

# TD0 & TD1

Notions abordées :

- Contrôleurs
- Routage
- Vues
- Session et contexte

## Création

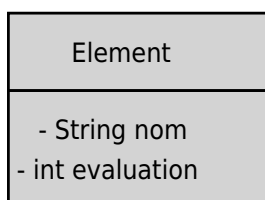
Créer le projet td0 :

- Group : s4.spring
- Artifact : td0
- Packaging : War
- Description : Gestion d'items
- Dependencies : Web, Devtools, Mustache

Configurer le projet dans **application.properties**, pour que le contextPath soit **td0/**, configurer Mustache

## Objet métier

Créer une classe Element dans s4.spring.models :



CREATED WITH YUML

Ajouter :

- Des accesseurs (getters/setters) sur les 2 membres
- Sur-définir la méthode **equals** pour que 2 items du même nom soit considérés comme égaux

## Initialisation d'un objet en session

L'annotation `@SessionAttributes` permet de définir des variables de session. Combinée à `@ModelAttribute`, elle permet l'initialisation correcte de la variable de session.

```
@SessionAttributes("items")
public class MainController {

    @ModelAttribute("items")
```

```
public List<String> getItems(){  
    return new ArrayList<>();  
}  
}
```

## Routes

Adresse	Description
items/	Affiche la liste des éléments stockée en Session + 1 bouton pour ajouter un élément (/items/new)
items/new	Afficher un formulaire d'ajout d'élément (seulement son nom), la validation va vers /items/addNew puis redirige vers /items
items/inc/{nom}	Incrémente l'évaluation de l'élément de 1, puis redirige vers /items
items/dec/{nom}	Décrémente l'évaluation de l'élément de 1, puis redirige vers /items

## Redirection

```
@GetMapping("items/addNew")  
public RedirectView addNew(@RequestParam Strin id) {  
    Optional<Joueur> opt=repo.findById(id);  
    if(opt.isPresent()) {  
        Joueur joueur=opt.get();  
        List<Pari> paris=joueur.getParis();  
        if(paris.size()==0)  
            repo.delete(joueur);  
    }  
    return new RedirectView("/joueur/");  
}
```

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:  
<http://slamwiki2.kobject.net/framework-web/spring/td0?rev=1549016095>

Last update: **2019/08/31 14:43**

