

Spring MVC controller

Avec Spring MVC, un contrôleur comporte l'annotation `@Controller`. Il reçoit les requêtes Http en réponses aux RequestMappings définis grâce aux annotations et fournit une réponse Http

```
@Controller
public class HelloController {

    @GetMapping("/hello")
    public @ResponseBody String index() {
        return "Hello world!";
    }
}
```

Request

Les annotations `@RequestMapping` et dérivées permettent de définir le routage :

```
@Controller
@RequestMapping("/persons")
class PersonController {

    @GetMapping("/{id}")
    public Person getPerson(@PathVariable Long id) {
        // ...
    }

    @PostMapping
    @ResponseStatus(HttpStatus.CREATED)
    public void add(@RequestBody Person person) {
        // ...
    }
}
```

Url définies :

- GET: /persons/1 ⇒ Affiche la personne d'id 1
- POST: /persons ⇒ Ajoute la personne passée en paramètre
- voir [Request mapping](#)

Injection de dépendance :

Certains objets, en fonction de leur type, ou grâce aux annotations, peuvent-être directement passés à une méthode de contrôleur.

Exemple : récupération de la requête `HttpServletRequest`

```
@GetMapping("/{id}")
public Person getPerson(@PathVariable Long id, HttpServletRequest request) {
    // ...
}
```

- voir [Methods arguments](#)

Response

Affichage direct

Sans passage par une vue, la chaîne retournée constitue le response body.

```
@RequestMapping
public @ResponseBody String index() {
    return "Hello World!";
}
```

Appel d'une vue

Par défaut, le String retourné correspond à un nom de vue (dans ce cas main/index.html à partir du dossier des templates défini dans application.properties):

```
@RequestMapping
public String index() {
    return "main/index";
}
```

Passage de variables à la vue

Il est possible d'injecter une instance de la classe **ModelMap** ou **Model** :

```
@GetMapping
public String messageAction(ModelMap model) {
    model.put("message", "Ceci est un message");
    return "main/index";
}
```

Il est également possible de charger une vue et de lui passer des variables depuis une instance de **ModelAndView** :

```
@RequestMapping
public ModelAndView index() {
```

```
ModelAndView mv=new ModelAndView("main/index");
mv.addObject("Title","Titre de la page");
return mv;
}
```

voir [Return values](#)

Redirections

Une action d'un contrôleur peut éventuellement rediriger vers une autre action, après avoir effectué ses propres traitements, avec éventuel passage de paramètres de type flash :

```
@RequestMapping
public RedirectView submitAction(@RequestParam String value, RedirectAttributes
attrs) {
    //Traitements
    attrs.addFlashAttribute("message","Formulaire envoyé...");
    return new RedirectView("/");
}
```

La vue associée à la route / pourra accéder à la variable **message** et l'afficher.

Ressources

- [RequestMapping annotation](#)

From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/framework-web/spring/controllers>

Last update: **2023/09/26 23:51**

