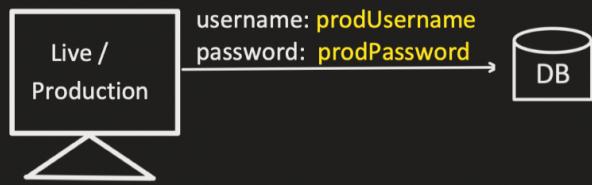
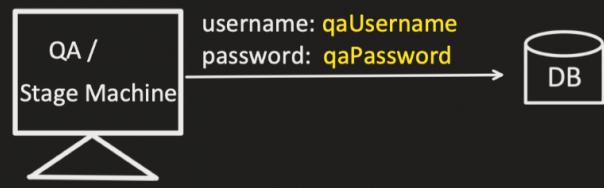
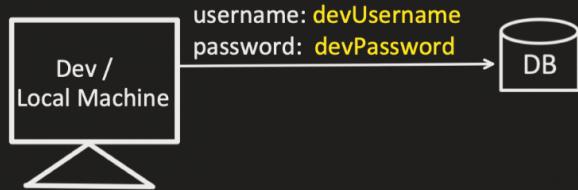


## Spring boot: @Profile

Let's deep dive into **@Profile** annotation to understand it better



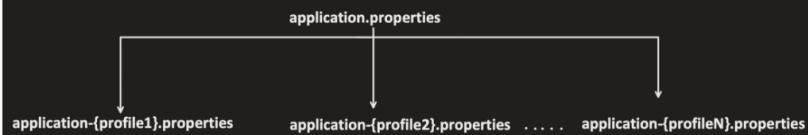
This "username", "password" is just one example.  
There are so many other configuration, which are different for different environments, like:

- URL and Port number
- Connection timeout values
- Request timeout values
- Throttle values
- Retry values etc.

### How to do it?

We put the configurations in "application.properties" file.  
But how to handle, different environment configurations?

That's where **Profiling** comes into the picture



**application.properties**

```

@Component
public class MySQLConnection {
    @Value("${username}")
    String username;

    @Value("${password}")
    String password;

    @PostConstruct
    public void init(){
        System.out.println("username: " + username + " password: " + password);
    }
}

```

**application-dev.properties**

```

username=devUsername
password=devPassword

```

**application-qa.properties**

```

username=qByUsername
password=q>Password

```

**application-prod.properties**

```

username=prodUsername
password=prodPassword

```

2024-06-01T22:55:25.548+05:30 INFO 48119 --- [main] c.c.l.SpringbootApplication : The following 1 profile is active: "dev"
2024-06-01T22:55:25.549+05:30 INFO 48119 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port: 8080 (http)
2024-06-01T22:55:25.549+05:30 INFO 48119 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-01T22:55:25.549+05:30 INFO 48119 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-06-01T22:55:25.549+05:30 INFO 48119 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Embedded WebApplicationContext
2024-06-01T22:55:25.549+05:30 INFO 48119 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : [Tomcat] started on port 8080 (http) with context path ''
2024-06-01T22:55:25.549+05:30 INFO 48119 --- [main] c.c.l.SpringbootApplication : Started SpringbootApplication in 0.993 seconds (process running for 0.993)

And during application startup, we can tell Spring Boot to pick specific "application properties" file, Using "spring.profiles.active" configuration.

**application.properties**

```

@Component
public class MySQLConnection {
    @Value("${username}")
    String username;

    @Value("${password}")
    String password;

    @PostConstruct
    public void init(){
        System.out.println("username: " + username + " password: " + password);
    }
}

```

**application-dev.properties**

```

username=devUsername
password=devPassword
spring.profiles.active=qa

```

**application-qa.properties**

```

username=qByUsername
password=q>Password

```

**application-prod.properties**

```

username=prodUsername
password=prodPassword

```

2024-06-02T08:04:24.009+05:30 INFO 52175 --- [main] c.c.l.SpringbootApplication : The following 1 profile is active: "qa"
2024-06-02T08:04:24.009+05:30 INFO 52175 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port: 8080 (http)
2024-06-02T08:04:24.009+05:30 INFO 52175 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-02T08:04:24.009+05:30 INFO 52175 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-06-02T08:04:24.009+05:30 INFO 52175 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Embedded WebApplicationContext
2024-06-02T08:04:24.031+05:30 INFO 52175 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Root WebApplicationContext: initialization completed in 413 ms
2024-06-02T08:04:24.159+05:30 INFO 52175 --- [main] o.a.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
2024-06-02T08:04:24.159+05:30 INFO 52175 --- [main] c.c.l.SpringbootApplication : Started SpringbootApplication in 0.78 seconds (process running for 0.993)

We can pass the value to this configuration "spring.profiles.active" during application startup itself.

`mvn spring-boot:run -Dspring-boot.run.profiles=prod`

or

Add this in `Pom.xml`

**Pom.xml**

```

<profiles>
    <profile>
        <id>local</id>
        <properties>
            <spring-boot.run.profiles>dev</spring-boot.run.profiles>
        </properties>
    </profile>
    <profile>
        <id>production</id>
        <properties>
            <spring-boot.run.profiles>prod</spring-boot.run.profiles>
        </properties>
    </profile>
    <profile>
        <id>qa</id>
        <properties>
            <spring-boot.run.profiles>qa</spring-boot.run.profiles>
        </properties>
    </profile>
</profiles>

```

2024-06-02T08:21:57.336+05:30 INFO 53096 --- [main] c.c.l.SpringbootApplication : The following 1 profile is active: "prod"
2024-06-02T08:21:57.647+05:30 INFO 53096 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port: 8080 (http)
2024-06-02T08:21:57.652+05:30 INFO 53096 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-02T08:21:57.652+05:30 INFO 53096 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-06-02T08:21:57.673+05:30 INFO 53096 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Embedded WebApplicationContext
2024-06-02T08:21:57.673+05:30 INFO 53096 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 318 ms
username: prodUsername password: prodPassword
2024-06-02T08:21:57.804+05:30 INFO 53096 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''

`mvn spring-boot:run -Production`

So, now we know, what's Profiling is. Let's see what is `@Profile` annotation.

Using `@Profile` annotation, we can tell Spring Boot to create bean only when particular profile is set.

**application.properties**

```

username=defaultUsername
password=defaultPassword

```

**application-dev.properties**

```

username=devUsername
password=devPassword

```

**application-qa.properties**

```

username=qByUsername
password=q>Password

```

**application-prod.properties**

```

username=prodUsername
password=prodPassword

```

**MySQLConnection.java**

```

@Component
@Profile("prod")
public class MySQLConnection {
    @Value("${username}")
    String username;

    @Value("${password}")
    String password;

    @PostConstruct
    public void init(){
        System.out.println("MySQL: " + username + " " + password);
    }
}

```

**NoSQLConnection.java**

```

@Component
@Profile("qa")
public class NoSQLConnection {
    @Value("${username}")
    String username;

    @Value("${password}")
    String password;
}

```

```

@PostConstruct
public void init(){
    System.out.println("NoSQL username: " + username + " password: " + password);
}

```

<b>application-dev.properties</b>	<b>application-qa.properties</b>	<b>application-prod.properties</b>
username=devUsername password=devPassword	username=qaUsername password=qaPassword	username=prodUsername password=prodPassword

mvn spring-boot:run

```

2024-06-02T00:38:44.935+05:30 INFO 55848 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-06-02T00:38:44.940+05:30 INFO 55848 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-02T00:38:44.941+05:30 INFO 55848 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-06-02T00:38:44.944+05:30 INFO 55848 --- [main] o.a.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-06-02T00:38:44.944+05:30 INFO 55848 --- [main] o.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 410 ms
2024-06-02T00:38:47.109+05:30 INFO 55848 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''

```

mvn spring-boot:run -Dspring-boot.run.profiles=prod

```

2024-06-02T00:42:30.543+05:30 INFO 55216 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-06-02T00:42:30.543+05:30 INFO 55216 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-02T00:42:30.548+05:30 INFO 55216 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-06-02T00:42:30.549+05:30 INFO 55216 --- [main] o.a.c.c.[Tomcat].[localhost]. [/] : Initializing Spring embedded WebApplicationContext
2024-06-02T00:42:30.549+05:30 INFO 55216 --- [main] o.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 327 ms
MySQL username: prodUsername password: prodPassword
2024-06-02T00:42:30.550+05:30 INFO 55216 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''

```

We can also set multiple profiles at a time.

```

@Component
@Profile("prod")
public class NoSQLConnection {
    @Value("${username}")
    String username;
    @Value("${password}")
    String password;

    @PostConstruct
    public void init(){
        System.out.println("NoSQL username: " + username + " password: " + password);
    }
}

@Component
@Profile("qa")
public class NoSQLConnection {
    @Value("${username}")
    String username;
    @Value("${password}")
    String password;

    @PostConstruct
    public void init(){
        System.out.println("NoSQL username: " + username + " password: " + password);
    }
}

2024-06-02T00:42:36.744+05:30 INFO 56858 --- [main] o.a.c.c.[Tomcat].[localhost]. [/] : The following 2 profiles are active: "prod", "qa"
2024-06-02T00:42:37.149+05:30 INFO 56858 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-06-02T00:42:37.154+05:30 INFO 56858 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-02T00:42:37.155+05:30 INFO 56858 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-06-02T00:42:37.179+05:30 INFO 56858 --- [main] o.a.c.c.[Tomcat].[localhost]. [/] : Initializing Spring embedded WebApplicationContext
2024-06-02T00:42:37.179+05:30 INFO 56858 --- [main] o.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 433 ms
MySQL username: qaUsername password: qaPassword
2024-06-02T00:42:37.327+05:30 INFO 56858 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''

```

<b>application-dev.properties</b>	<b>application-qa.properties</b>	<b>application-prod.properties</b>
username=defaultUsername password=defaultPassword spring.profiles.active=prod,qa	username=qaUsername password=qaPassword	username=prodUsername password=prodPassword

Now lets come back to the previous question, which I asked:

You have 2 Application and 1 common code base, how will you make sure that, BEAN is only created for 1 Application, not for other?

### Common Codebase

```

@Component
@Profile("app1")
public class NoSQLConnection {
    @Value("${username}")
    String username;
    @Value("${password}")
    String password;

    @PostConstruct
    public void init(){
        System.out.println("NoSQL username: " + username + " password: " + password);
    }
}

```

**Application1**  
(*application.properties*)

spring.profiles.active=app1

**Application2**  
(*application.properties*)

spring.profiles.active=app2