Spring Boot *..:: Quick RECAP ::..*

- Approccio «opinionated» all'uso del framework Spring e altre librerie evitando codice ripetitivo
- Auto Configuration
- Maven Starters
  - **`spring-boot-starter-web`**
  - **`spring-boot-starter-data-jdbc`**
  - **`spring-boot-starter-data-jpa`**
- Annotazioni java
  - **`@SpringBootApplication`**
  - **`@Component / @Bean / @Repository / @Service`**
  - **`@Autowired`**
  - **`@RestController`**
  - **`@RequestMapping`**
  - **`@RequestBody`**
  - **`@GetMapping`**
  - **`@PostMapping`**
  - **`@DeleteMapping`**

Continueed (1) .. .. ..

- **Maven Starters**
  - spring-boot-starter-thymeleaf
  - spring-boot-starter-test
  - spring-boot-configuration-processor
  - spring-boot-starter-aop
  - spring-boot-starter-security
    - thymeleaf-extras-springsecurity5
  - spring-boot-starter-aop
  - spring-boot-starter-actuator
  - spring-boot-starter-logging *{ logging con LogBACK usando Simple Logging Facade for Java (SLF4J) }*
  - spring-boot-starter-tomcat *{ Servlet Server Tomcat }*
  - spring-boot-starter-log4j2 *{ logging con Log4j }*
  - spring-boot-starter-jetty *{ Servlet Server Jetty }*

Continueed (2) .. .. ..

- **Annotazioni java**
  - @Controller
  - @Configuration
  - @ConfigurationProperties
  - @Profile
  - @Value
  - @Aspect
    - ✓ @Before
    - ✓ @After
    - ✓ @AfterReturning
    - ✓ @Around

Profiles

I Profiles sono una funzionalità chiave che ci permette di associare la creazione di componenti a diversi profili/contesti – ad esempio dev, test, prod. … Possiamo quindi usare profili diversi per situazioni diverse.

Usando la annotazione java `@Profile` associamo un bean ad un particolare profile; l'annotazione si aspetta semplicemente il nome di uno o più profili.

Quando annotiamo un bean/component con un profile "dev" un'istanza del componente `DevDatasourceConfig` sarà creata/attiva solo se allo start-up si chiede di usare il profilo `dev`

```
@Component
@Profile("dev")
public class DevDatasourceConfig
```

Spring Boot permette di avere/definire profile-specific application.properties files con il formato

```
applications-{profile}.properties.
```

Spring Boot caricherà automaticamente le properties nel file application.properties comune a tutti i profiles e quelle definite nei "profile-specific .properties files" riferiti allo start-up come attivi.

I nomi profilo possono essere passati come JVM system parameters; il nome profile usato come segue sarà attivato allo start-up dell'applicativo:

```
java -jar -Dspring.profiles.active=dev ...
```

oppure

```
java -jar app-file-name.jar --spring.profiles.active=dev
```

Security

Wikipedia – Basic Access AUTH

https://it.wikipedia.org/wiki/Basic_access_authentication

Creare un file in formato

- jks (Java Keystore File) oppure
- p12 [PKCS12 (§)]

https://www.baeldung.com/spring-boot-https-self-signed-certificate
https://mkyong.com/spring-boot/spring-boot-ssl-https-examples/
https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html#Prepare_the_Certificate_Keystore

(§) PKCS12: Public Key Cryptographic Standards

https://en.wikipedia.org/wiki/PKCS_12

Actuators

"An actuator is a manufacturing term that refers to a mechanical device for moving or controlling something. Actuators can generate a ***large amount of motion from a small change.****"*

Spring Boot include delle features aggiuntive per aiutarci a monitorare e a gestire la nostra applicazione quando essa viene usata in produzione.

Possiamo scegliere di gestire/monitare la nostra applicazione attraverso endpoints HTTP oppure con le API Java JMX.

Esempi:

- `httptrace, health,`
- `metrics, mappings, env,`
- `beans, loggers, logfile,`
- `threaddump, heapdump.`

Starter: **`spring-boot-starter-actuator`**

Default URL per HTTP endpoints : **`/actuator`** **`/actuator/<actuator-name>`**

https://docs.spring.io/spring-boot/docs/2.2.6.RELEASE/reference/htmlsingle/#production-ready

Deploying Spring Boot Applications

- **Linux**

  https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#deployment-service

- **Windows service**

  https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#deployment-windows
  https://github.com/kohsuke/winsw
  https://github.com/snicoll-scratches/spring-boot-daemon

- **Deploying in Docker Container**

  https://medium.com/swlh/deploying-spring-boot-applications-15e14db25ff0

- **Deploying in Microsoft Azure**

  https://spring.io/guides/gs/spring-boot-for-azure/
  https://docs.microsoft.com/it-it/azure/developer/java/spring-framework/deploy-spring-boot-java-app-with-maven-plugin

Links

- **Spring «Getting Started» Docs**

  https://spring.io/quickstart

  https://spring.io/guide

  https://start.spring.io/

- **Spring Boot/Spring Docs**

  https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

  https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/index.html

- **Baeldung**

  https://www.baeldung.com/start-here

  https://www.baeldung.com/spring-boot

- **mkyong.com**

  https://mkyong.com/tutorials/spring-boot-tutorials/

THANK YOU ALL!

Antonio