Spring boot command line arguments with Maven

Running application with Maven

To run your application with Maven you just run the following command:

```
mvn spring-boot:run
```

Specifying profiles

You can specify an active profile for you Spring boot application with the following command for mvn

```
mvn spring-boot:run -Dspring-boot.run.profiles=local,dev
```

Why using -D with mvn don't do anything

The Spring boot application is executed in a forked process, thus setting the properties with the command-line using -D for example will not affect the application. That is if you're running the application using maven.

How to set system property with maven

Like previously mentioned since the Spring boot application is executed in a forked process, if you set a system property using -D it will not be used in the actual application. To set a system property for the Spring boot application to use you will have to use

```
mvn spring-boot:run -Dspring-boot.run.jvmArguments="-Dproperty1=overridden"
```

This is also how you set debugging properties for when maven actually executes with Java.

What if you want to run with java

If instead you opt to build the jar using maven and then run the Spring boot application with `java` command, then you can provide in the -D and it will be used appropriately within the application and not ignored due to forked process.

However, when you are building the jar file you must also provide in the -D system properties. The active profile that you want to build for will need to be provided as well under `-Dspring.profiles.active` maven CLI argument.

Revision #1
Created 2023-04-08 00:20:01 UTC by Tamarine
Updated 2023-04-08 01:22:10 UTC by Tamarine