Profiling in spring boot using environment variables in local.

**What Are Profiles?**

Every enterprise application has many environments, like:

Dev | Test | Stage | Prod | UAT / Pre-Prod

Each environment requires a setting that is specific to them. For example, in DEV, we do not need to constantly check database consistency. Whereas in TEST and STAGE, we need to. These environments host specific configurations called Profiles.

We make properties files for each environment and set the profile in the application accordingly, so it will pick the respective properties file.

 Spring Boot — by default — provides just one property file ( application.properties).

we need to create three   more application.properties:

1. application-dev.properties
2. application-test.properties
3. application-prod.properties

The application.properties will remain as a master properties file,

**application.properties**

spring.profiles.active=${MYENV}

spring.application.name=Profiles

server.port=8087

**application-dev.properties**

app.message= This is the property file for the ${spring.application.name} specific to DEV Environment

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.url=jdbc:h2:mem:db;DB\_CLOSE\_DELAY=-1

spring.datasource.username=root

spring.datasource.password=root

**application-prod.properties**

app.message = This is the property file for the ${spring.application.name} specific to PRODUCTION Environment!! Be ALERT!!

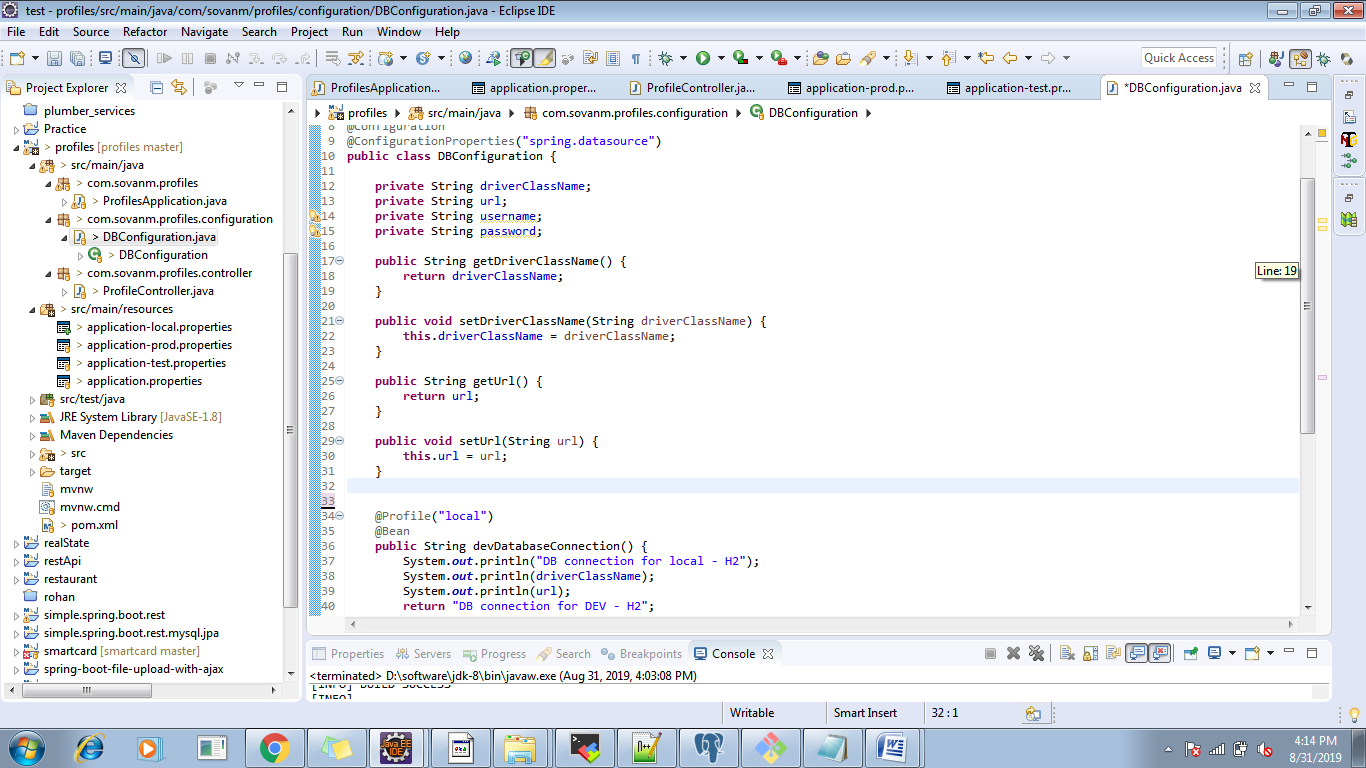
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/profiles

spring.datasource.username=root

spring.datasource.password=root

### Step1:Using @Profile on Configuration Class



#### Step:2 Environment Variable

Set up an environment variable spring\_profiles\_active.

