



# Spring MVC - Formulaire

# Lire les données d'un formulaire

Etape 1 : afficher un formulaire

page.jsp

<a ...>

button

1 get /getForm

Salutation  
Controleur

@RequestMapping("/getForm")

@RequestMapping("/postForm")

2

salutation-form.jsp

Quel est votre  
nom ?

Valider

3  
Post  
/postForm

4

salutation-message.jsp

Bonjour à l'univers Spring !

Votre nom est : Tartanpion

Etape 1 : afficher un formulaire

page.jsp

A diagram representing the content of page.jsp. It consists of a white rectangular box containing two blue rectangular buttons. The top button contains the text "<a ...>" and the bottom button contains the text "button".



**salutation-form.jsp**

A diagram representing the content of salutation-form.jsp. It consists of a white rectangular box containing the text "Quel est votre nom ?" and a blue rectangular button with the text "Valider".

Etape 2 : envoyer un formulaire, récupérer les données, les afficher dans une autre page

**salutation-form.jsp**

A diagram representing the content of salutation-form.jsp. It consists of a white rectangular box containing the text "Quel est votre nom ?" and a blue rectangular button with the text "Valider".



**salutation.jsp**

A diagram representing the content of salutation.jsp. It consists of a white rectangular box containing the text "Bonjour à l'univers Spring !" and "Votre nom est : Tartanpion".

# Pas à pas

- Créer un bouton déclencheur d'une requête du formulaire
- Créer la classe contrôleur
  - y créer la méthode `getSalutationForm()` "mappée" sur la requête du formulaire
  - créer le `salutationFrom.jsp`
  - y créer la méthode `postSalutationForm()` "mappée" sur la soumission du formulaire
  - créer la page `salutationMessage.jsp`

# ( 1 ) la requête du formulaire : ajouter un lien html au menu

Main-menu.jsp

```
<!DOCTYPE html>
<html>

  <body>
    <h2>Welcome to main menu!</h2>

    <hr>
    <a href="getSalutationForm">Hello Spring MVC form</a>
  </body>

</html>
```

<http://localhost:8080/getSalutationForm>

# Le contrôleur

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class SalutationControleur {

    //afficher le formulaire
    @RequestMapping("/getSalutationForm")
    public String afficheSalutationForm() {
        return "salutation-form";
    }

    @RequestMapping("/postSalutationForm")
    public String traiteSalutationForm() {
        return "salutation-message";
    }
}
```

# SalutationForm.jsp (dans WEB-INF/view/salutation-form.jsp)

```
<!DOCTYPE html>
<html>
<head>
<title>Salutation - Form</title>
</head>

<body>

<form action="postSalutationForm" method="GET">

<input type="text" name="userName"
placeholder="What's your name?" />

<input type="submit" />

</form>

</body>

</html>
```

# salutation-message.jsp

```
<!DOCTYPE html>
<html>

<body>

Hello World of Spring!

<br><br>

User name is : toto

<br><br>

The message:  ??

</body>

</html>
```

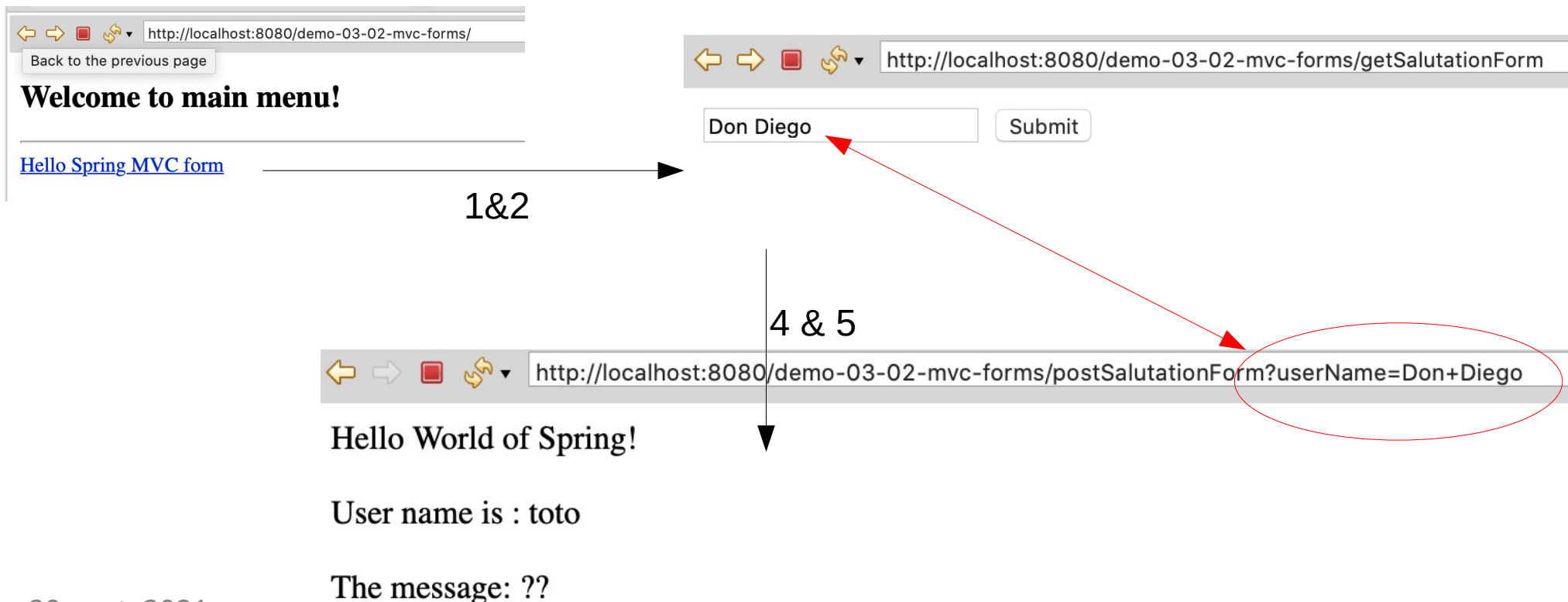
Il s'agit pour l'instant  
d'un  
message statique

J'exécute l'application pour tester cet " application flow". => succès !!



# On doit gérer les données du formulaire pour générer un message dynamique

- Les données saisies dans le formulaire sont déjà injectées dans la requête de soumission par spring
- jetez un œil dans la barre d'adresse de votre navigateur lors de l'affichage du message statique



# HttpServletRequest & Model

Dans la méthode `postSalutationForm()` du contrôleur, on va pouvoir accéder à la requête `http` qui contient les données de la page du formulaire et à un objet `model` pour transporter vos données entre contrôleur et pages.

```
@RequestMapping("/postSalutationForm2")
public String receptionneSalutationForm(HttpServletRequest request, Model model) {

    // read the request parameter from the HTML form
    String theName = request.getParameter("userName");
    // convert the data to all caps
    theName = theName.toUpperCase();
    // create the message
    String result = "Yo ! " + theName;
    // add message to the model
    model.addAttribute("message", result);

    return "salutation-message";
}
```

# Depuis la page vous avez accès au conteneur de données Model

```
<!DOCTYPE html>  
<html>
```

```
<body>
```

Hello World of Spring!

```
<br><br>
```

User name is : \${param.userName}

```
<br><br>
```

The message: \${message}

```
</body>
```

```
</html>
```

# Run as > RUN ON SERVER

← → ■ 💰 http://localhost:8080/demo-03-02-mvc-forms/

**Welcome to main menu!**

[Hello Spring MVC form](#)



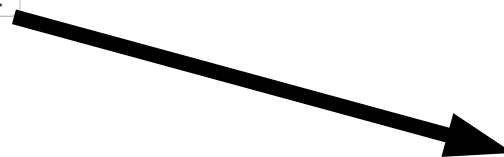
main-menu.jsp SalutationControleur.java salutation-form.jsp  
← → ■ 💰 http://localhost:8080/demo-03-02-mvc-forms/getSalutationForm

What's your name? Submit

← → ■ 💰 http://localhost:8080/demo-03-02-mvc-forms/getSalutationForm

Rodriguez ..Père et Fils !

Submit



← → ■ 💰 http://localhost:8080/demo-03-02-mvc-forms

**Hello World of Spring!**

User name is : Rodriguez ..P?re et Fils !

The message: Yo ! RODRIGUEEZ ..P?RE ET FILS !