

L3. NoSQL

S2. Miniproject

Dr A Boronat

Table of Contents

- ① Motivation
- ② Spring MVC
- ③ Views
- ④ Exercises

Cloud Integration: Twitter

Use cases

- to build bots that interact with Twitter
- to read/log tweets
- to build intelligence from Twitter data

Components

- REST API
- Front-end: in our case a (Groovy) web app

Web application

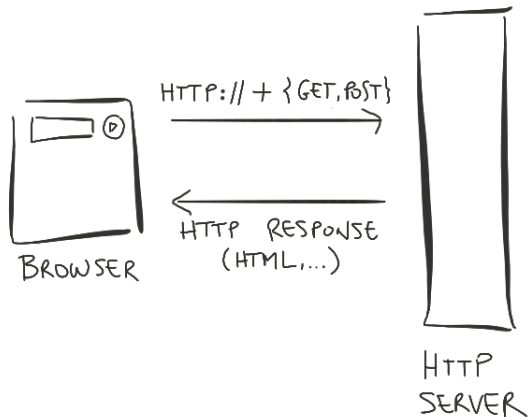
- **Web application**: client-server software application in which
 - the client (or user interface) runs in a web browser
 - the application server listens at some URL (**base URL**) and a port
 - when developing a web application this will be

http://localhost:8080

by default

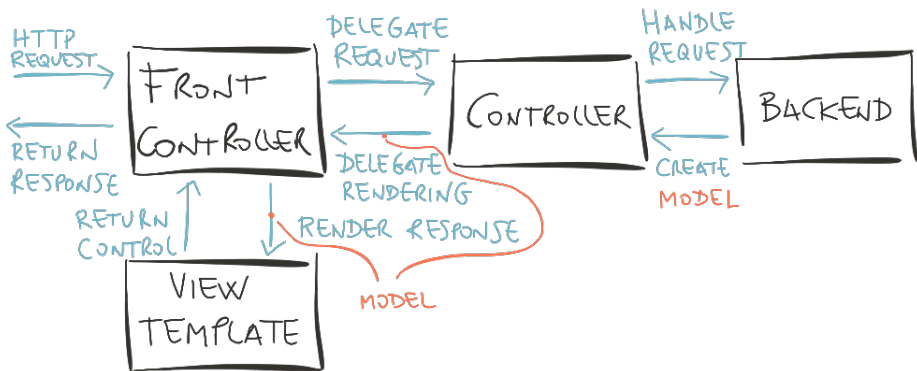
- web applications may contain
 - **static content**: HTML, images
 - **dynamically generated content**: HTML produced by JSPs after querying a database

DEALING WITH HTTP REQUESTS

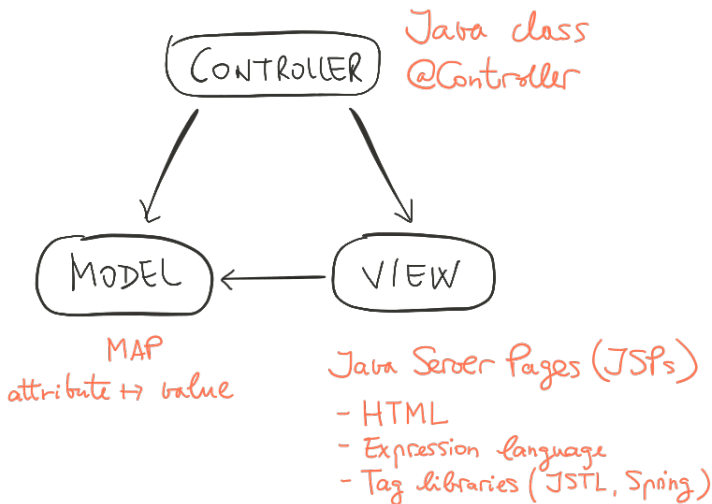


INSIDE THE HTTP SERVER:

HTTP REQUEST / RESPONSE LIFECYCLE



MODEL VIEW CONTROLLER



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)

- Model parameter: allows us to access the model

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

- **@ModelAttribute**

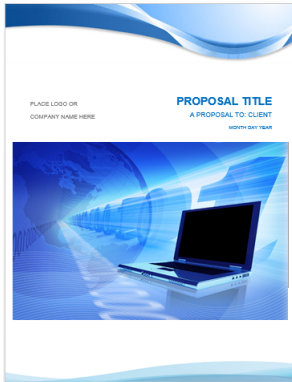
- fetches the object associated with the attribute `user` from the model

```
@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

Views: JSPs (JavaServer Pages)

- JSP files work as templates



- The controller chooses which template to apply by name (return value)
- The view resolver (configured in **WebConfig.java**) resolves the template:
 - instantiates the template: fills in gaps with information from model
 - generates code

Views: JSPs (Java Server Pages)

Generation of dynamic content (HTML)

- information from **model**, prepared by the **controller**
- tag libraries for controlling generation of HTML: loops, conditions
- tag libraries for forms: to post information

Ingredients

- **Expression language**: to fetch attribute values from model
- **JSTL** (JavaServer Pages Standard Tag Library): tags to define loops and conditions
- **Spring form tag library**: to design web forms that integrate well with Spring MVC

Views: Expression Language (EL)

EL

- language to evaluate expressions (returning a value)
- no loops, no conditions

How to use it

- `${expr}`: outputs the result of the expression in an HTML page
 - in view `example.jsp`: `<p>${product.getName()}</p>`
- we can refer to model attributes

```
// in the controller class
@RequestMapping(...)
public String productDetail(@ModelAttribute("product") Product product, ... ) {
    ...
    return "example"
}
```

- difference with GStrings in Groovy: the variables in expression `expr` are fetched from the **model** (as opposed to be local or global variables in the Groovy script)

Pulling data from Twitter

Twurl

- **curl** for the Twitter API
- to grant an access token to a client application for a specified user and then sign all requests with that access token
- support for multiple access tokens to easily switch between different client applications and Twitter accounts

Exercise 1

- to test functionality: comment `./gradlew migrateToMongoDb` in script `twitterMigrate.sh`
- create collections on MongoDB instance

Analysing data from Twitter

Exercise 2

- list those friends who also follow you

Exercise 3

- global favorite count
- count popular tweets (>5 favorites)

Essay: Impact of NoSQL Technologies on Twitter System

NoSQL Stores

- What NoSQL store has been used?
- What features are provided by that chosen NoSQL store?
- Why has it been chosen?

Scalability in Twitter

- how partitioning and replication is implemented in the architecture of Twitter

What's next?

- Next week is study time to do the miniproject
- For campus-based students: lab session on Tuesday 31 October
- For DL students: tutor meetings
- Deadline: 3 November, 9am
- Submission using your private repository on GitHub