

Gestione dei controller – Parte I

Cosa imparerai

- Come definire ed utilizzare i controller in una web app Spring MVC

Gestione dei controller

Spring MVC consente di definire i controller utilizzando due annotation:

- ❑ **@Controller (org.springframework.stereotype.Controller)**: è presente dalla versione 2.5 di Spring ed è utilizzata per indicare al container che una classe è un controller Spring MVC.

L'annotation *@Controller* è una specializzazione dell'annotation *@Component*.

```
@Controller
public class HelloController {
    @GetMapping("/")
    public String showForm() {
        return "index";
    }
}
```

```
@Controller
public class HelloController {
    @ResponseBody
    @RequestMapping("/")
    public String showForm() {
        return "hello!";
    }
}
```

Gestione dei controller

- ❑ **@RestController**: è presente dalla versione 4.0 di Spring ed è utilizzata per indicare al container che la classe è un *RESTful web services* controller.

@RestController è una specializzazione dell'annotation *@Controller*.

Questa annotation è la combinazione delle due annotation *@Controller* e *@ResponseBody*.

Prima di questa annotation, infatti, era necessario utilizzare l'annotation *@Controller* sulla classe e *@ResponseBody* su tutti i metodi della classe.

Utilizzando *@RestController* sulla classe, quindi, non abbiamo necessità di annotare i metodi con *@ResponseBody*.

```
@RestController
public class HelloRestController {
    @GetMapping("/rest")
    public String showForm() {
        return "CIAO!!!!";
    }
}
```

Gestione dei controller

Una classe annotata con `@Controller` o `@RestController` è un bean e segue la logica dei bean di Spring.

Le annotation `@Controller` e `@RestController` hanno le seguenti caratteristiche:

- ❑ consentono il rilevamento automatico (perché le annotation sono specializzazioni di `@Component`)
- ❑ indicano al container che la classe ha un ruolo di componente web (MVC o RESTFul web services).

Per abilitare il rilevamento automatico dei bean annotati con `@Controller` o `@RestController`, è necessario aggiungere la scansione dei componenti nella classe in cui definiamo la configurazione, utilizzando l'annotation `@ComponentScan`.

```
@EnableWebMvc
@Configuration
@ComponentScan(basePackages = { "it.test.web.controller" })
public class WebConfig {
```

Gestione dei controller

Mapping delle URL

Per mappare uno o più path di una URL ad un controller è possibile utilizzare l'annotation **@RequestMapping**.

L'annotation può essere utilizzata:

- ❑ **A livello di classe:** utilizzata se vogliamo mappare dei path condiviso tra più metodi
- ❑ **A livello di metodo:** utilizzata per mappare dei path su un metodo

```
@RestController
@RequestMapping(path = "/rest")
public class HelloRestController {
    @RequestMapping(path = "/1")
    public String showForm() {
        return "CIAO!!!!";
    }

    @RequestMapping(path = "/2")
    public String showForm2() {
        return "CIAO 2!!!!";
    }
}
```

← *http://mysite.com/rest/1*

← *http://mysite.com/rest/2*

Gestione dei controller

Mapping delle URL

L'annotation ha vari attributi che consentono di specificare:

- ❑ **path**: il path a cui sarà associato il metodo o la classe
- ❑ **method**: il metodo HTTP (GET, POST, PUT,...)
- ❑ **params**: i parametri di richiesta
- ❑ **headers**: le intestazioni della richiesta
- ❑ **consumes e produces**: i tipi di media

Ognuno di questi parametri accetta una stringa o un array di stringhe.

```
• consumes : String[] - RequestMapping
• headers : String[] - RequestMapping
• method : RequestMethod[] - RequestMapping
• name : String - RequestMapping
• params : String[] - RequestMapping
• path : String[] - RequestMapping
• produces : String[] - RequestMapping
• value : String[] - RequestMapping
```



Di cosa abbiamo parlato in questa lezione

- Come definire ed utilizzare i controller in una web app Spring MVC