## Development Environment Setup

This document describes how to setup the development environment for our project.

## Install Software

### Git Client

You will need git to checkout and checkin code.

Windows

* Download From: <https://git-scm.com/download/win>
* Path Options:
  + I would choose “Use Git from the Windows Command Prompt”.
* After installing check that git is available from the command line.
  + Go to a console window and type git –version

C:\Users\bryan>git --version

git version 2.15.1.windows.2

## Java Development Kit

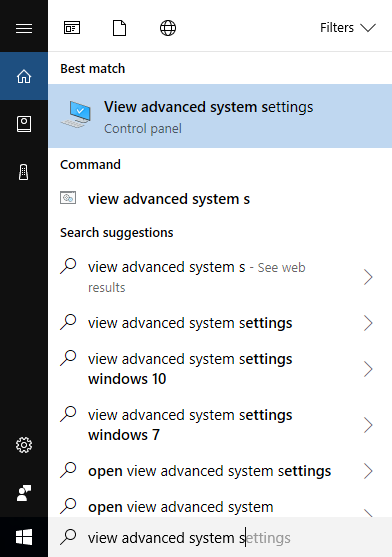
## Download and install the JDK

* Download from: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

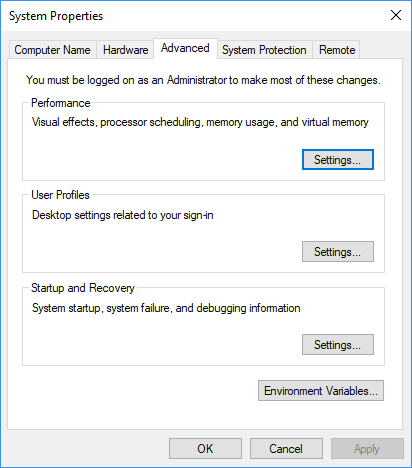
## Configure your PATH (windows)

After installing it is recommended that your setup your path to include java related tools for command line access.

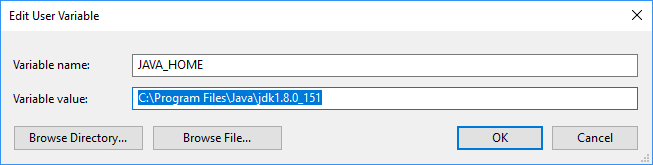
#### Go to Advanced system settings.



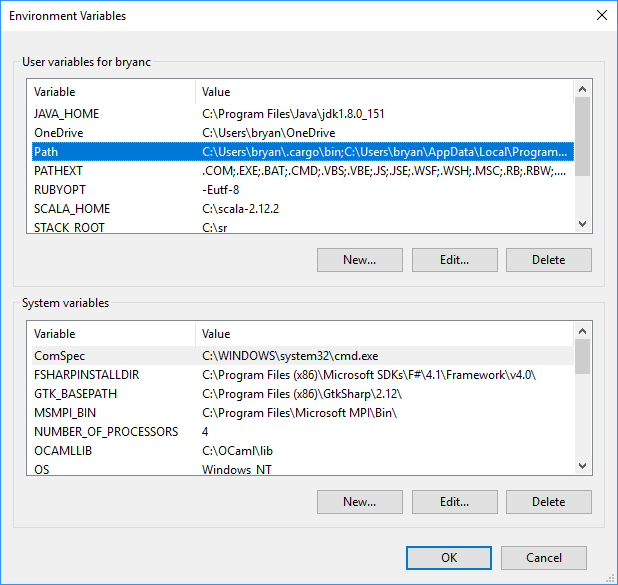
#### Choose “Environment Variables”.

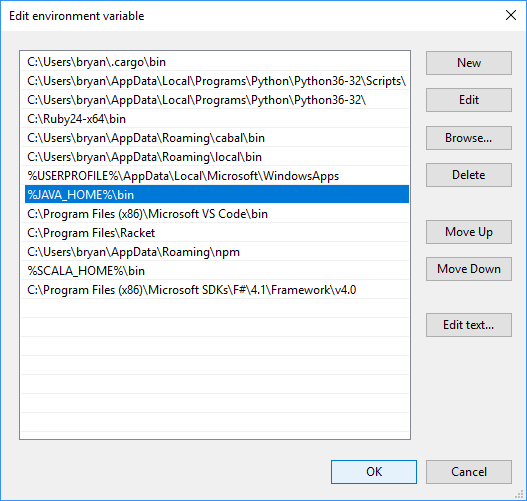


#### Add user variable for JAVA\_HOME. The value should be the directory that the JDK was installed.

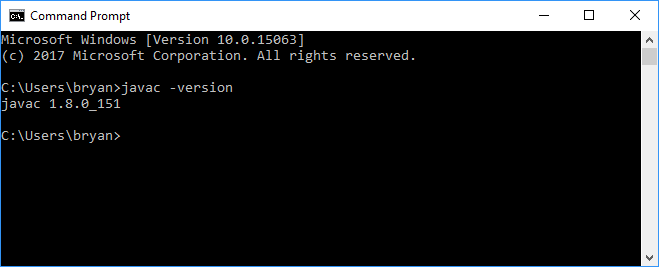


#### Edit the PATH variable and put %JAVA\_HOME%\bin in it.





#### After editing the PATH, open a new command prompt and double check that you can run javac.

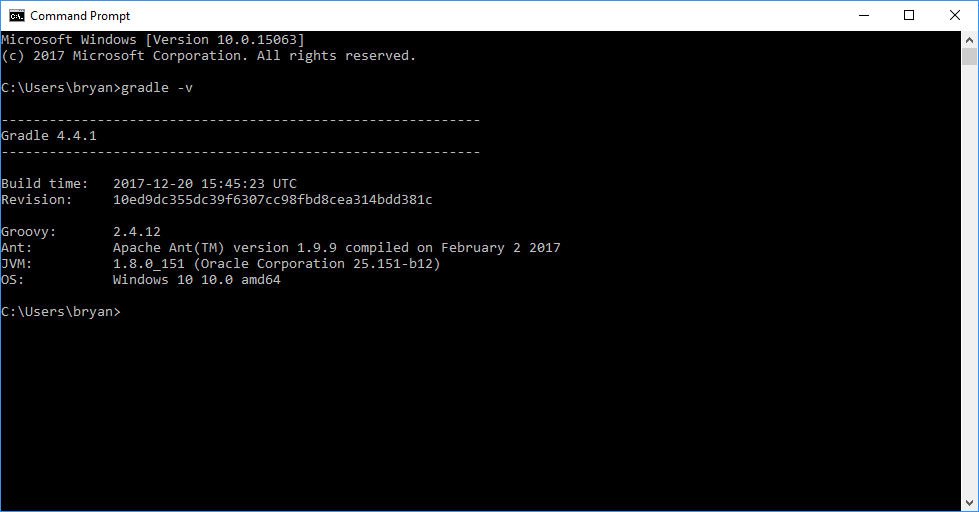


## Gradle

Gradle is a build tool for java applications. Installing this helps setup your development environment and is used for managing java libraries.

* Download: <https://services.gradle.org/distributions/gradle-4.4.1-all.zip>
* Install Instructions: <https://gradle.org/install/#manually>

#### After installing open a new command prompt and check that you can run gradle.

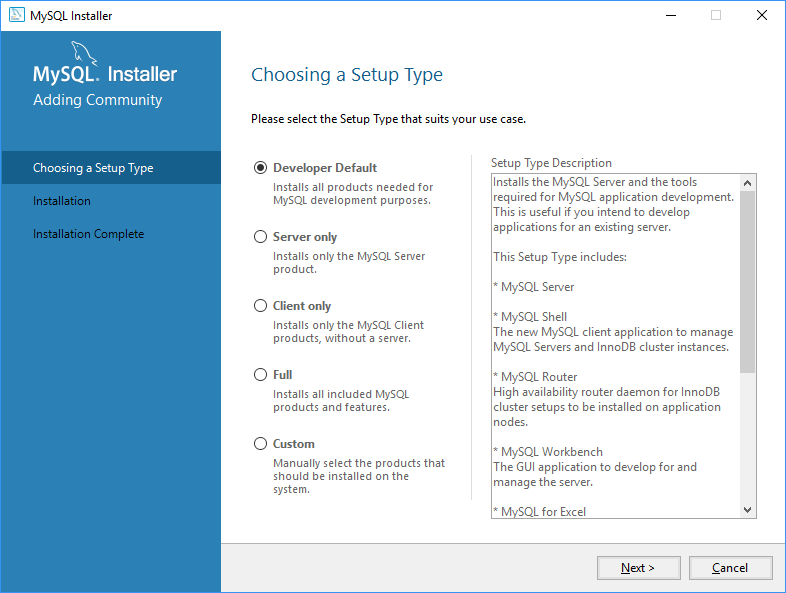


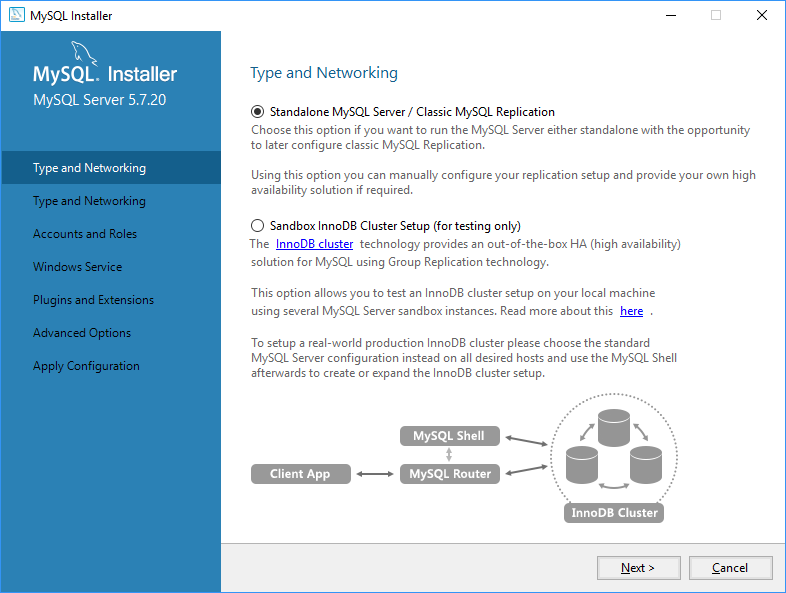
## MySQL Server

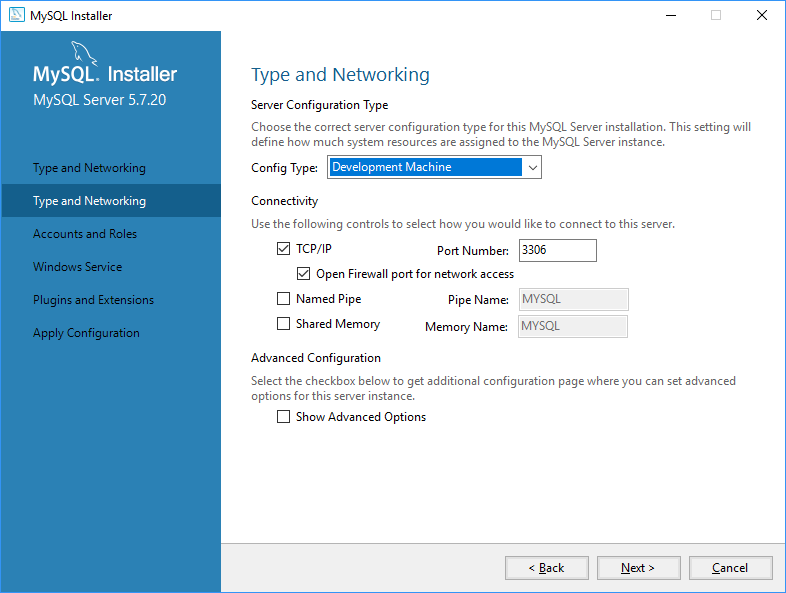
We will be using MySQL for the database.

* <https://dev.mysql.com/downloads/mysql/>
* I used MySQL installer for windows and I picked “Developer Default”. We probably don’t need all of the products, but you probably want “MySQL Server”, “MySQL Shell”, and “MySQL Workbench” at the least.

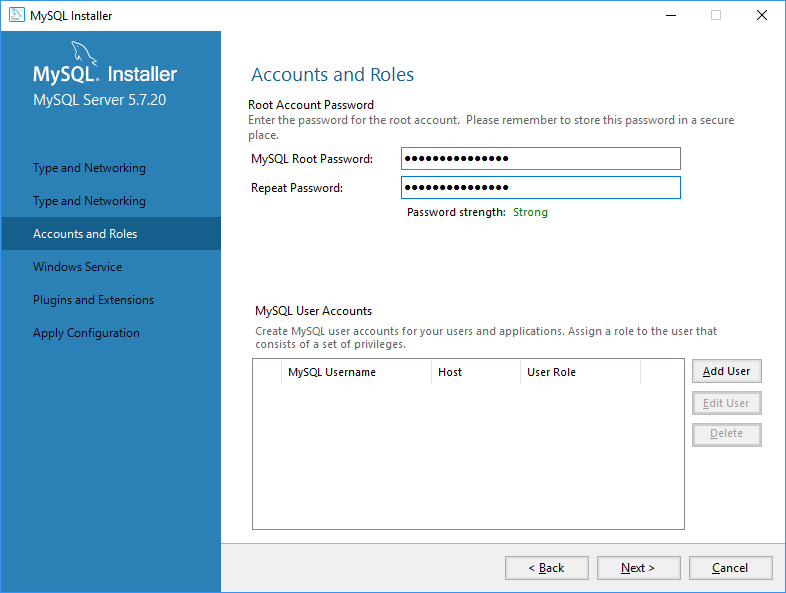
Here are some of the steps I went through to install MySQL.

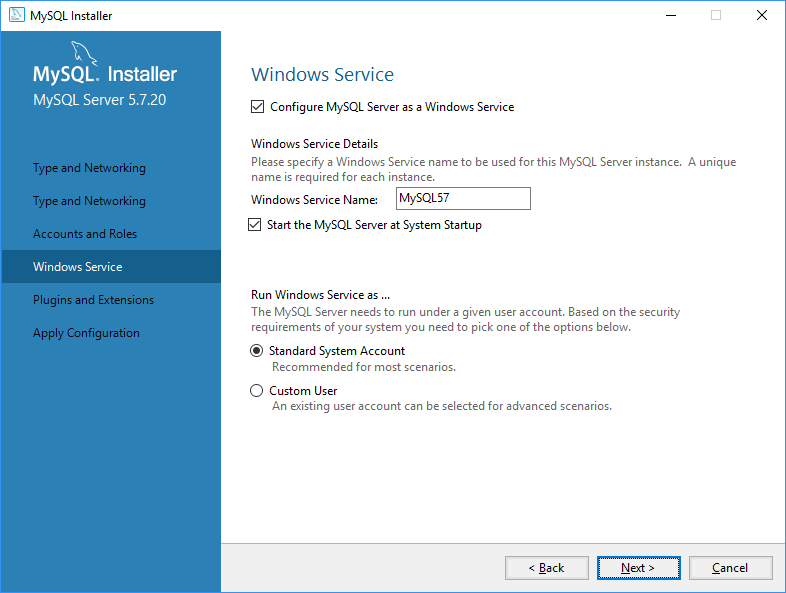


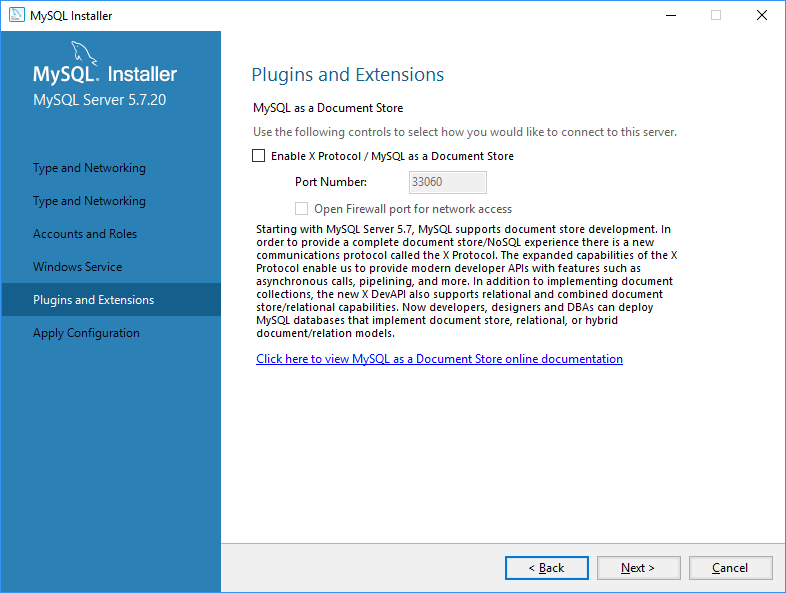


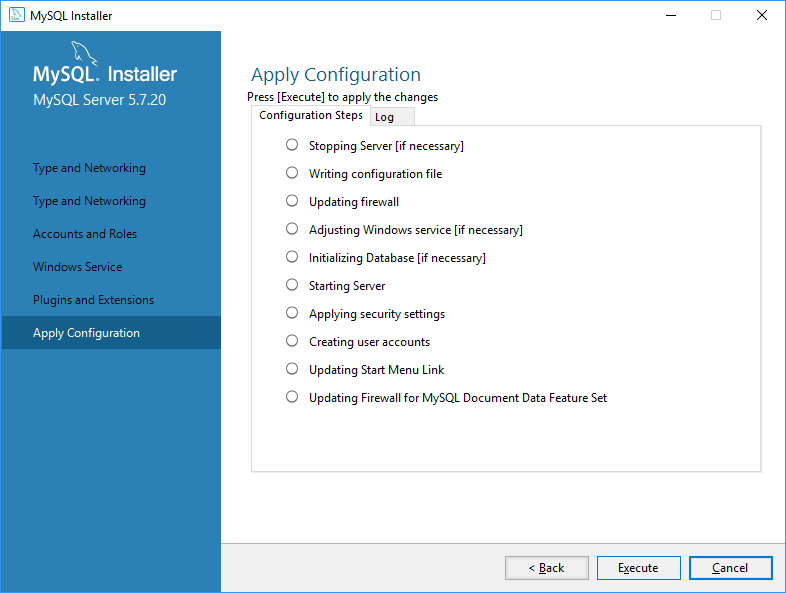


#### Pick a password for root and save it.





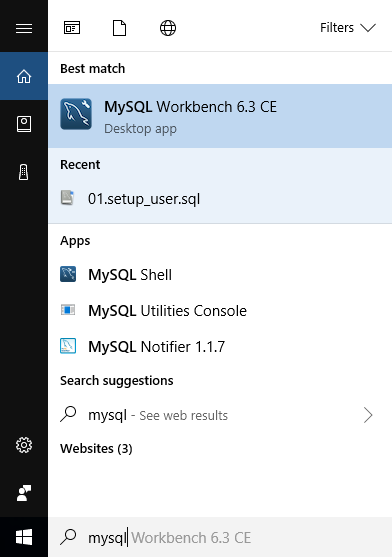




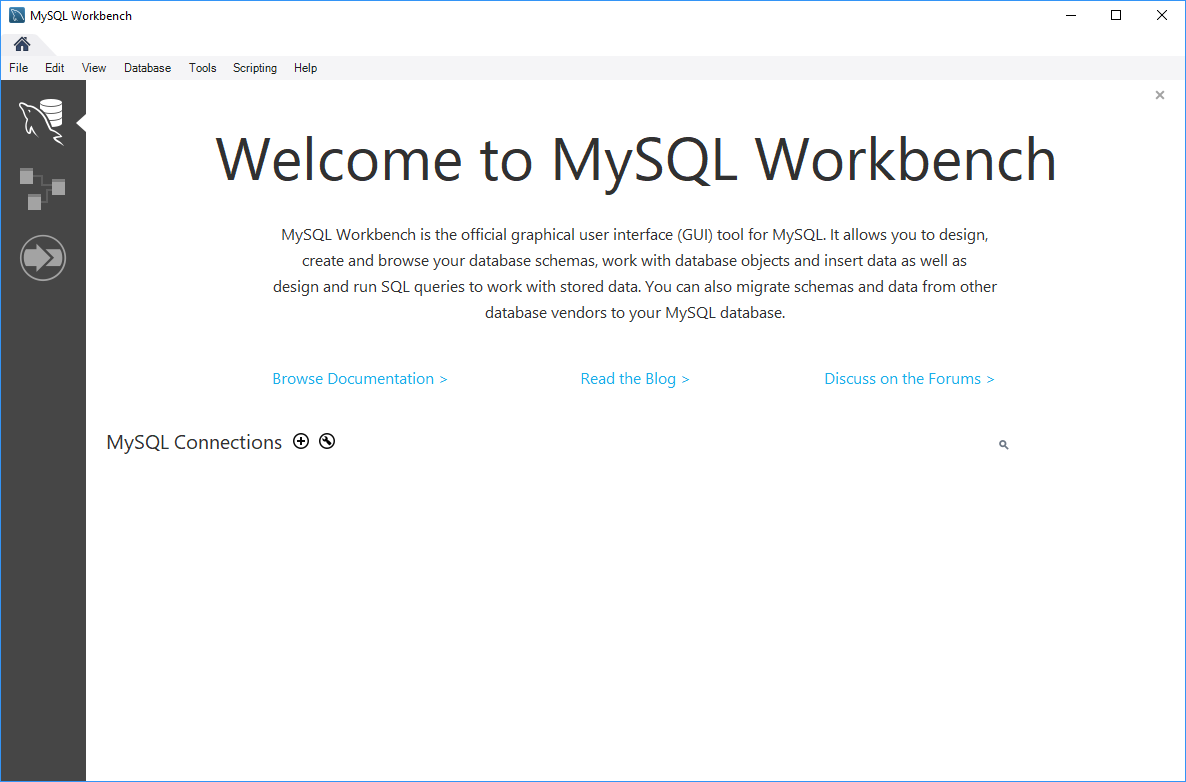
There are a few more steps, but it should be pretty clear.

#### Try connecting to MySQL.

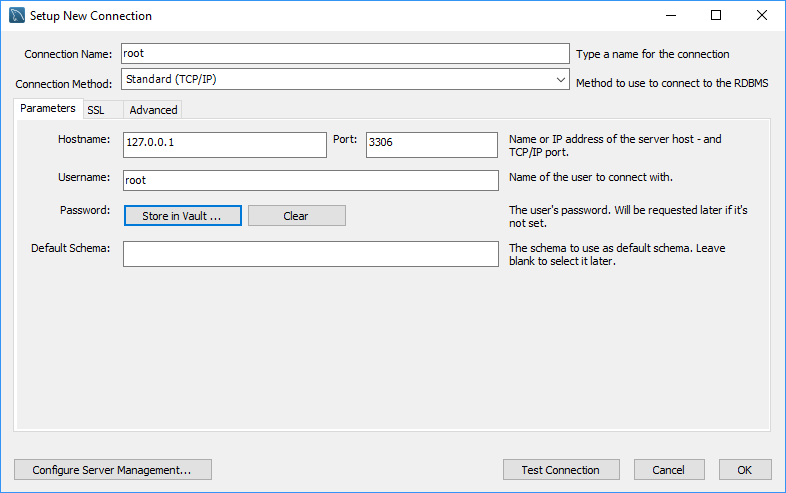
Start typing MySQL at the search bar. You should see MySQL Workbench. You can use this to connect, view the database, and query the DB.



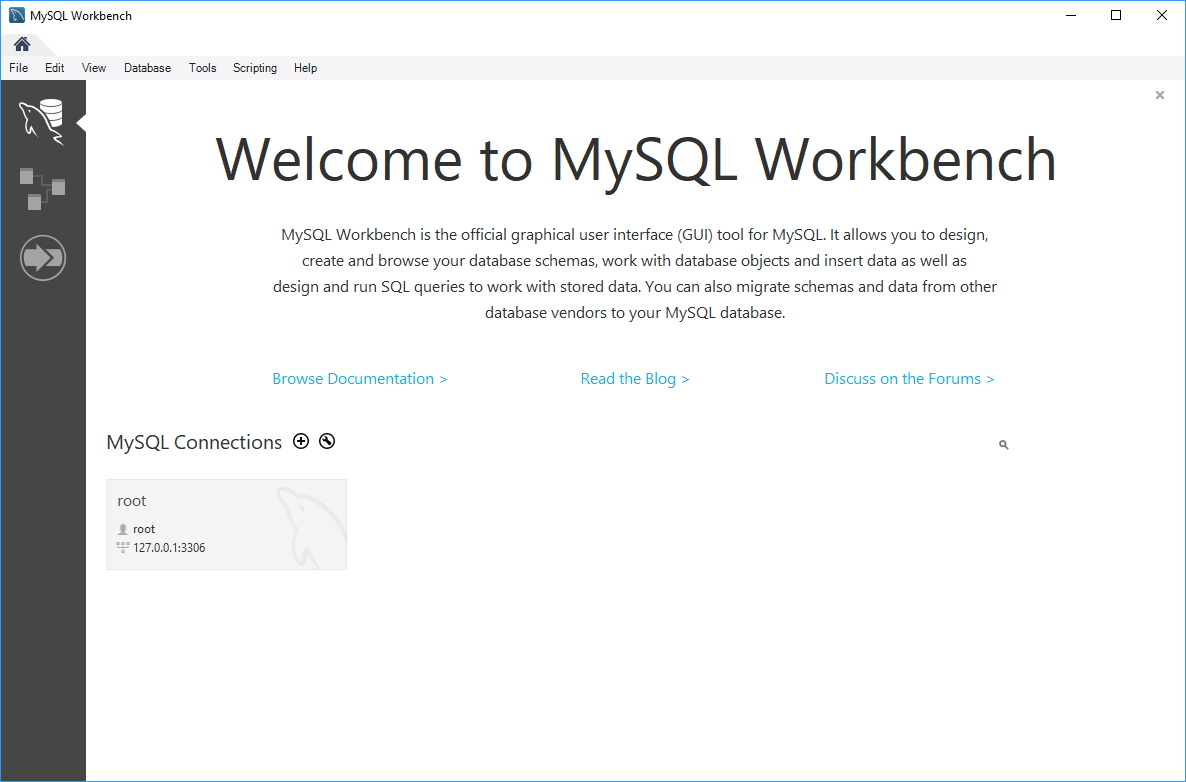
#### Once you open up the app, you will want to configure a connection. Click the + icon next to MySQL Connections.



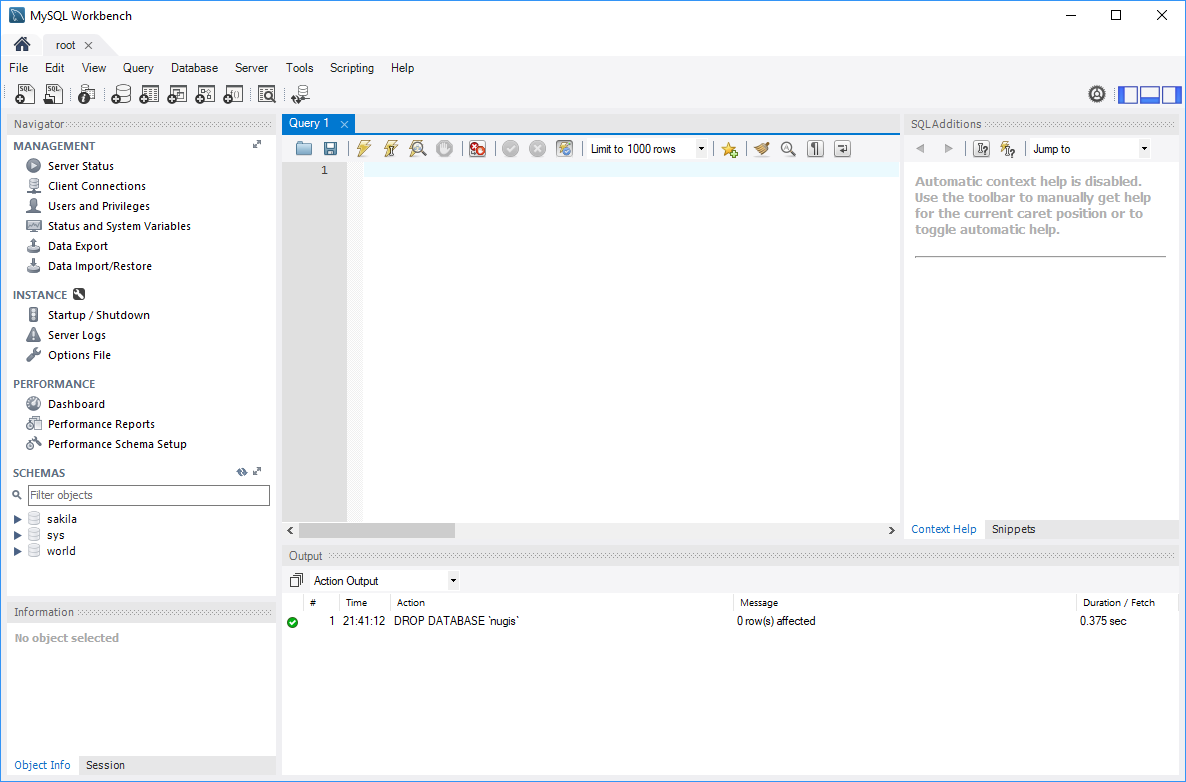
#### Create a connection named root for the root user. This is what you will use for initial setup.



#### You should now have a root tile under connections. Click that.



#### You should have a screen now with a query window.



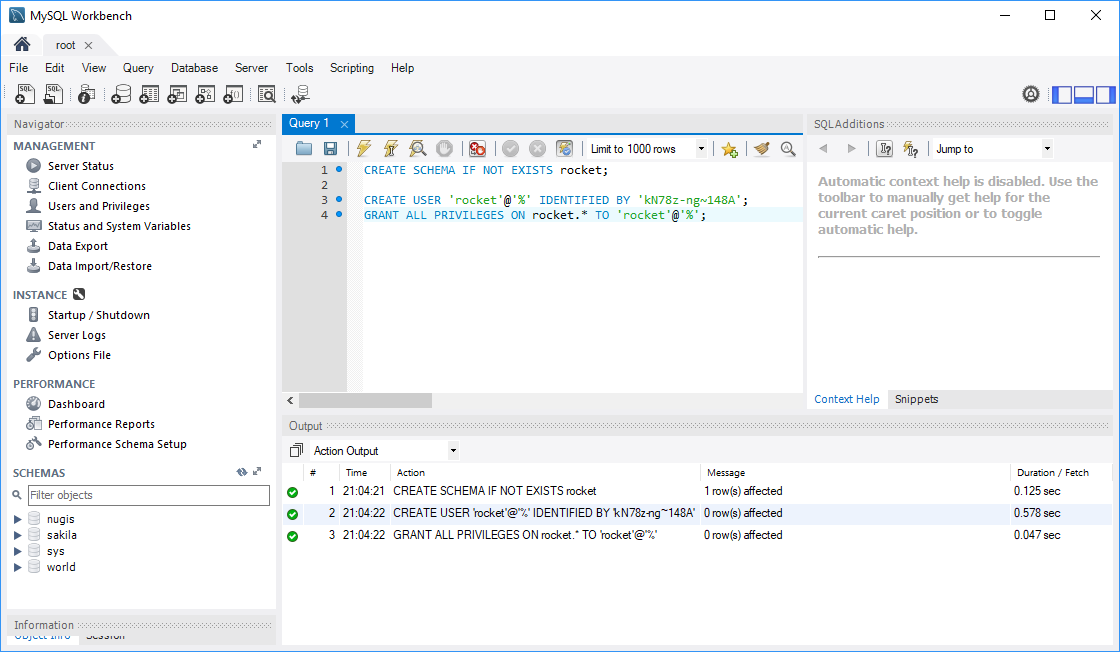
#### The next step is to create a user and a schema for our project. We will not be using the root user for the application.

Run the following SQL:

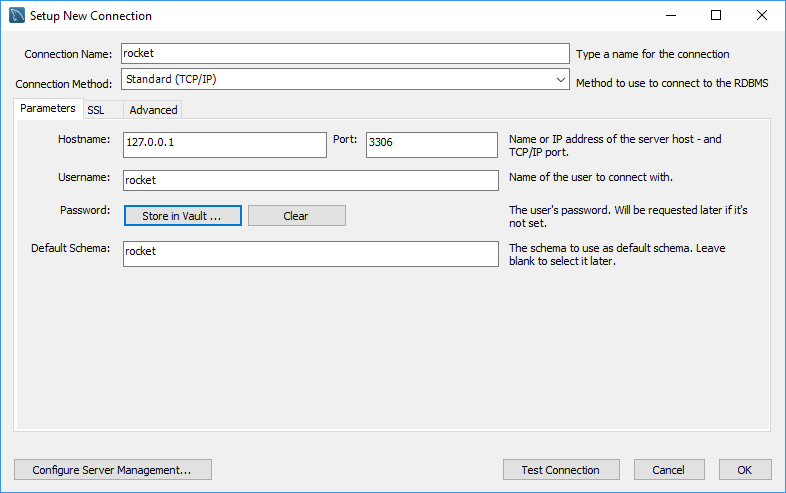
CREATE SCHEMA IF NOT EXISTS rocket;

CREATE USER 'rocket'@'%' IDENTIFIED BY 'kN78z-ng~148A';

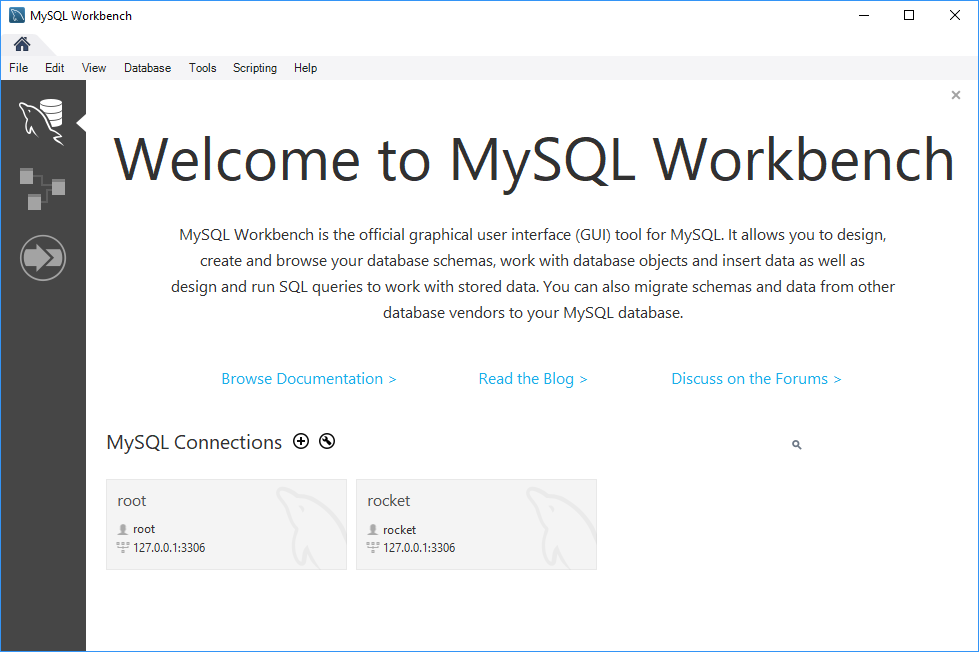
GRANT ALL PRIVILEGES ON rocket.\* TO 'rocket'@'%';



#### Close the root tab and create a new connection for the schema and user we just created.



#### You should now have 2 connection tiles.



#### Install Eclipse

You can certainly use any IDE you like for the project, but the gradle build script has support for generating either a Eclipse or IntelliJ project files. I decided to use Eclipse to setup the project.

* Download Eclipse: <https://www.eclipse.org/downloads/>
* When you run the installer, there are options for which version of Eclipse you want to run. I picked the normal Java developer, but it didn’t have as much support for JavaScript, html, and CSS. I ended up running the eclipse plugin manager afterwards and installed modules for JavaScript, HTML, and CSS. You might have better luck going with JavaEE developer.

## Software Installation Done

At this point you should have the basic software needed to write code for the project. The next step is to get the source, set it up in your IDE and run it.

## Get Access to Source

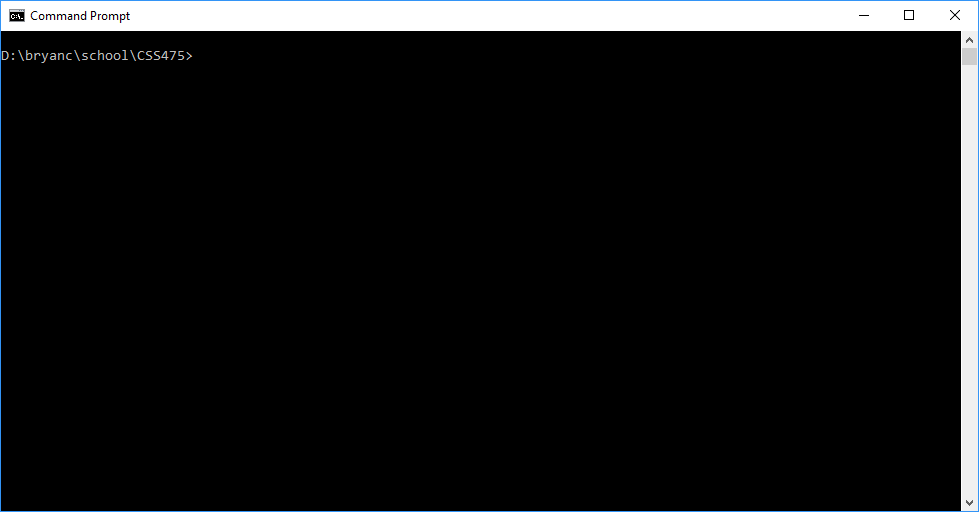
The project is stored on gitlab and is hosted here.

<https://gitlab.com/uwb-team-rocket>

If you don’t have access to the project, you need to create a gitlab account and have Bryan ([castillo.bryan@gmail.com](mailto:castillo.bryan@gmail.com)) add you as a user.

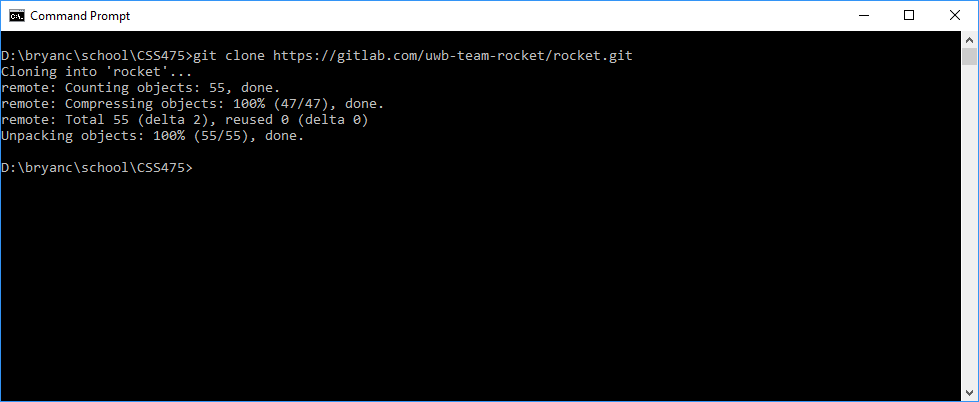
## Checkout Code

The next step is to checkout the code. Open a command line and go to a directory where you want the code to be on your local system.



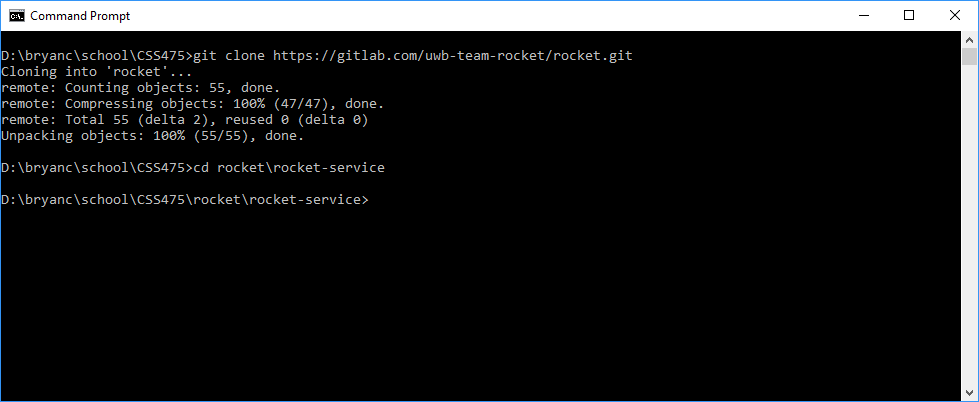
Next clone the repository using git:

git clone https://gitlab.com/uwb-team-rocket/rocket.git



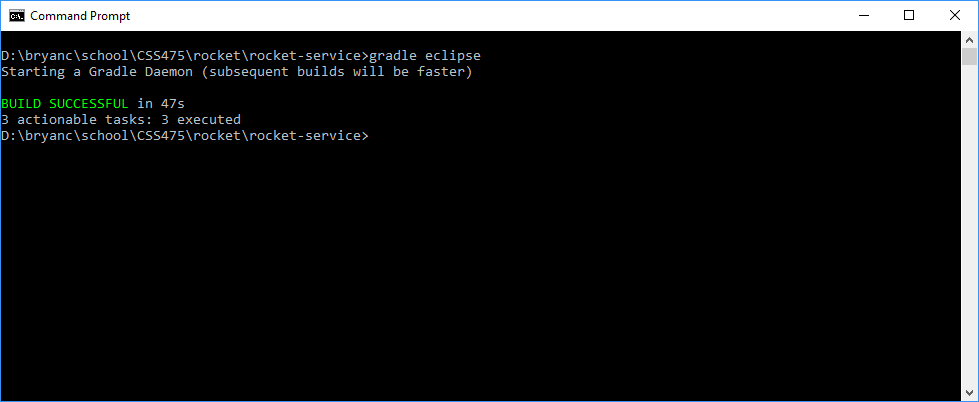
Next go into the rocket-service directory.

cd rocket\rocket-service



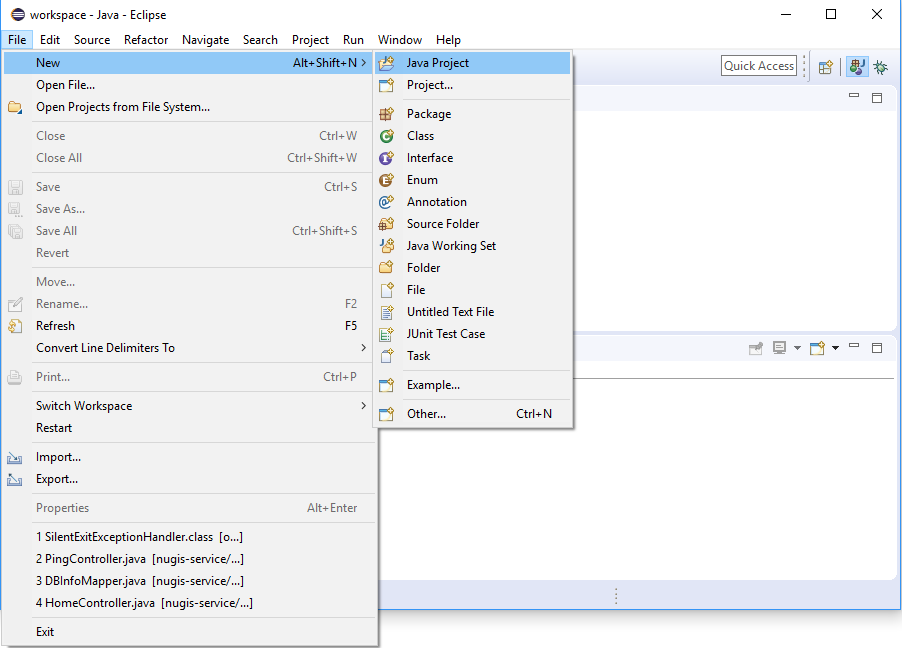
You can now use gradle to generate project files for your IDE:

* For Eclipse run: gradle eclipse

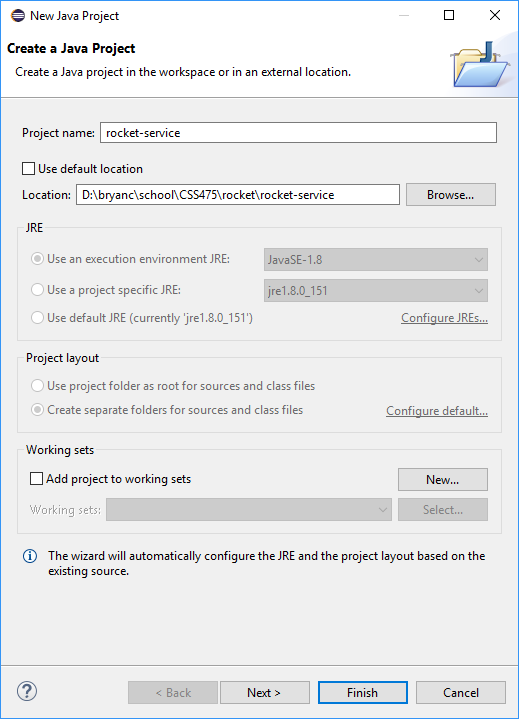


## Attach Eclipse to the Project

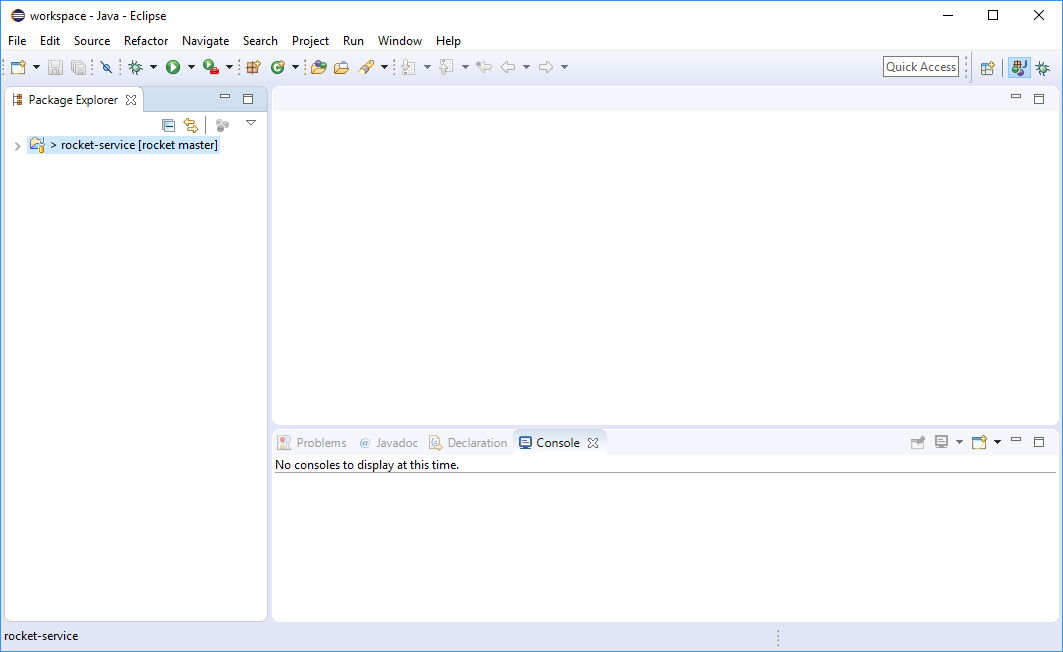
Choose to create a new Java Project



Uncheck “Use default location” and choose the rocket-service directory that you checked out. Then click Finish.

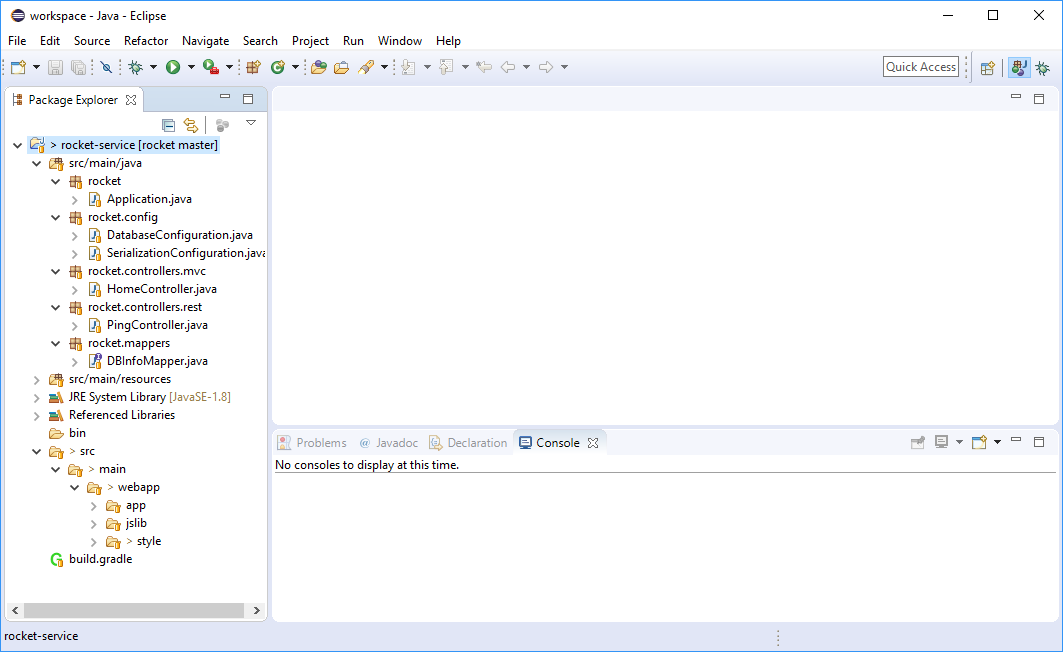


You should now have the project in the explorer on the left.



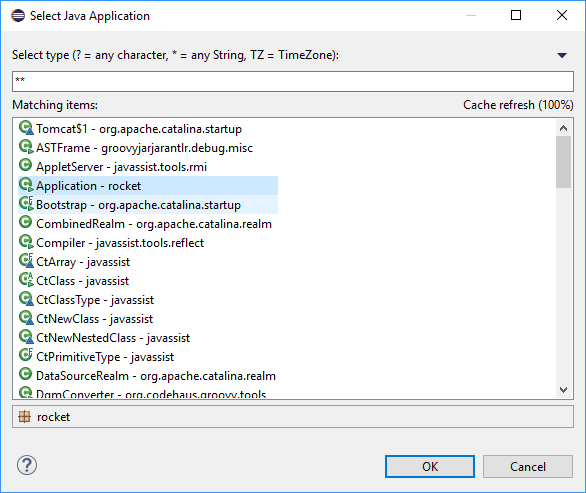
You should be able to start drilling into the project and you should be able to find:

* Java source under src/main/java
* Html, JS, and CSS files for the app under src/main/webapp/app

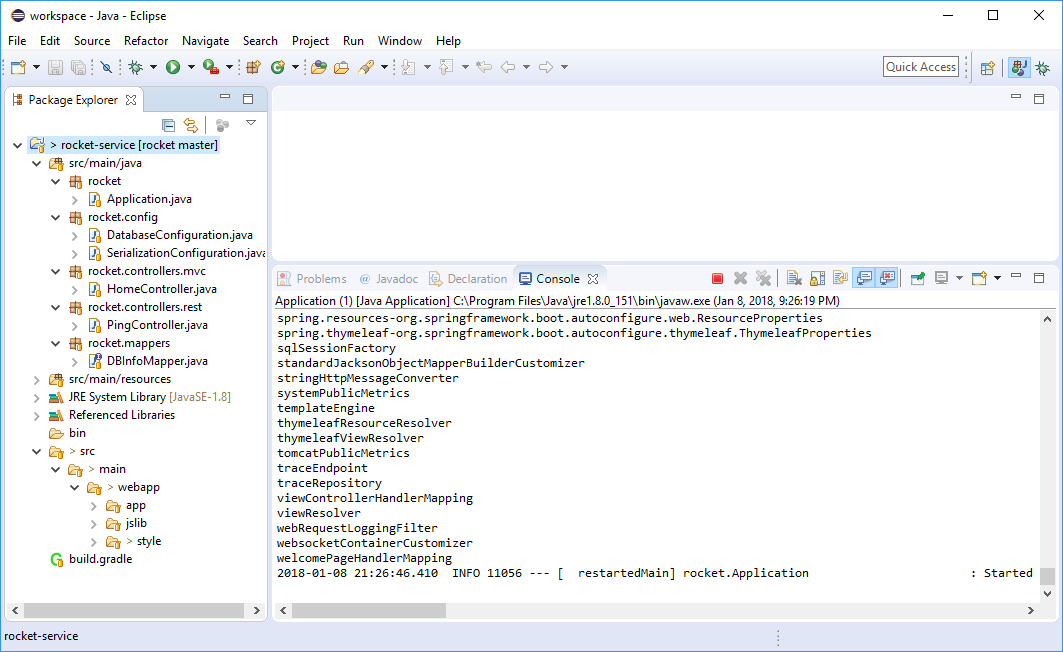


You should now be able to run the app from Eclipse. Chose Run from the Run menu.

The first time you might have to select the Java Application to run. Choose “Application – rocket”.

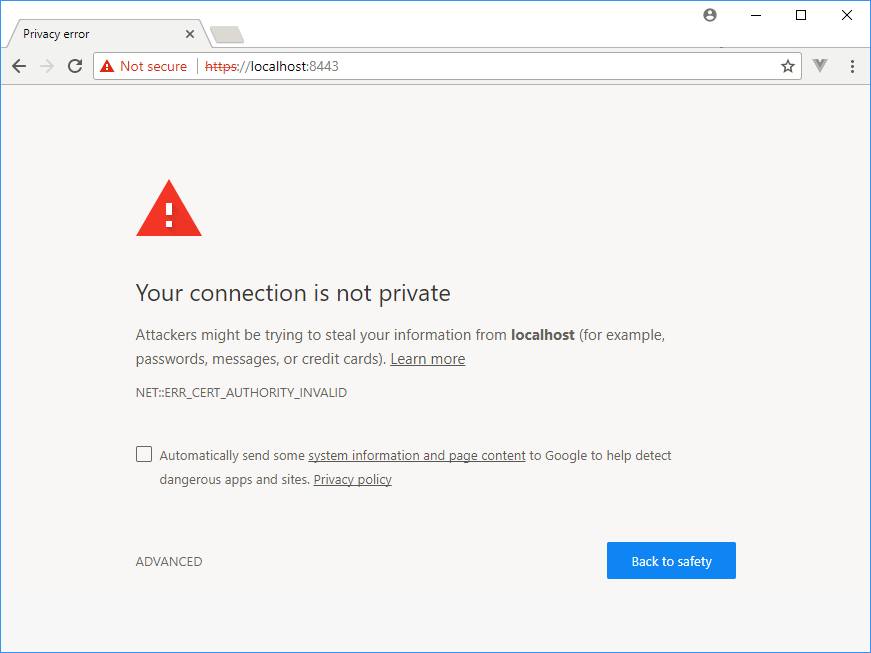


You should see some text start showing up in the Console window.



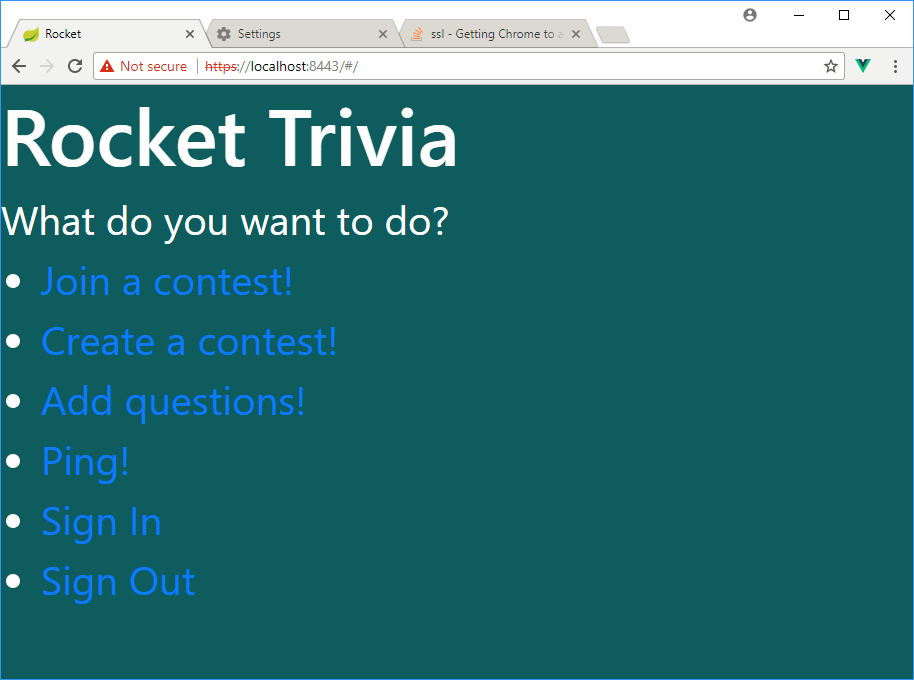
You should now be able to goto <https://localhost:8443/>

The site is running HTTPS with a self-signed certificate. The first time you bring up the site in your browser you will get a security warning.



Click “ADVANCED” and then proceed to site.

You should now see the rocket page. (For now, I created some stubs based on the Trivia Game idea. I haven’t put much into it, so we can still switch.)



Next click the Ping! link. It will call an API that hits the database from the application. It just gets the time from the database. This makes sure that you have everything configured properly.

