



Spring MVC: Controllers and Request Mappings

Artur Boronat

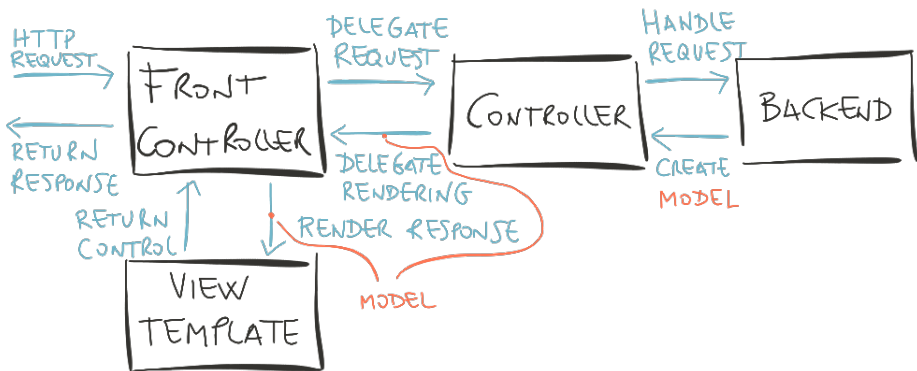


spring



INSIDE THE HTTP SERVER:

HTTP REQUEST / RESPONSE LIFECYCLE



Controller

Responsibility: HTTP request handling

- links a HTTP request to a method with an annotation `@RequestMapping`
- method parameters: get user input
- method body: population of the model
 - business logic (access to database, computations, etc.)
 - determines view
 - interprets exceptions arisen from business logic
- return value: view name

Handling HTTP requests (GET)

- attached to a class: defines relative url `http://localhost:8080/hello/`

```
@RequestMapping("/hello")  
public String hello(Model model) { .. }
```

- attached to a class: defines relative url `http://localhost:8080/index/hello/`

```
@RequestMapping("/index")  
public class IndexController {  
    @RequestMapping("/hello")  
    public String hello(Model model) { .. }  
}
```

Handling HTTP requests (GET): parameters

- using path variables with `@PathVariable`:

`http://localhost:8080/hello/World`

```
@RequestMapping("/hello/{value}")  
public String hello(@PathVariable String value, Model model)  
{ ... }
```

- request parameters: `http://localhost:8080/hello?value=World`

```
@RequestMapping("/hello")  
public String greetingParam(  
    @RequestParam(value="value", required=false,  
        defaultValue="World") String value,  
    Model model) { ... }
```

Handling HTTP requests (GET): type conversion

- getting primitive datatypes: <http://localhost:8090/hello?value=1>

```
@RequestMapping("/hello")
public String primitive(@RequestParam Integer value, Model
    model) { ... }
```

- getting dates: <http://localhost:8090/hello/2016-07-10>

```
@RequestMapping("/hello/{value}")
public String date(@PathVariable
    @DateTimeFormat(iso=ISO.DATE) Date value, Model model) {
    ... }
```

- getting collections: <http://localhost:8090/hello?values=1&values=2>

```
@RequestMapping("/hello")
public String collection(@RequestParam Collection<Integer>
    values, Model model) { ... }
```

Handling HTTP requests (GET)

- Model parameter: allows us to access the model

```
@RequestMapping("/hello")
public String hello(Model model) {
    model.addAttribute("name", "World");
    return "hello";
}
```

- **@ModelAttribute**

- fetches the object associated with the attribute `user` from the model

```
@RequestMapping("/hello")
public String hello(@ModelAttribute User user) {
    return "hello";
}
```

- if the entry is not present in the model, the object is instantiated and added to the model
 - the argument's fields are populated from all request parameters that have matching names

Common Mistakes

HTTP Error 404

- **This localhost page can't be found**
- Either the request URL is not correct or
- the view cannot be located (check view name and configuration).

The value does not appear in the view

- Check the attribute name used, both in the view and in the controller
- Check error messages in the output console