

## Atelier2

### Création d'une Application Web avec MVC

#### Etape1 :

Créer un projet « Spring Starter Project » nommé le TP2.

#### Etape2 :

Gérer les dépendances dans le fichier pom.xml.

#### Etape3 :

Créer une class controlleur sous le package `com.example.tp2.controller`

#### HelloController.java

```
package com.example.demo.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

@Controller
public class HelloController {

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

    @PostMapping("/hello")
    public String sayHello(@RequestParam("name") String name, Model model) {
        model.addAttribute("name", name);
        return "hello";
    }
}
```

#### Etape 4 :

Créer les fichiers index.jsp et hello.jsp sous le dossier `src/main/webapp/WEB-INF/`

```
src
├── main
│   ├── java
│   ├── resources
│   └── Webapp
│       ├── WEB-INF
│       │   ├── hello.jsp
│       │   └── index.jsp
```

## Hello.jsp

```

1<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
2<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
3<!DOCTYPE html>
4<html>
5<head>
6<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7<title>Spring Boot</title>
8</head>
9<body>
10 <h1>Spring Boot -Application Web Avec MVC</h1>
11 <hr>
12
13 <h2>Your name is ${name}</h2>
14
15</body>
16</html>

```

## Index.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<!-- Static content -->
<link rel="stylesheet" href="/resources/css/style.css">
<script type="text/javascript" src="/resources/js/app.js"></script>

<title>Spring Boot</title>
</head>
<body>
  <h1>Spring Boot - MVC web application example</h1>
  <hr>







  <div class="form">
    <form action="hello" method="post" onsubmit="return validate()">
      <table>
        <tr>
          <td>Enter Your name</td>
          <td><input id="name" name="name"></td>
          <td><input type="submit" value="Submit"></td>
        </tr>
      </table>
    </form>
  </div>

</body>
</html>

```

## Etape 5:

Avec Spring Boot, le contenu web statique est fourni à partir du répertoire /static ou /public ou /resources ou /META-INF/resources.

- ✓  src/main/resources
  - ✓  static
    - ✓  css
      -  style.css
    - ✓  js
      -  app.js

**style.css**

```
.form {  
    background-color: #efefef;  
    width: 400px;  
    height: 50px;  
    border-radius: 7px;  
    padding: 20px;  
}
```

**App.jsp**

```
function validate() {  
    var name = document.getElementById("name").value;  
    if (name == '') {  
        alert('Please enter a valid name.');        return false;  
    } else {  
        return true;  
    }  
}
```

## Etape 6 : propriété de l'application

Spring Boot charge les propriétés de l'application à partir du fichier application properties et les ajoute à l'environnement Spring.

Modifier le fichier application properties sous le répertoire .

**application.properties**

```
spring.mvc.view.prefix = /WEB-INF/views/  
spring.mvc.view.suffix = .jsp  
spring.mvc.static-path-pattern=/resources/**
```