URL field then press “Enter/Return” and, after what feels like a really long pause, you should see something like this:



6) If you can enter a value in the text field, click the button, and get a message containing the value you entered then Congratulations! You are ready for the workshop.
7) If you don't get a web page, or it doesn't respond then check to make sure you typed in "localhost:8080" in the browser. If you can't find any obvious problems, check the FAQ.txt file in the project directory, then try asking a friend to check it out. After that, if you're still having problems, please contact me at "burk.Hufnagel@Daugherty.com" and I'll do what I can to help.