HW4 CS577a - Environment Setup for HW4

Instructions:

- Follow the steps below **carefully!** There are 7 main steps to get you started on automated testing with cucumber.

Step 1: Download Virtual Machine

1. Install VirtualBox on your machine (https://www.virtualbox.org/wiki/Downloads). I am using the latest version 6.0.14
2. Download the virtual machine from this link: https://goo.gl/3po6wL
3. Once you finish downloading, unzip the file
4. Open VirtualBox -> New -> in New, input a name (anything you want) -> in Type, select **Linux** -> in Version, select **Ubuntu (64-bit)** -> click Continue
5. Select memory size of 2048 mb -> click Continue
6. Select "*Use an existing virtual hard disk file*", then browse and select the virtual machine file (.vdi) from step 3 -> click Create
7. On the toolbar -> click Start (This should start up the virtual machine)

Login credentials for sudo commands
Username: **student**
Password: **std**

What I have installed for you in this virtual machine:

- Java
- Ant
- Apache Tomcat 7 [with tomcat manager credentials: "**student**" as username and "**std**" as password]
- Mysql server - [with mysql login credentials: "**root**" as username and "**root**" as password]
- **Note:** Feel free to install any tool that you want to use for your development environment: e.g., Eclipse, Sublime Text, etc.
-

The steps below are to be done in the virtual machine (from STEP 7 above)

Step 2: Update and Upgrade our virtual machine

1. Follow the steps from this website: https://www.tecmint.com/install-virtualbox-guest-additions-in-ubuntu/

Step 3: Setting up our test project

1. Download Bookstore.zip from the link: https://goo.gl/odRHEQ
2. Unzip the file and copy the bookstore folder to any directory of your choice
3. Open a Terminal, "cd" to the project folder (from a step before)
4. Configure the app's database by executing the following command (in the project root dir)

```
student@student-VirtualBox:~/Desktop/Tutorial/bookstore$ mysql -uroot -proot < database/books
tore.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
```

Hint: you can restore the project's database with this command (in case you need it when you do the homework)

5. Edit **Common.jsp** in the project root dir like so (**NOT** from Terminal):

```
//Database connection string

  static final String DBDriver  ="com.mysql.jdbc.Driver";
  static final String strConn   ="jdbc:mysql://localhost/bookstore";
  static final String DBusername="root";
  static final String DBpassword="root";
```

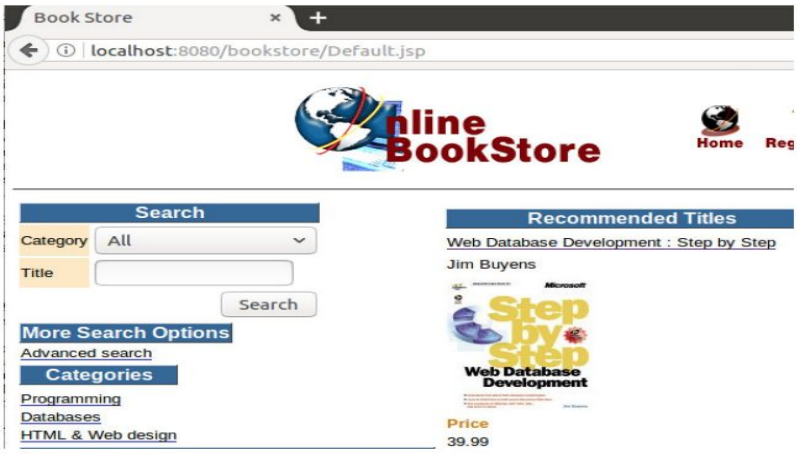
6. Now we need to copy mysql Driver file to tomcat library. In the project root dir (Terminal), execute the following command and then restart tomcat:

```
student@student-VirtualBox:~/Desktop/Tutorial/bookstore$ sudo cp lib/mysql.jar /usr/share/tom
cat7/lib/
[sudo] password for student:
student@student-VirtualBox:~/Desktop/Tutorial/bookstore$ sudo service tomcat7 restart
```

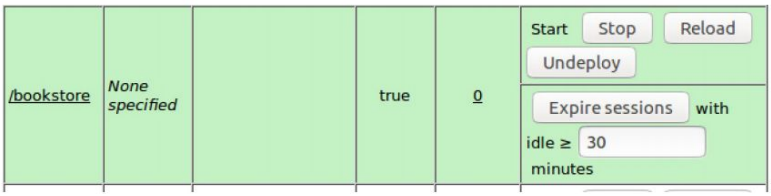
7. Then navigate to **localhost:8080/manager/html**. Again, you may be required to enter tomcat manager login credentials: "**student**" as username and "**std**" as password. Enter the information similar to what is shown below (*Note: directory URL depends on the path to your bookstore folder*) -> Click Deploy

8. You should now see bookstore listed as one of the application in the table above (in the same page). Now, navigate to **localhost:8080/bookstore/Default.jsp**. You should now

see our bookstore website



9. Navigate to Tomcat Manager page **http://localhost:8080/manager/html** in a row that contains "**bookstore**" select "**undeploy**"



Step 4: Install Gecko Drive from Mozilla (so that we can open Firefox from Terminal)

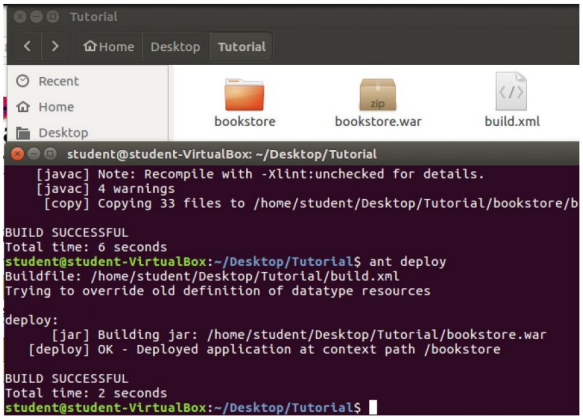
1. From Firefox, go to: https://github.com/mozilla/geckodriver/releases
2. Download the one with **linux64**
3. Open Terminal, navigate to the download folder
4. Execute: **tar -xvzf geckodriver-v0.2….**
5. Execute: **rm geckodriver-v0.2….**
6. Execute: **chmod +x geckodriver**
7. Execute: **sudo cp geckodriver /usr/local/bin/**

Step 5: Install Cucumber

1. Run these commands in Terminal
    - **sudo apt-get install ruby**
    - **sudo apt-get install ruby-dev**
    - **sudo apt-get install zlib1g-dev**
    - require you to follow these steps **https://www.brightbox.com/blog/2017/01/13/ruby-2-4-ubuntu-packages/**
    - **sudo gem install capybara**
    - **sudo gem install cucumber**
    - **sudo gem install selenium-webdriver**

Step 6: Configure Automate Build Files (so that we execute everything from Terminal)

1. Download WEB-INF: https://goo.gl/MMLbNE
2. Unzip and put the WEB-INF folder inside the bookstore directory.
3. Download Build.xml file: http://bit.ly/33fQKgu
4. Put the file in the directory containing the application folder (**NOT** inside the bookstore folder, see the picture below - in **Step 8**).
5. Execute **from the directory containing build.xml** file: **ant compile**
6. You should get a "*BUILD SUCCESSFUL*" message (from step 5)
7. Then type in "**ant deploy**" ← to deploy the .war file to the Tomcat server After this step, you should see "bookstore.war" in the directory



8. Navigate to **localhost:8080/bookstore/Default.jsp** You should be able to surf the website again.

9. To undeploy, simply type in "**ant undeploy**"

Step 7: Cucumber Tutorial

1. In the directory that contains our bookstore folder, execute in Terminal "**cucumber --init"** it should show that 4 items are created. The most important one is the "features" folder
2. Go into "features/support" you should see a file called "env.rb"
3. In env.rb put in

```
require 'capybara/cucumber'
require 'selenium/webdriver'
require 'test/unit/assertions'

Capybara.configure do |config|
    config.run_server = true
    config.default_driver = :selenium
    config.app_host = "localhost:8080"
end

World(Test::Unit::Assertions)
```

This is to setup the environment to run cucumber tests

4. Go to "step_definitions" folder, create a file called "steps.rb". Then add our step definitions like so

```
Given(/^I am on the default page$/) do
  visit("/bookstore/Default.jsp")
end

When(/^I navigate to "([^"]*)"$/) do |link|
  visit(link)
end

Then(/^I should see "([^"]*)"$/) do |text|
  assert page.has_content?(text)
end
```

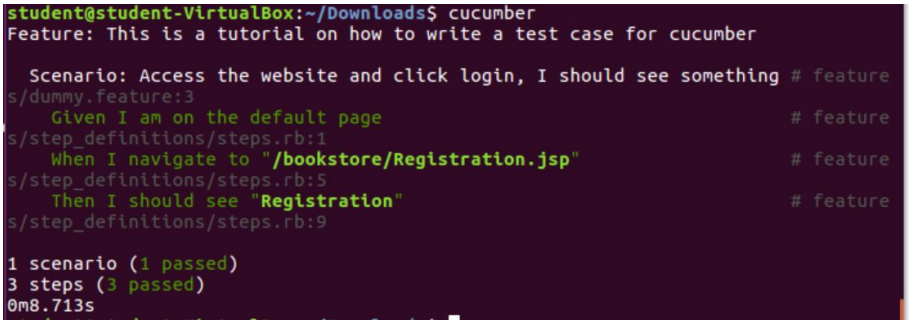
5. In Step 4, we defined 3 basic steps for our test.
6. Let's create our test case. In "features" folder (folder that contains both step_definitions folder and support folder), create a file name "***dummy.feature***"
7. Add content to our dummy.feature file like so

dummy.feature

```
Feature: This is a tutorial on how to write a test case for cucumber

        Scenario: Access the website and click login, I should see the word registration
                Given I am on the default page
                When I navigate to "/bookstore/Registration.jsp"
                Then I should see "Registration"
```

8. Then at the root folder (folder that contains our bookstore folder and features folder), execute "**cucumber**" - You should now see your test runs automatically!



Troubleshooting

- If you cannot install anything (when run “sudo apt-get install <packagename>”) because of **dpkg lock**
  - Try running:
    - ```
      sudo systemctl disable apt-daily.service # disable run when system boot
      sudo systemctl disable apt-daily.timer   # disable timer run
      ```
    - Then restart the virtual machine.
      ```