Spring Profiles:

What is a Profile?

A profile is **segregation between different runtime environments in which the Spring application can run.** For example, an application can run locally, in a development or test environment as well as in production. It is not easy to replicate the same hardware and software setup in all the above deployment environments. So the expectation is to configure the application in such a way that it must behave correctly in all these environments.

A profile is **not limited to only deployment environments**. It can be any type of segregation. For example, an application can connect to MySQL database in one profile while it will connect to H2 database in another profile. So here we have database configuration specific profiles. Similarly, profiles can be segregated on any basis.

So, in simple words, **the purpose of spring profiles is to activate different beans in different environments at bootstrap time.** Under the hood, a profile can control only two things:

* which beans are loaded into the application context
* which application properties are applied

@Profile("dev")

@Configuration

public class DevelopmentProfileConfiguration {

@Bean

public DataSource mySqlDataSource() {/\*...\*/}

}

Profile Specific Properties Files

Spring Boot allows grouping the configuration parameters for different environments into the different application.properties configuration files. Spring Boot will automatically pick up the right configuration file depending on the activated profile and load the configuration properties from that file.

Properties files have to be named in the format application-{profile}.properties. For example,

application.properties – contains properties applicable to all environments

application-local.properties – configure the application when local profile is activated

application-dev.properties – configure the application when dev profile is activated

application-prod.properties – configure the application when prod profile is activated

application-local.properties

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

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

spring.datasource.username=sa

spring.datasource.password=

**Multiple Profiles in *application.properties***

It is also possible to specify all the profile specific configurations in only the*application.properties* file using the spring.config.activate.on-profile property as the separator to indicate the profile.

application.properties

**application.name**=My Application

*#--- local profile ---#*

**spring.config.activate.on-profile**=local

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

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

**spring.datasource.username**=sa

**spring.datasource.password**=

*#--- dev profile ---#*

**spring.config.activate.on-profile**=dev

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

**spring.datasource.url**=jdbc:mysql://localhost:3306/db

**spring.datasource.username**=root

**spring.datasource.password**=password