

Spring MVC

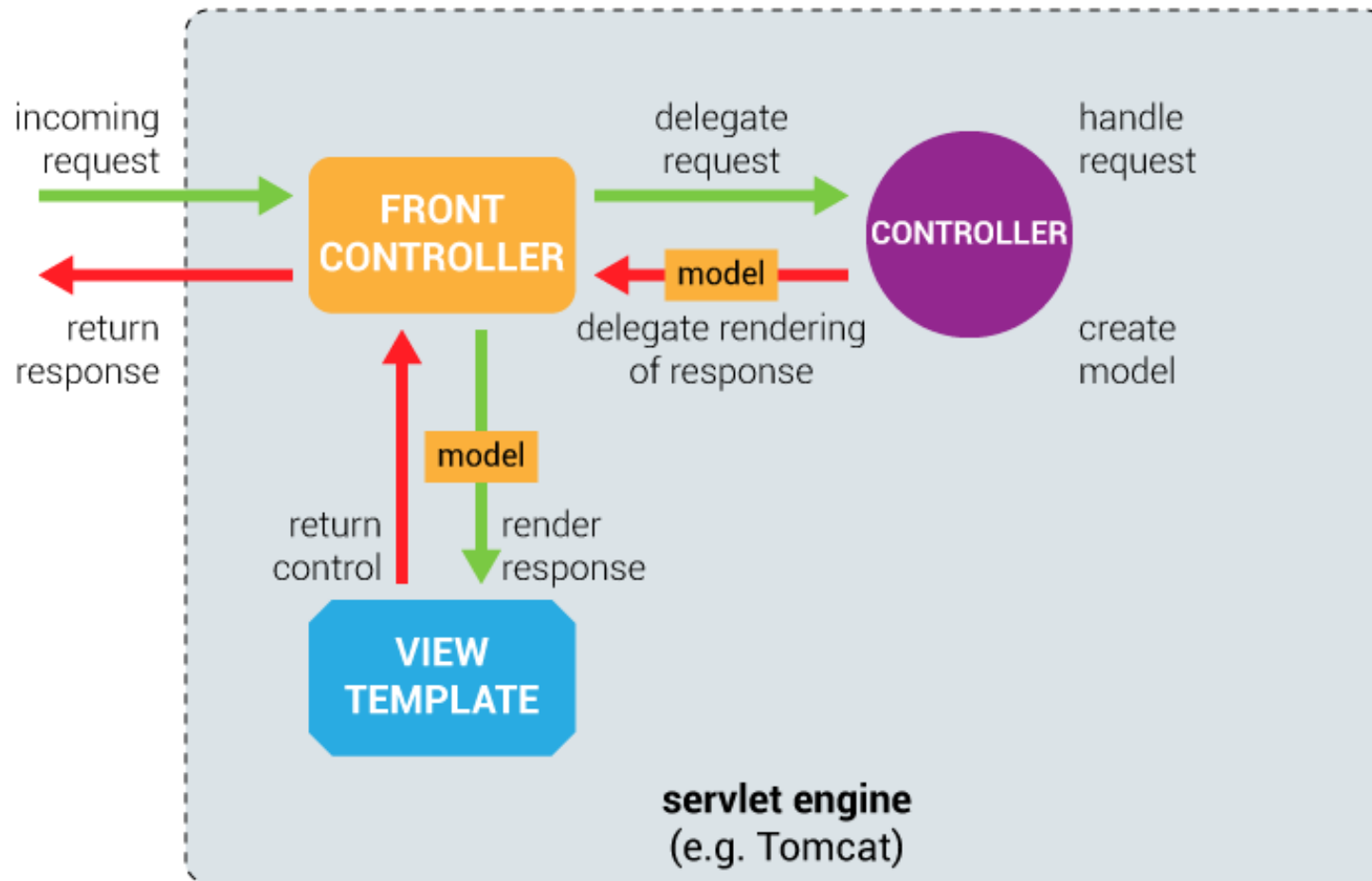
Guillaume Dufrêne – Lionel Seinturier

Université de Lille – Sciences et Technologies

Définition

application framework

- Outils et de règles
- Faciliter le développement d'applications Java EE en MVC

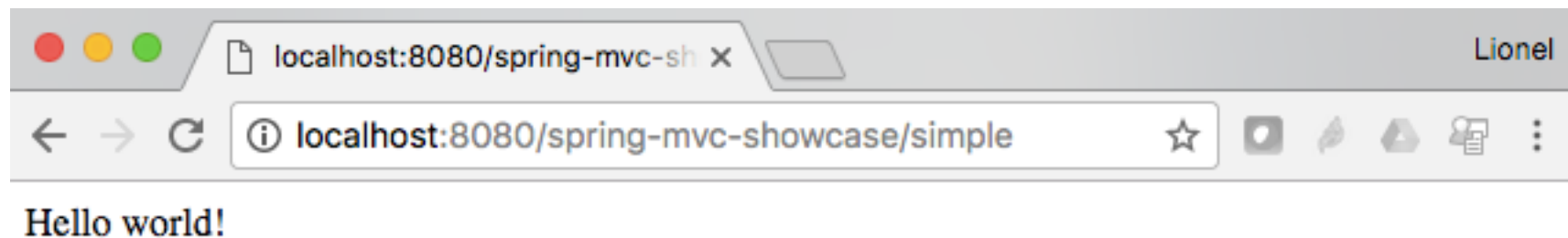


Définition

- point d'entrée qui définit les services
- reçoit des requêtes HTTP
- les transforme en commandes pour le modèle et les vues

Exemple de code

```
@Controller  
public class SimpleController {  
  
    @RequestMapping("/simple")  
    @ResponseBody  
    public String simple() {  
        return "Hello world!";  
    }  
}
```



Définition

- Association d'une URL et d'une méthode

```
@RequestMapping("/rdv")
@Controller
public class RdvController {

    @RequestMapping("/nouveau")
    public String nouveau() { ... }

    @RequestMapping("/liste")
    public String liste() { ... }
}
```

http://.../rdv/nouveau

http://.../rdv/liste

Paramètres

- URLs paramétrées {...} avec annotation @PathVariable

```
@RequestMapping("/rdv")
@Controller
public class RdvController {
```

```
    @RequestMapping("/details/{nom}")                http://.../rdv/details/Bob
    public String findByName(
        @PathVariable String nom ) { ... }
```

```
    @RequestMapping("/details/{nom}/{date}")        http://.../rdv/details/Bob/mai
    public String findByNameAndDate(
        @PathVariable String nom,
        @PathVariable String date ) { ... }
```

```
}
```

Paramètres

- Utilisation d'expressions régulières `{regexp}`
- Association d'un nom de variable `{varName: regexp}`

Exemples

```
@RequestMapping("/hotels/{hotel}/*")
```

```
@RequestMapping("/fichier/{symbolicName:[a-z]+}{extension:\\.[a-z]+}")
```


Requêtes HTTP

- différentiation des méthodes HTTP utilisées

```
@RequestMapping("/clients")
@Controller
public class ClientController {

    @RequestMapping("/{nom}", method = RequestMethod.GET)
    public String lire(...) { ... }

    @RequestMapping("/{nom}", method = RequestMethod.POST)
    public String ajouter(...) { ... }
}
```

Requêtes HTTP

- depuis Spring 4.3

```
@RequestMapping("/clients")
@Controller
public class ClientController {

    @GetMapping("/{nom}")
    public String lire(...) { ... }

    @PostMapping("/{nom}")
    public String ajouter(...) { ... }
}
```

Différents types de paramètres supportés

- session
- requête HTTP
- flux d'entrée et de sortie de la requête
- paramètres de la requête
- etc.

```
@Controller @RequestMapping("/ecommerce")
public class MagasinController {

    @GetMapping("/ajouter/{id}")
    public String ajouter( HttpSession session, @PathVariable String id ) {
        Panier monPannier = session.getAttribute("panier");
        if( monPannier == null ) {
            monPannier = new Pannier();
            session.setAttribute("panier", monPannier);
        }
        monPannier.ajouter(id);
    }
}
```

Différents types de paramètres supportés

- contenu de fichier HTML
- vue
- objet sérialisé en JSON
- redirection
- etc.

```
@Controller @RequestMapping("/ecommerce")
public class MagasinController {
```

```
    @GetMapping( path = "/details/{id}",
                produces = MediaType.APPLICATION_JSON_UTF8_VALUE )
    public Article details( @PathVariable String id ) {
        Article article = findById(id);
        return article;
    } }
```

```
public class Article {
    private String id;
    private String nom;
    private double prix;
    // constructeur
    // getter/setter
}
```

```
{
    "id": "42",
    "nom": "DVD",
    "prix": 9.99
}
```

Différents types de paramètres supportés

- redirection

```
@Controller @RequestMapping("/ecommerce")
public class MagasinController {

    @GetMapping( path = "/commander" )
    public String commander() {
        magasin.enregistrerCommande();
        return "redirect:/ecommerce/index.html";
    }
}
```

Spring MVC

- un *framework* pour les applications web MVC
- Contrôleur : traitement des requêtes
- S'appuie sur des données
- Utilise des vues