While developing an application I got requirement that our application will have different environments (ex: development, test and production) and based on environment the application take corresponding environment values).

To provide solution for this requirement we can achieve in following in three ways.

1. By creating Maven profile
2. By configuring properties in Web Server or Application
3. By using Spring profile

I have opted to third approach to address my requirement.

**Question:** Why not Maven

**Answer:** To answer your question

1. If the project is not in maven then for achieving environment handling then we need to convert project to maven project.

2. If the project is maven but for creating profile we need to learn and do some exercise to lean maven profiles.

**Question:** Why not configuring properties in Web Server or Application Server

**Answer:** If the application is having one or two properties then we can add it. But if the application is having multiple properties say for example (email properties, database properties and application have additional properties like’s image paths, video paths and etc.)

Then configuring these properties in all environments is a tedious job.

Because the database, email, application properties of development will vary from test and production.

If we configure mistakenly then application will not behave as expected.

**Question:** Why spring profile.

**Answer:** Spring Profiles provide a way to segregate parts of your application configuration and make it only available in certain environments.

I have uploaded the spring profile sample on github and here is the link for example application.

**Question: How to run spring profile in Java application?**

**Answer :**

While running the application you can specify –VM arguments

Ex: -Dspring.profiles.active=development or test or product

**Question: How to run spring profile in Web application**?

**Answer :**

**1) Development Environment.**

1. We need to add this property to your Tomcat **catalina.properties** file

# Setting a property value

spring.profiles.active=development

2. Generate war and deploy to Development Environment.

**2) QA Environment.**

1. We need to add this property to your Tomcat **catalina.properties** file

# Setting a property value

spring.profiles.active=test

2. Generate war and deploy to QA Environment.

**3) Production Environment**

1 We need to add this property to your Tomcat **catalina.properties** file

# Setting a property value

spring.profiles.active=production

2. Generate war and deploy to Production Environment.

**Note:** If you want deploy war automatically then choose Jenkins or EC