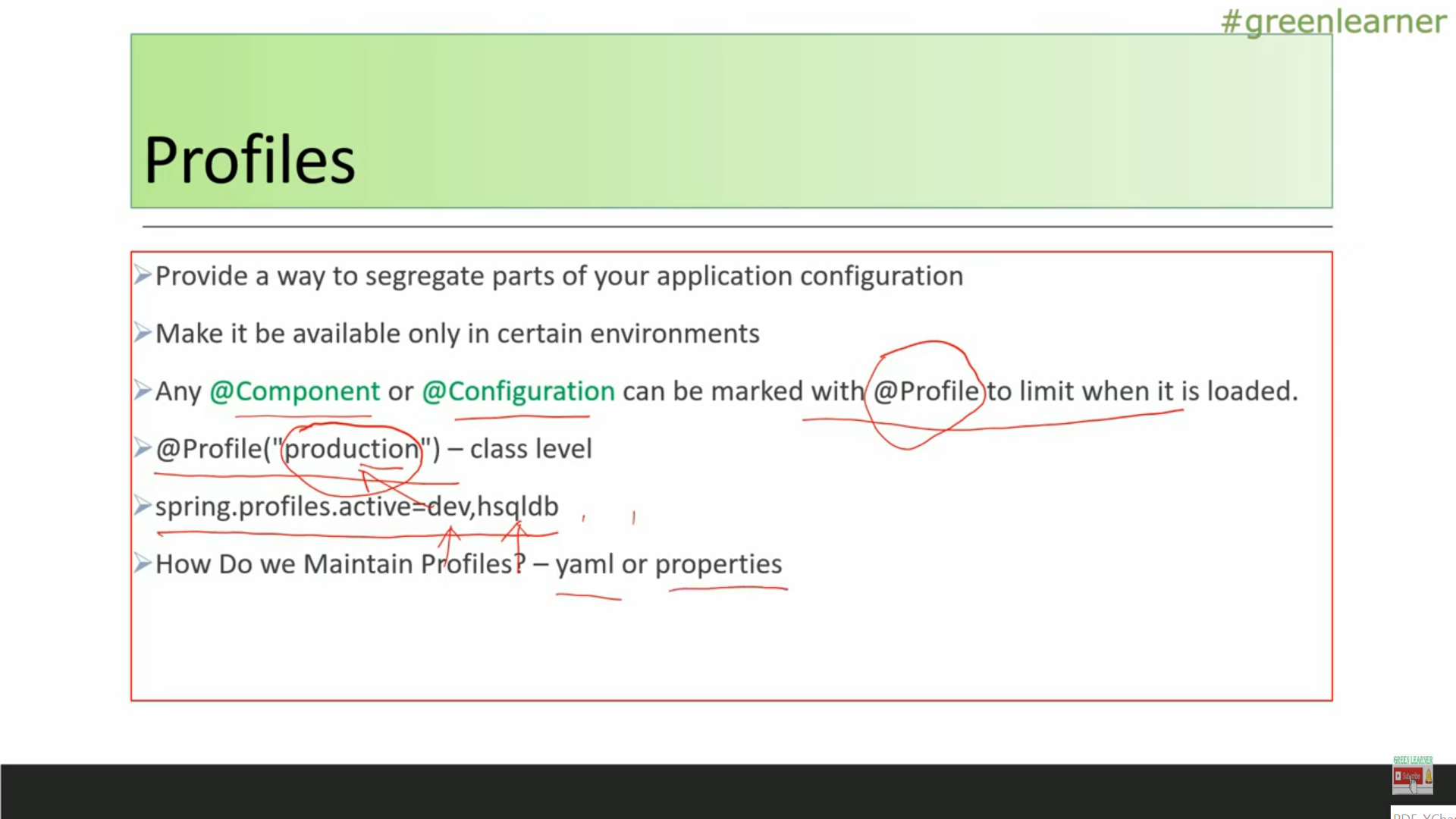
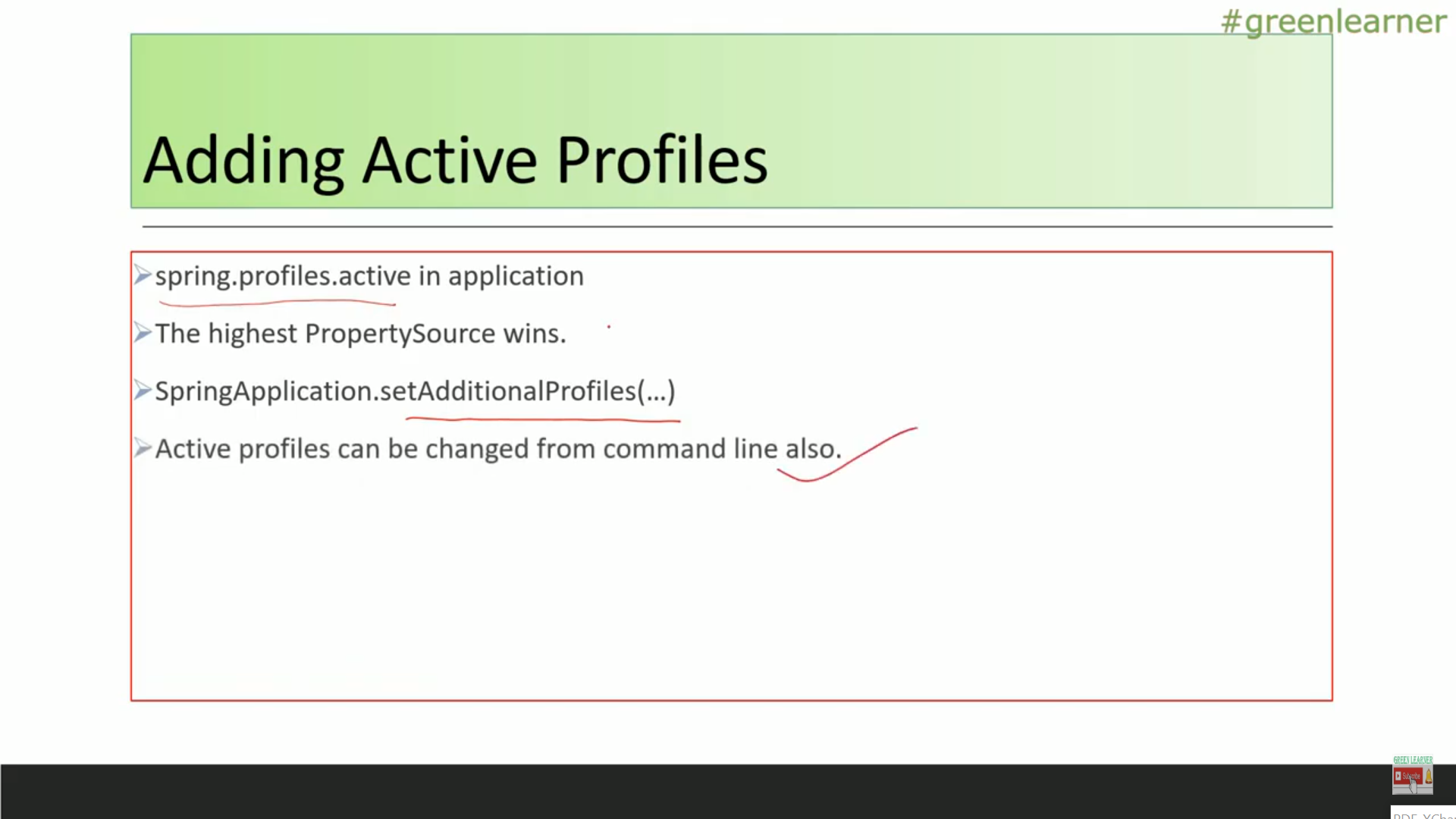


# Microservices with Spring Boot #9 || Profiles in Spring Boot || How to use profiles in spring boot

Ur application first runs in DEV environments…then it will run in TEST env…and finally it will run in PROD…

How do we do that without even touching or changing the code…





Just creating the @Bean with @Profile(“dev”)…..

Application.yaml

app:  
 server:  
 detail: this is general server  
 name: geenral  
server:  
 port: 8083  
spring:  
 application:  
 name: This is the Demo project for spring Profiles  
 datasource:  
 password: 12345  
 url: jdbc:hsqldb:hsql://localhost/geenral  
 username: geenral-user  
 profiles:  
 active: dev

application-dev.yaml

app:  
 server:  
 detail: this is tomcat for dev  
 name: tomcat  
spring:  
 datasource:  
 password: 12345  
 url: jdbc:hsql:hsql://localhost/enrolments  
 username: dev

application-prod.yaml

app:  
 server:  
 detail: this is glassfish for production  
 name: glassfish  
spring:  
 datasource:  
 password: heybabe  
 url: jdbc:oracle:thin:@hostname:80:rocky  
 username: oracle-prod

application-test.yaml

app:  
 server:  
 detail: this is jboss for test  
 name: jboss  
spring:  
 datasource:  
 password: arvsd  
 url: jdbc:mysql://localhost:3306/rock  
 username: mysql

package com.always.learning.profilesDemo;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class ProfilesDemoApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(ProfilesDemoApplication.class, args);  
 }  
  
}

package com.always.learning.profilesDemo;  
  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.stereotype.Component;  
  
@Component  
public class EnvDetails  
{  
 @Value("${app.server.name}")  
 private String appServerName;  
 @Value("${app.server.detail}")  
 private String appServerDetail;  
 @Value("${spring.datasource.url}")  
 private String dbUrl;  
 @Value("${spring.datasource.username}")  
 private String dbUser;  
 @Value("${spring.datasource.password}")  
 private String dbPassword;  
  
 public String getAppServerName() {  
 return appServerName;  
 }  
  
 public void setAppServerName(String appServerName) {  
 this.appServerName = appServerName;  
 }  
  
 public String getAppServerDetail() {  
 return appServerDetail;  
 }  
  
 public void setAppServerDetail(String appServerDetail) {  
 this.appServerDetail = appServerDetail;  
 }  
  
 public String getDbUrl() {  
 return dbUrl;  
 }  
  
 public void setDbUrl(String dbUrl) {  
 this.dbUrl = dbUrl;  
 }  
  
 public String getDbUser() {  
 return dbUser;  
 }  
  
 public void setDbUser(String dbUser) {  
 this.dbUser = dbUser;  
 }  
  
 public String getDbPassword() {  
 return dbPassword;  
 }  
  
 public void setDbPassword(String dbPassword) {  
 this.dbPassword = dbPassword;  
 }  
  
 @Override  
 public String toString() {  
 return "EnvDetails{" +  
 "appServerName='" + appServerName + '\'' +  
 ", appServerDetail='" + appServerDetail + '\'' +  
 ", dbUrl='" + dbUrl + '\'' +  
 ", dbUser='" + dbUser + '\'' +  
 ", dbPassword='" + dbPassword + '\'' +  
 '}';  
 }  
}

package com.always.learning.profilesDemo;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.annotation.Profile;  
  
@Configuration  
public class AppConfig  
{  
 @Autowired  
 private EnvDetails envDetails;  
  
 @Bean  
 @Profile("dev")  
 public EnvDetails devEnv()  
 {  
 System.*out*.println("Dev Environments Stuff ....");  
 System.*out*.println(envDetails);  
 return envDetails;  
 }  
  
 @Bean  
 @Profile("test")  
 public EnvDetails testEnv()  
 {  
 System.*out*.println("Test Environments Stuff ....");  
 System.*out*.println(envDetails);  
 return envDetails;  
 }  
  
 @Bean  
 @Profile("prod")  
 public EnvDetails prodEnv()  
 {  
 System.*out*.println("Prod Environments Stuff ....");  
 System.*out*.println(envDetails);  
 return envDetails;  
 }  
}

console:--

16:32:00: Executing task 'ProfilesDemoApplication.main()'...

> Task :compileJava UP-TO-DATE

> Task :processResources

> Task :classes

> Task :ProfilesDemoApplication.main()

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v2.3.1.RELEASE)

2020-07-06 16:32:04.175 INFO 26880 --- [ main] c.a.l.p.ProfilesDemoApplication : Starting ProfilesDemoApplication on IN2393357W1 with PID 26880 (C:\GIT\_REPO\profilesDemo\build\classes\java\main started by Kaushal.Kumar1 in C:\GIT\_REPO\usermanagementapp)

2020-07-06 16:32:04.178 INFO 26880 --- [ main] c.a.l.p.ProfilesDemoApplication : The following profiles are active: dev

Dev Environments Stuff ....

EnvDetails{appServerName='tomcat', appServerDetail='this is tomcat for dev', dbUrl='jdbc:hsql:hsql://localhost/enrolments', dbUser='dev', dbPassword='12345'}

2020-07-06 16:32:04.701 INFO 26880 --- [ main] c.a.l.p.ProfilesDemoApplication : Started ProfilesDemoApplication in 0.991 seconds (JVM running for 1.619)

BUILD SUCCESSFUL in 4s

3 actionable tasks: 2 executed, 1 up-to-date

16:32:04: Task execution finished 'ProfilesDemoApplication.main()'.

**How can we change the profile outside of this application without changing the code…**

C:\GIT\_REPO\profilesDemo>gradlew clean build

C:\GIT\_REPO\profilesDemo>cd build/libs

C:\GIT\_REPO\profilesDemo\build\libs>dir

Directory of C:\GIT\_REPO\profilesDemo\build\libs

06-07-2020 17:21 <DIR> .

06-07-2020 17:21 <DIR> ..

06-07-2020 17:21 8,209,955 profilesDemo-0.0.1-SNAPSHOT.jar

1 File(s) 8,209,955 bytes

2 Dir(s) 200,797,683,712 bytes free

**If I run this command:-**

**java -jar profilesDemo-0.0.1-SNAPSHOT.jar**

**it runs the application with default changes…**

2020-07-06 17:28:57.955 INFO 34332 --- [ main] c.a.l.p.ProfilesDemoApplication : Starting ProfilesDemoApplication on IN2393357W1 wit

h PID 34332 (C:\GIT\_REPO\profilesDemo\build\libs\profilesDemo-0.0.1-SNAPSHOT.jar started by Kaushal.Kumar1 in C:\GIT\_REPO\profilesDemo\build\libs)

2020-07-06 17:28:57.965 INFO 34332 --- [ main] c.a.l.p.ProfilesDemoApplication : The following profiles are active: prod

Prod Environments Stuff ....

EnvDetails{appServerName='glassfish', appServerDetail='this is glassfish for production', dbUrl='jdbc:oracle:thin:@hostname:80:rocky', dbUser='oracle-pr

od', dbPassword='heybabe'}

2020-07-06 17:29:00.694 INFO 34332 --- [ main] c.a.l.p.ProfilesDemoApplication : Started ProfilesDemoApplication in 4.847 seconds (J

VM running for 6.63)

If I run this command

C:\GIT\_REPO\profilesDemo\build\libs>java -jar profilesDemo-0.0.1-SNAPSHOT.jar --spring.profiles.active=dev…in application.properties profile is prod…but from cmd providing dev…lets see first…

spring:  
 application:  
 name: This is the Demo project for spring Profiles  
 datasource:  
 password: 12345  
 url: jdbc:hsqldb:hsql://localhost/geenral  
 username: geenral-user  
 profiles:  
 active: prod

try this---

C:\GIT\_REPO\profilesDemo\build\libs>java -jar profilesDemo-0.0.1-SNAPSHOT.jar --spring.profiles.active=dev

now see the output…

2020-07-06 17:39:34.678 INFO 5596 --- [ main] c.a.l.p.ProfilesDemoApplication : Starting ProfilesDemoApplication on IN2393357W1 with

PID 5596 (C:\GIT\_REPO\profilesDemo\build\libs\profilesDemo-0.0.1-SNAPSHOT.jar started by Kaushal.Kumar1 in C:\GIT\_REPO\profilesDemo\build\libs)

2020-07-06 17:39:34.687 INFO 5596 --- [ main] c.a.l.p.ProfilesDemoApplication : The following profiles are active: dev

Dev Environments Stuff ....

EnvDetails{appServerName='tomcat', appServerDetail='this is tomcat for dev', dbUrl='jdbc:hsql:hsql://localhost/enrolments', dbUser='dev', dbPassword='12345'}

2020-07-06 17:39:37.101 INFO 5596 --- [ main] c.a.l.p.ProfilesDemoApplication : Started ProfilesDemoApplication in 4.213 seconds (JV

M running for 6.152)

C:\GIT\_REPO\profilesDemo\build\libs>