Lab 1 - Setup

Summary

The purpose of this assignment is to ensure everyone has a functional development environment installed and tested for ITMD4515.

Requirements

Download and Install

Download and install the following:

1. Download the latest **OpenJDK 11 (LTS)** with the Hotspot JVM from https://adoptopenjdk.net and install with the defaults.

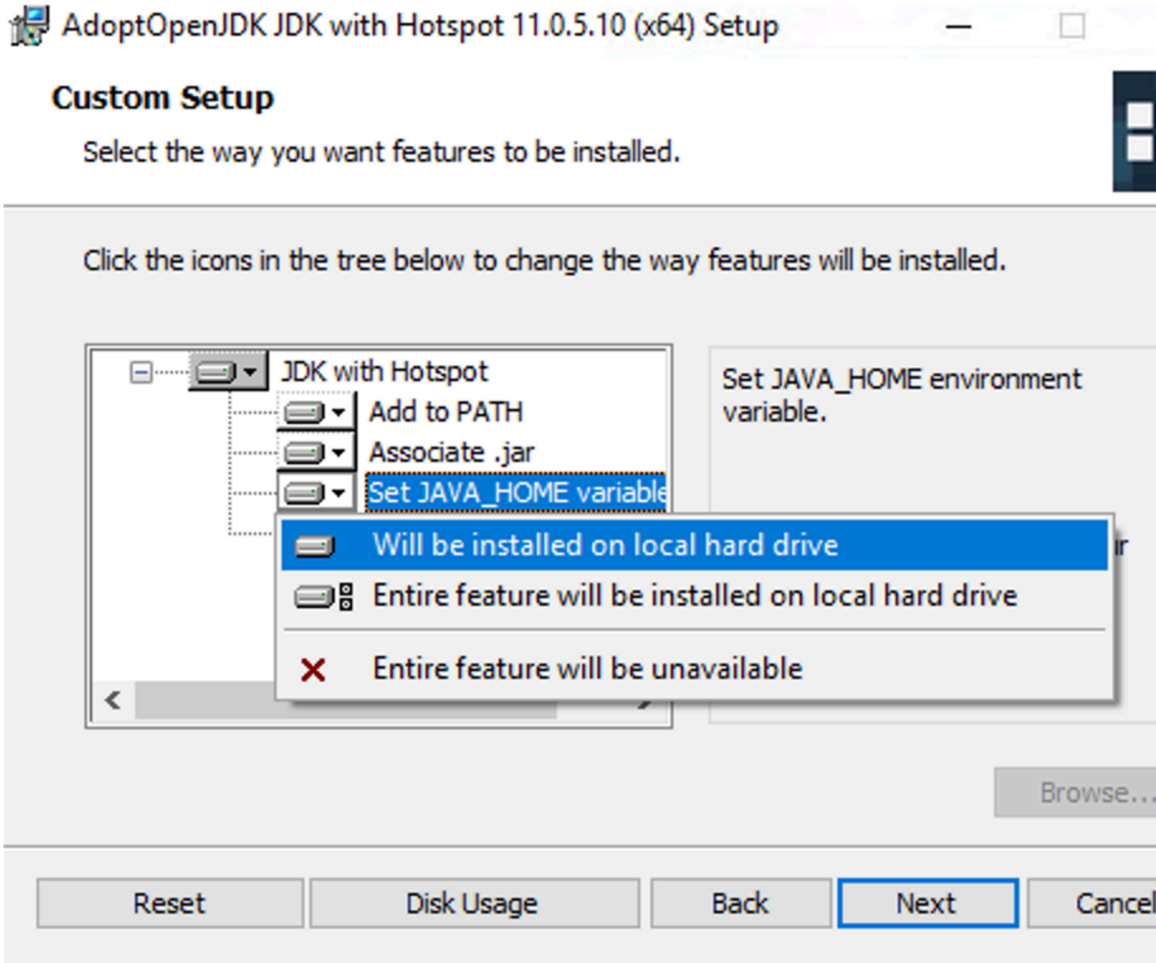
   If you already have the latest version 11 of another JDK, you can use that. **You must have a JDK installed, not just a JRE.**

   1. This will be an MSI installer on windows, a PKG installed for Mac, or a tar.gz archive on linux. On Mac, if you use the homebrew package manager you can also install with the brew command:

      linux/macOS

      ```
      # linux
      cd /your/preferred/location/for/the/jdk
      tar -zxvf OpenJDK11U-jdk_x64_linux_hotspot_11.0.9.1_1.tar.gz

      # macOS only - if you are using the homebrew package manager
      brew cask install adoptopenjdk11
      ```

   2. Run the installer with default options, but on Windows make sure to select the checkbox to **Set JAVA_HOME environment variable** which is not selected by default. This will save you a lot of time, especially if you are unfamiliar with environment variables or operating systems in general:

    1. 

AdoptOpenJDK JDK with Hotspot 11.0.5.10 (x64) Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Set JAVA_HOME environment variable.

JDK with Hotspot
Add to PATH
Associate .jar
Set JAVA_HOME variable
Will be installed on local hard drive
Entire feature will be installed on local hard drive
Entire feature will be unavailable
Browse...
Reset
Disk Usage
Back
Next
Cancel

3. Additional installation instructions are available for all platforms here: https://adoptopenjdk.net/installation.html
4. For Mac users, you may need to update your own environment variables (ref: http://www.mkyong.com/java/how-to-set-java_home-environment-variable-on-mac-os-x/):
    1. Open the Terminal application
    2. Add the following to the profile you use, typically .profile or .bash_profile:

       MacOS

       ```
       # if you installed AdoptOpenJDK as outlined above, via PKG
       installer or homebrew, you can just do this:
       export JAVA_HOME=$(/usr/libexec/java_home -v 11)
       export PATH=${JAVA_HOME}/bin:${PATH}
       ```

5. For Linux users, add the following to the profile you use, typically .profile or .bash_profile:

   Linux

```
export JAVA_HOME=/wherever/you/extracted/the/JDK/from/above/jdk-
11.0.9.1+1
export PATH=${JAVA_HOME}/bin:${PATH}
```

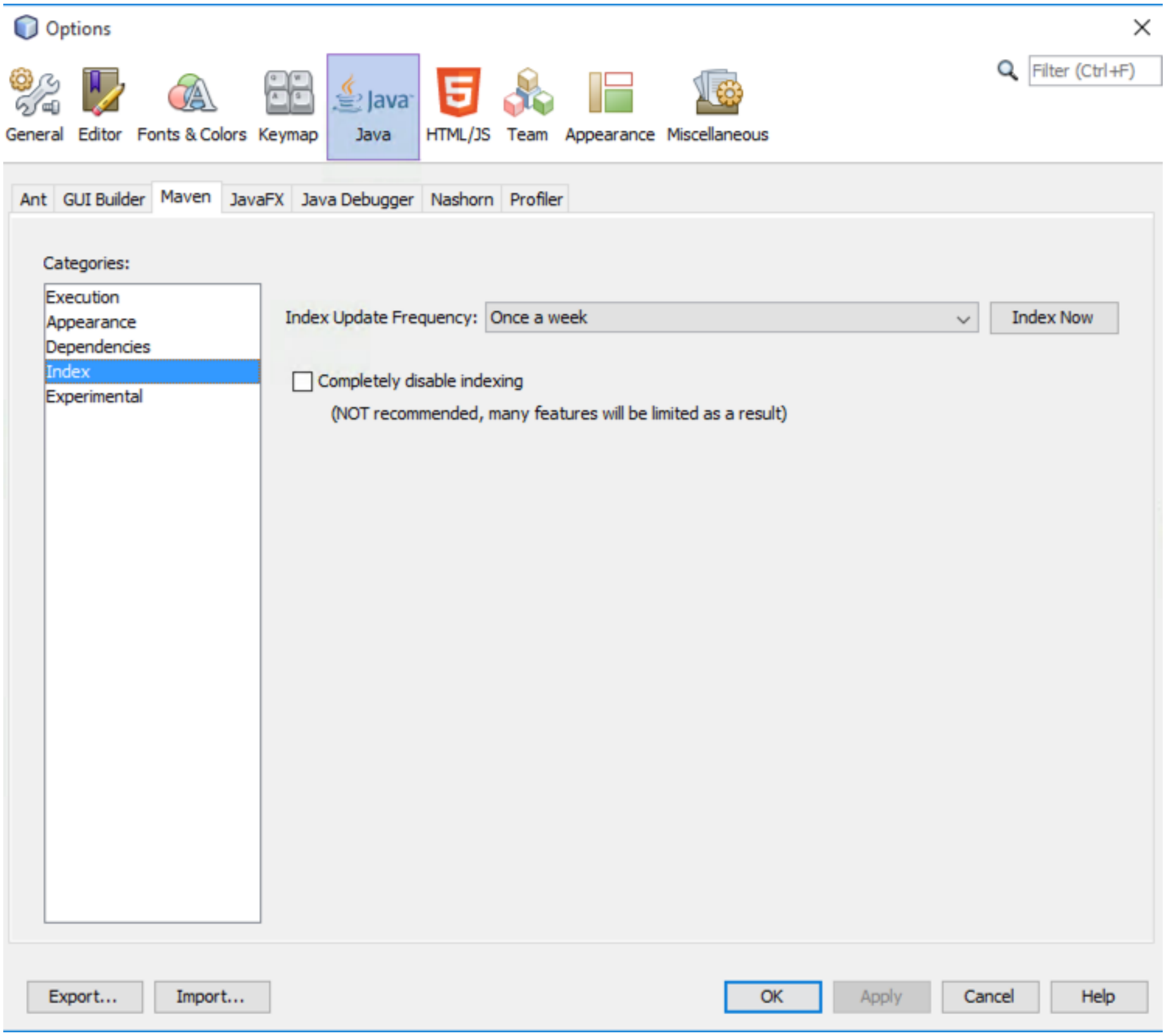
If you choose to use Linux in this class, I applaud your choice but also expect you to know the command line.

2. Test! Open up a cmd prompt or PowerShell on Windows (PowerShell illustrated below) and do the following. Make sure your output is sane. The following is an abbreviated example, not complete. Make sure you **see** the environment variables you just set, and that you can execute commands within the JDK. For Mac/Linux, open your native Terminal application:

```
PS C:\Users\sas691> gci env:
...
JAVA_HOME                      C:\Program Files\AdoptOpenJDK\jdk-
11.0.9.1-hotspot\
...
Path                           C:\Program Files\AdoptOpenJDK\jdk-
11.0.9.1-hotspot\bin;C:\ProgramData\DockerDesktop\version-
bin;C:\Program
Files\Docker\Docker\Resources\bin;C:\WINDOWS\system32;C:\WIND...
...

PS C:\Users\sas691> java -version
openjdk version "11.0.5" 2019-10-15
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.9.1+1)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.9.1+1, mixed mode)

PS C:\Users\sas691> javac -version
javac 11.0.9.1

# on linux or Mac, you can view environment variables differently in a
terminal:
env | egrep 'JAVA|PATH'
```

3. Download **Apache NetBeans 12** from https://netbeans.apache.org/download/index.html. Install with the defaults.

   The first time you run NetBeans, or perhaps the first time you run a maven project within NetBeans, you might find a very slow and long-running task named **Unpacking index for Central Repository**. Let it run and finish the first time, which might take as long as 10 or 20 minutes. After it finishes, you can control how often this task runs in your NetBeans preferences:



Index Update Frequency: Once a week

□ Completely disable indexing

(NOT recommended, many features will be limited as a result)

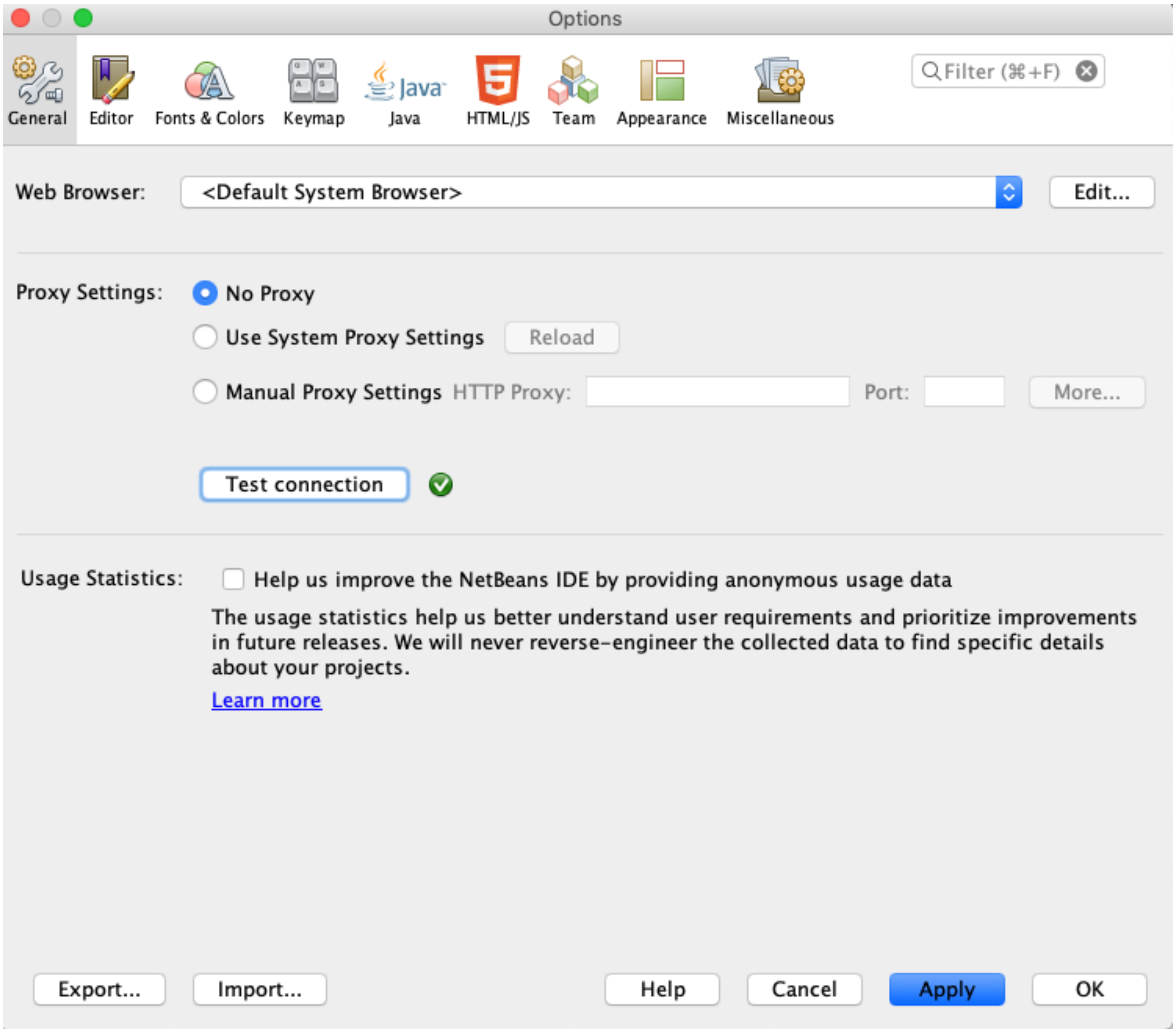
Additionally, you will find that at various initial stages (first time you open a maven project, first time you start a Java EE project, first time you add a server, etc) NetBeans will want to install and activate various plugins. Allow it to do so. If it appears that NetBeans is locked up, first be patient - it may just be activating a plugin. If it doesn't respond after a reasonable amount of time, close and re-open NetBeans.

Optionally, see How-to articles for additional NetBeans settings that I personally like to use.

4. Download Payara Server Community, which is fully compatible with Jakarta EE 8: https://www.payara.fish/downloads/payara-platform-community-edition/. At the time of this writing, the version is 5.2020.7.
5. Follow my demo, or the Java EE 8 Hands on Lab - Initial Setup instructions, to add Payara to NetBeans as a server. If you follow the HOL instructions, be aware they are

written for Glassfish. The general steps are still fine for Payara, but you must mentally substitute Payara for Glassfish.

Important - on macOS, you may need to adjust the NetBeans proxy settings as follows (select No Proxy):



Web Browser: <Default System Browser>

Proxy Settings:
● No Proxy
○ Use System Proxy Settings
○ Manual Proxy Settings HTTP Proxy: Port:

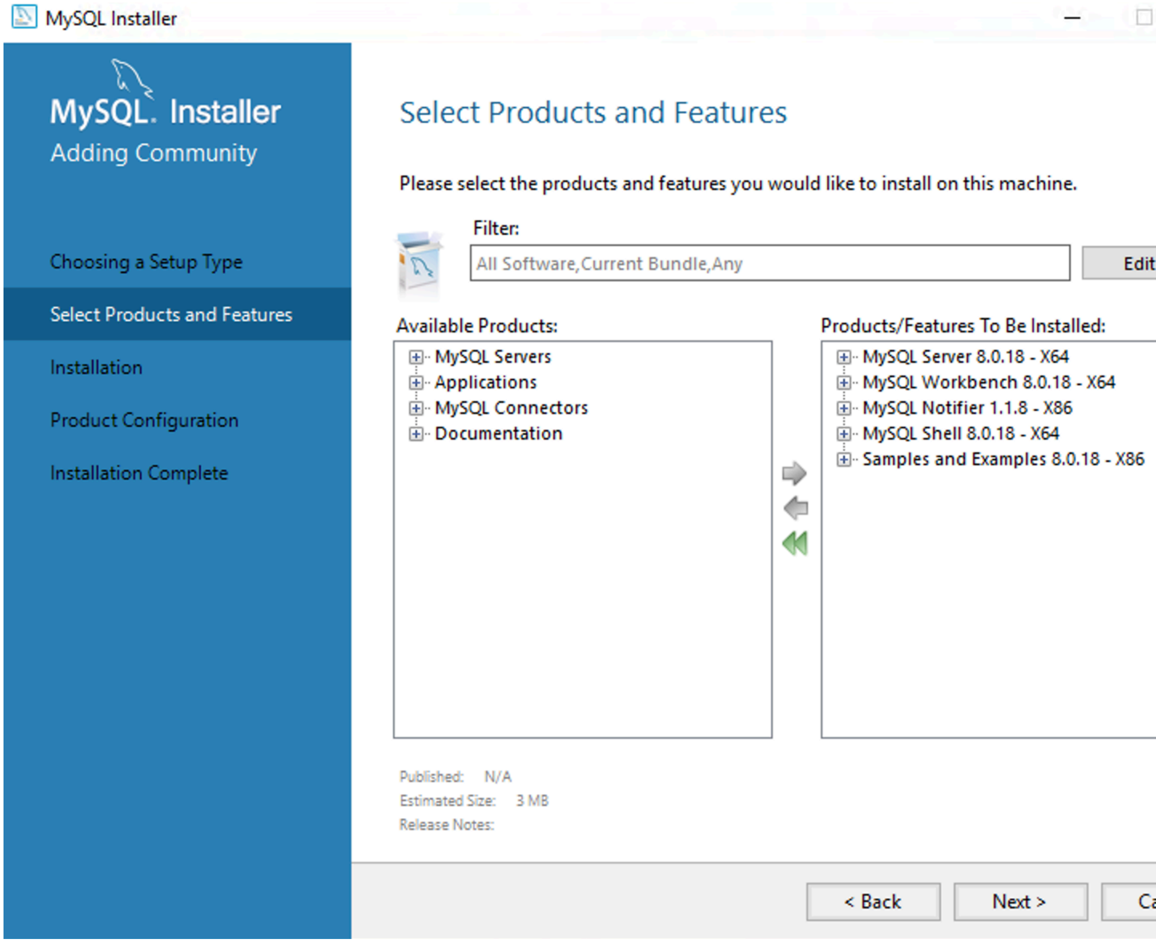
Usage Statistics: □ Help us improve the NetBeans IDE by providing anonymous usage data

The usage statistics help us better understand user requirements and prioritize improvements in future releases. We will never reverse-engineer the collected data to find specific details about your projects.

If Payara starts normally for you, don't worry about this. If Payara is slow to start, or NetBeans complains about starting/stopping Payara, then set the "No Proxy" and try again. (ref: https://netbeans.org/bugzilla/show_bug.cgi?id=268076). I wish I could be more specific about this bug, but unfortunately it is not consistent. On Mojave, I have never experienced it, but 2 students in Spring 2019 did experience it on Mojave.

6. Download the latest MySQL Community Server 8.0 from https://dev.mysql.com/downloads/mysql/
    1. At the time of this writing, that is 8.0.21

2. I recommend selecting a Custom Setup Type, so you don't end up with Excel and Visual Studio add-ons.
3. Then choose the following:
    1. MySQL Server 8.0 (latest version)
    2. MySQL Workbench (latest version)
    3. MySQL Shell (latest version)
    4. MySQL Notifier (latest version)
    5. Samples and Examples 8.0 (latest version to match MySQL Server above)
    6. The following screenshot shows everything you should install (just make sure you have the latest versions):



MySQL Installer

Select Products and Features

Please select the products and features you would like to install on this machine.

Filter: All Software,Current Bundle,Any

Available Products:
- MySQL Servers
- Applications
- MySQL Connectors
- Documentation

Products/Features To Be Installed:
- MySQL Server 8.0.18 - X64
- MySQL Workbench 8.0.18 - X64
- MySQL Notifier 1.1.8 - X86
- MySQL Shell 8.0.18 - X64
- Samples and Examples 8.0.18 - X86

Published: N/A
Estimated Size: 3 MB
Release Notes:

    7.

You can follow whatever Setup Type you want, but please be aware that if you chose Developer Default the installer will try and install add-ons for Visual Studio, Excel, Python, and other tools you may not even have on your laptop.

4. If you are missing any dependencies, the installer will prompt you to install them. For example, on my Windows VM I needed Visual C++ Runtime. The

installer will download the missing dependencies for you, and prompt you to execute them.

5. After successful installation, MySQL will prompt you to configure. Accept the default Development Machine configuration, and set a MySQL root password of your choosing.
6. On the "Windows Service" screen, it is up to you whether to start the MySQL service at boot. I chose **not** to do so on my Windows environments, because I prefer to manually control the database. I recommend de-selecting the checkbox next to "Start the MySQL Server at System Startup."
7. Finish the installation
8. MySQL Notifier
    1. If you don't see this in the **initial** installation options, skip it and install the other services. Then, re-run MySQL setup when everything is successfully installed and install notifier.
    2. If you still don't see it, you can download the Notifier directly from here: https://downloads.mysql.com/archives/notifier/
    3. In the past I have had trouble with the MySQL Notifier the first time I use it to stop/start a MySQL instance. Restart your computer after installing MySQL notifier. If you are still having trouble with MySQL notifier on Windows after restarting, ask a Question so I can help.
9. Mac users: There are several different ways to install MySQL on Mac. If you already have a preferred way, use it. If you need assistance or a recommendation, please reach out to me. Personally, I use the Homebrew package manager for MySQL and many other things on Mac, and I manually start and stop MySQL when I need it for development

    MacOS

    ```
    brew install mysql
    mysql.server start
    mysql_secure_installation
    ```

10. Linux users: This will vary by distribution. Here are some articles for various distributions:
    1. Install MySQL Server on Ubuntu (from Rackspace Support)
        1. **Note** - unless you are setting up a server for remote access, you can skip the "Allow remote access" section. The important commands are:

            Linux

            ```
            sudo apt-get update
            sudo apt-get install mysql-server
            systemctl start mysql
            systemctl enable mysql

            # if mysql_secure_installation didn't run
            automatically after installation, run it manually
            ```

7. Open MySQL Workbench
    1. Make sure you can connect as the root user to one of the sample databases
8. In your documentation, answer the following:
    1. Brief introductory paragraph about your development environment (Mac, Windows, Linux, etc).
    2. Screenshot of **connected** MySQL Workbench. I describe in the Week 1 lecture/demo what this means.
9. Open NetBeans

    As mentioned above, if this is the first time you have opened a maven or Java EE project in NetBeans, you will be prompted to install plugins. Do so, and see the notes above if you are having trouble.

    1. Clone the Java EE 8 hands on lab from https://github.com/javaee/j1-hol using NetBeans
        1. Open the demos/jsonbbeanvaldemo project and allow NetBeans to complete the "priming build"
        2. Explore the project
        3. Right click the jsonbbeanvaldemo project and select "Build with Dependencies"
        4. View the source code in NetBeans. What are your observations?
        5. Right click **Dependencies** within the jsonbbeanvaldemo. Download Javadoc and Download Sources.
        6. View the source code in NetBeans again. What are your observations regarding what has changed?
        7. Right click the jsonbbeanvaldemo project and select "Run." Copy the output of this command from the "Output" window. This is NetBeans displaying what you would normally see on standard output at the command line.
        8. What are your observations about the web browser window that initially opens?
        9. Go to http://localhost:8080/jsonbbeanvaldemo/ and submit the form. Are there any errors or challenges submitting the form? What date format seems to be required?
            1. Hint - during Week 1 lecture/demonstration we fixed a bug in this demo project.
            2. Make the bugfix (watch the video) in the index.html file and re-run the project
            3. Go back to http://localhost:8080/jsonbbeanvaldemo/ and submit the form. What fields are required?
        10. **Right click the jsonbbeanvaldemo project and select "Clean"**
10. In your documentation answer the following:
    1. Paste only the output from jsonbbeanvaldemo indicating a **successful** submission.
    2. Include a screenshot of your NetBeans environment showing the project opened.

11. Go to your NetBeans Projects directory. Create a zip file of the jsonbbeanvaldemo folder and submit it to the Blackboard assignment.