Running a Spring.IO project with XAMPP

Written by Gabriel Charlebois

Tools

- XAMPP (apache & mysql)
- INTELLIJ

Files

- ServerStarter.zip

mySQL

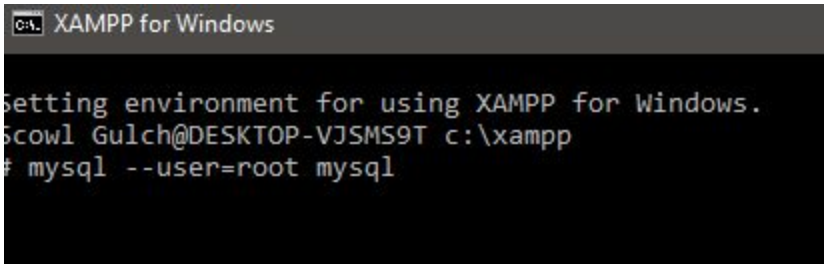
After installing XAMPP on your machine, start `xampp-control`

Make sure both Apache and MySQL are running, if not start them under actions. We do not need to toy with apache, but it is important to modify mySQL admin account's default values and to add a test database for our project.

We will do that by using the shell that is accessible from XAMPP.

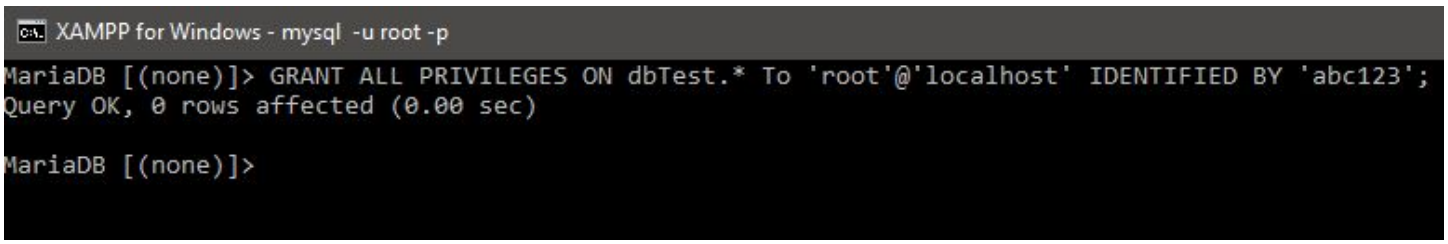
We want to start mySQL in admin mode (with a default install, this is the command).

```
mysql --user=root mysql
```



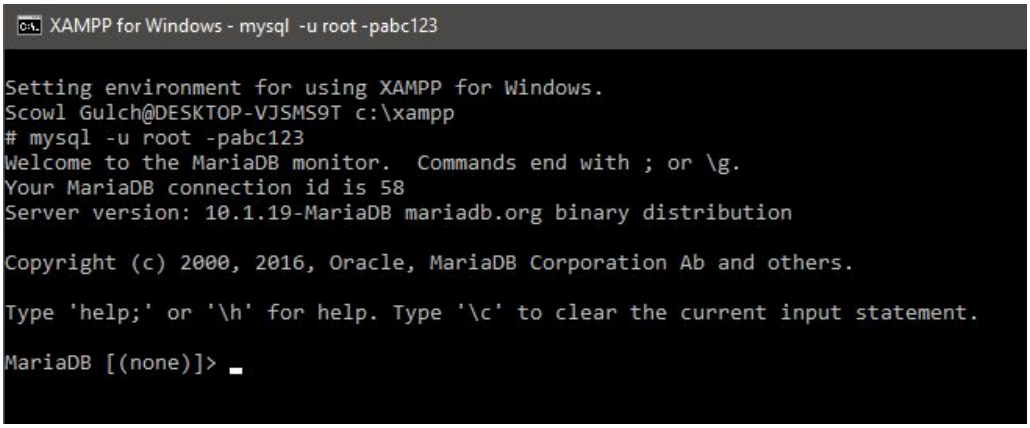
Now we change the default account's settings

```
GRANT ALL PRIVILEGES ON dbTest.* To 'root'@'localhost' IDENTIFIED BY 'abc123';
```



It's key to keep in mind that from now on, we will start mySQL command line with the new credentials, as the defaults have been overwritten.

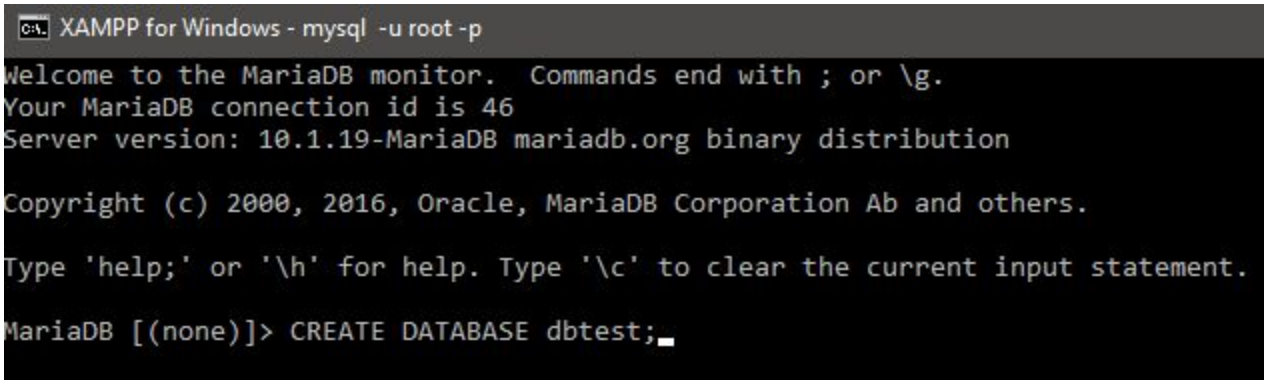
```
mysql -u root -pabc123
```



Important Note: there is no space between -p and the password. If you put one, mySQL will think you want to access database abc123.

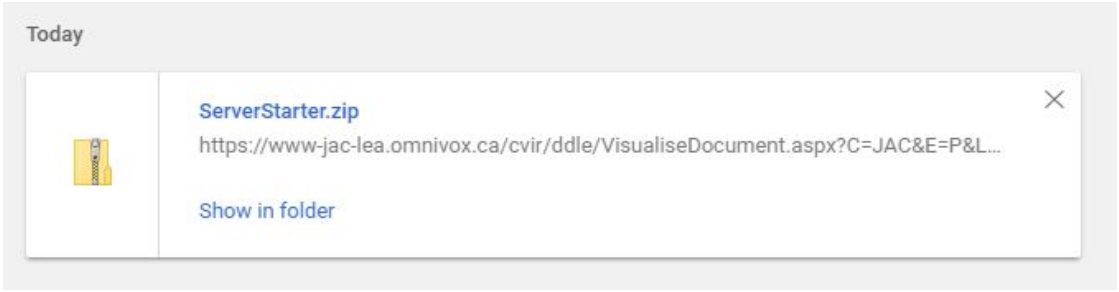
We add a test database to our mySQL installation.

```
CREATE DATABASE dbtest;
```



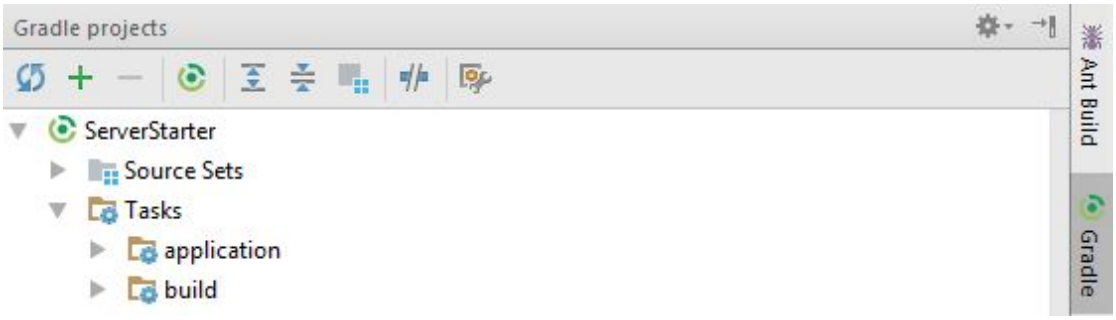
Spring.io

Unzip the ServerStarter.zip into the `AndroidProject/ServerStarter`



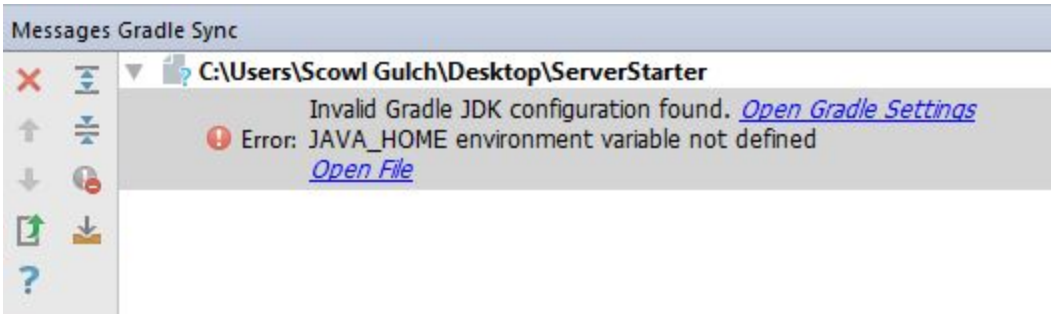
Start INTELLIJ and open the project

We want to refresh gradle, and will be refreshing it a few more times later on.

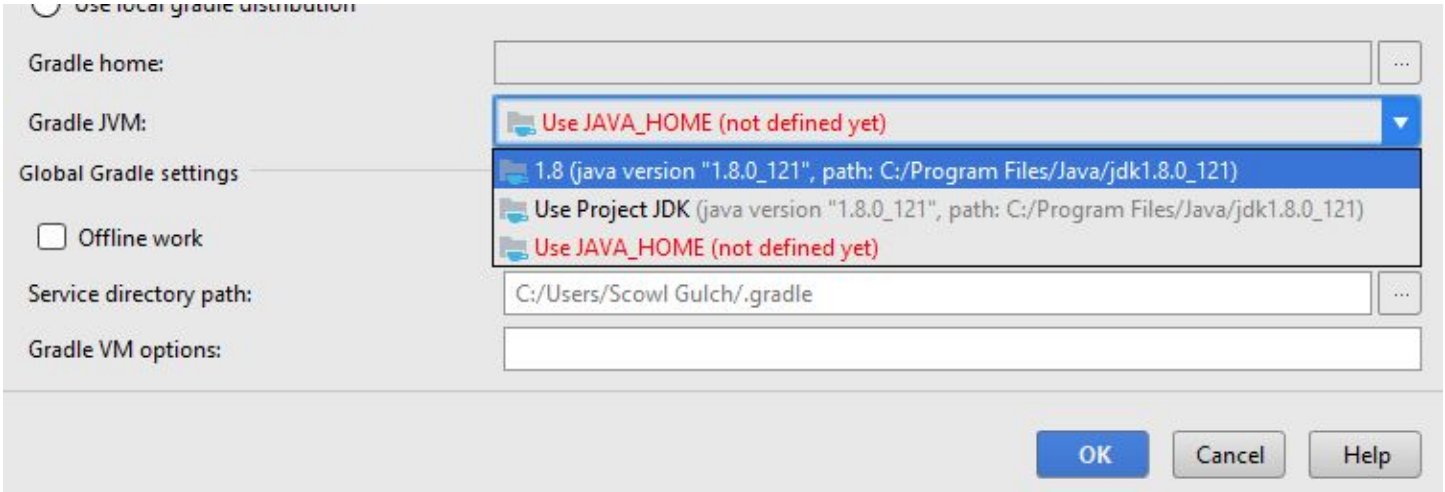


Now's the fun part, you want to error check to see what is wrong with the gradle project. The following are errors that I ran into, but for your installation it might differ. My advice: follow the default solutions that are given, worst case scenario you google.

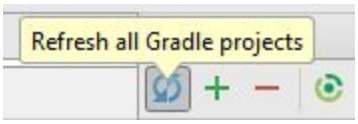
Your Java path might be incorrect, so click the link *Open Gradle Settings*



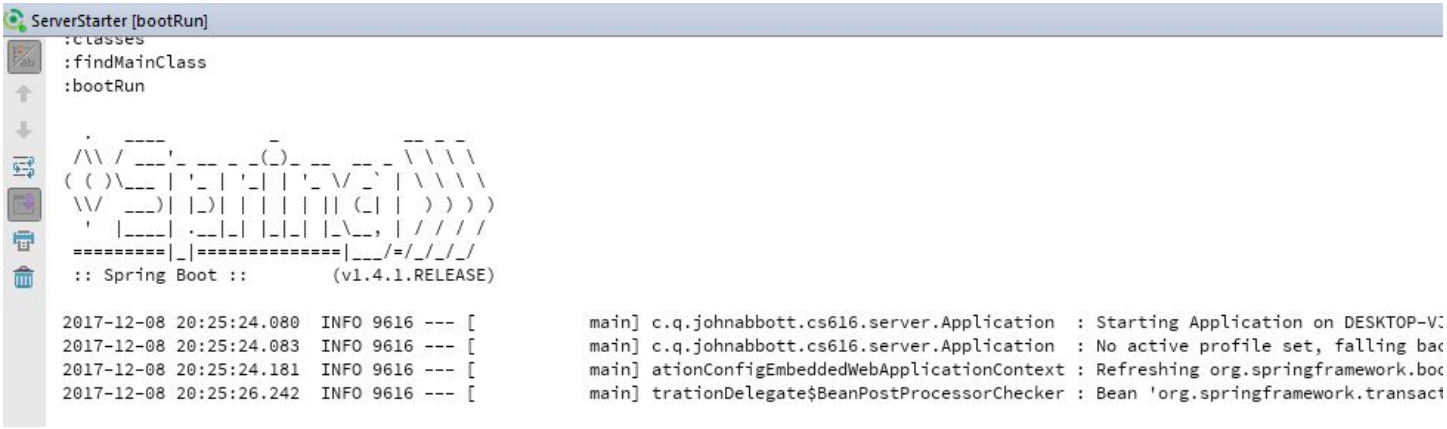
Change your Java path



Refresh gradle.



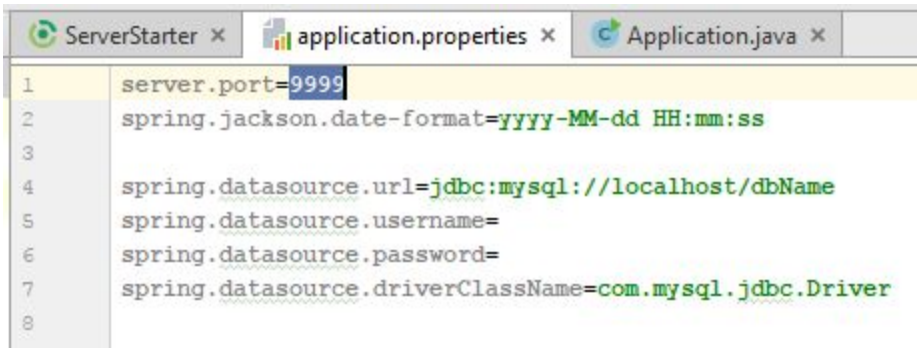
Run the configuration ServerStarter [bootRun]. You should see Spring's header.



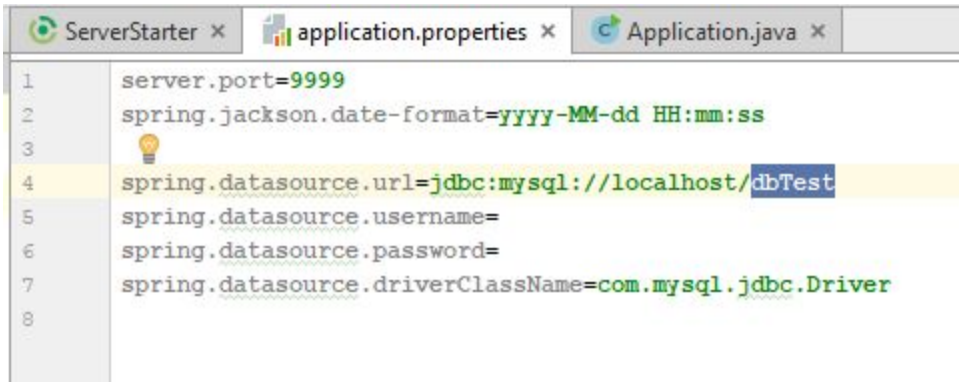
Errors will show up, but this is to make sure gradle did its job.

We will now change `application.properties` to fit the mySQL installation from earlier

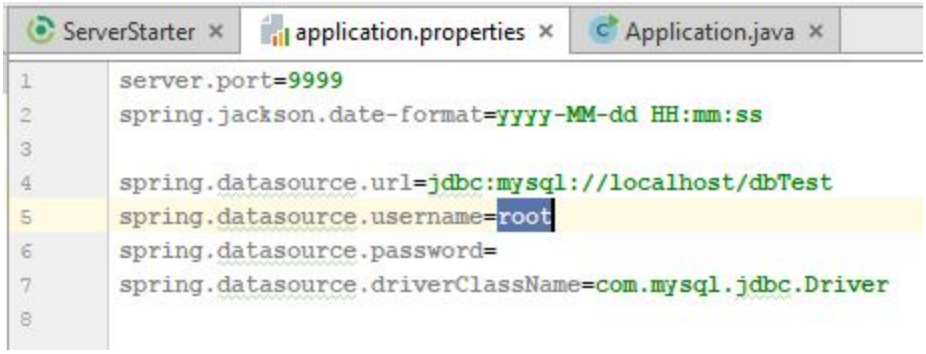
Changing the port is not necessary for now. Running this on Cloud9 would require you to use port 8081, but for our purposes, 9999 will do.



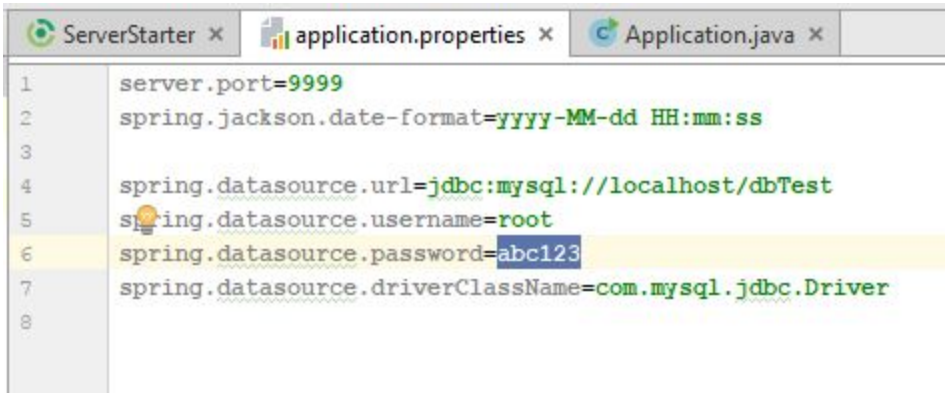
We want to set the URL of our repositories to fit the test database we've created.



Set the username to `root`



Set the password to whatever you set up earlier.



Save your changes to `applications.properties`

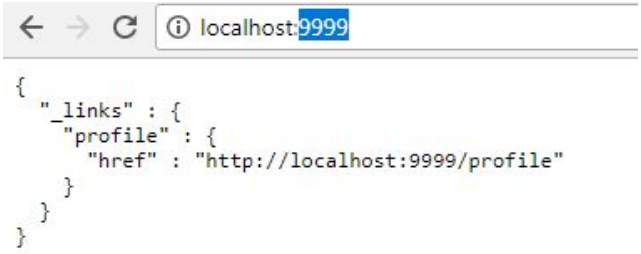
Run the configuration ServerStarter [bootRun]. Scroll down the console, you should see that our application has started.

```
main] o.s.d.r.w.RepositoryRestHandlerAdapter   : Looking for @ControllerAdvice: org.springframework.boot.context
main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.context
main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.Respon
main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.spring
main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto handler of type [class org.s
main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of type [class org.springfra
main] .m.m.a.ExceptionHandlerExceptionResolver : Detected @ExceptionHandler methods in repositoryRestExceptionHa
main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**/favicon.ico] onto handler of type [class
main] o.s.j.e.a.AnnotationMBeanExporter        : Registering beans for JMX exposure on startup
main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 9999 (http)
main] c.q.johnabbott.cs616.server.Application  : Started Application in 5.242 seconds (JVM running for 5.571)
```

This means that Spring.io is running on your system.

Final words

Try to access the repository in a browser



We can now make some changes to the database using the credentials we have set as well as updating the server starter with proper classes that Spring.io should recognize. You should be able to connect to the your computer's server from any other device on the same network by checking your ip in Windows' command line.

```
ipconfig
```