Profiles: Spring provides a great tool for grouping configurations properties into so called profiles(dev, qa, prod) allowing us to activate a bunch of configurations based on the active profile.

Profiles are perfect for setting up our application for different enviourments, but they are also being used in another use cases like bean creation based on a profile etc.

So basically a profile can influence the application properties loaded and beans which are loaded into the Spring context.

Note: The default profile is always active. Spring Boot loads all properties in application.properties into the default profiles.

We can create another profiles by creating properties file like below.

1. application\_qa.properties (QA Env)

2. application\_prod.properties (Prod Env)

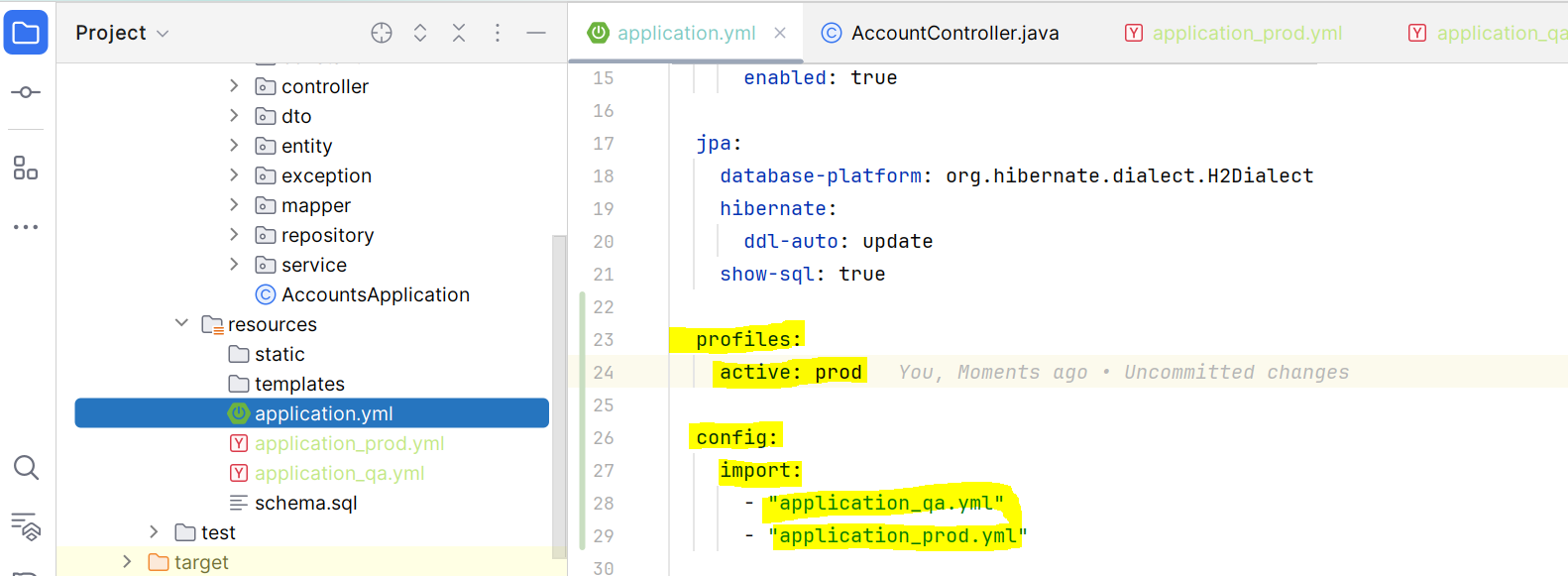
We can activate a specific profile by using **spring.profiles.active** property like below.

**Spring.profiles.active=prod**

**Note:** An important point to consider is that once a application is built and packaged, it should not be modified. If any configuration changes are required, such as updating credentials or database handles, they should be made externally.

**1. Define configuration in application.yml**

spring:   
 profiles:  
 active: prod  
  
 config:  
 import:  
 - "application\_qa.yml"  
 - "application\_prod.yml"



**2. Mark below entry in application\_qa and application\_prod file**

spring:

config:  
 activate:  
 on-profile: prod/qa

