

# Technologie JEE

## Rapport TP N°5 : SPRING

































Réalisé par :

Abdelghani MOUSSAID

Encadré par :

Pr. Noredine GHERABI

## Architecture finale du projet :

- ▼  GestEtudiant
  - >  ApplicationSpringHibernate
  - >  JAX-WS Web Services
  - ▼  Java Resources
    - ▼  src
      - ▼  controller
        - >  EtudiantController.java
      - ▼  dao
        - >  Etudiant.java
        - >  EtudiantDao.java
        - >  EtudiantImpl.java
        - >  package-info.java
      - ▼  service
        - >  EtudiantService.java
        - >  ImplEtudiantService.java
        - >  package-info.java
        -  log4j.properties
    - >  Libraries
    - >  JavaScript Resources
    - >  Referenced Libraries
    - ▼  WebContent
      - ▼  jsp
        -  afficher.jsp
        -  index.jsp
        -  saisir.jsp
        -  test.jsp
      - >  META-INF
      - ▼  WEB-INF
        - >  lib
          -  application-servlet-config.xml
          -  spring-beans.xml
          -  web.xml

## Couche DAO :

Contient l'entité Etudiant avec ses attributs et ses Getters/Setters.

Une interface EtudiantDAO qui est implémentée par la classe EtudiantImpl comme suit :

```
import java.util.ArrayList;

/**
 * @author Abdelghani
 */
public class EtudiantImpl implements EtudiantDao {

    private List<Etudiant> etudiants = new ArrayList<>();

    @Override
    public void addEtudiant(Etudiant E) {
        Long code = 1L;
        if (!etudiants.isEmpty())
            code = etudiants.get(etudiants.size() - 1).getCode() + 1;
        E.setCode(code);
        etudiants.add(E);
    }

    @Override
    public void deleteEtudiant(long code) {
        Optional<Etudiant> e = etudiants.stream().filter(et -> et.getCode() == code).findFirst();
        if (e.isPresent())
            etudiants.remove(e.get());
    }

    @Override
    public void updateEtudiant(Etudiant E) {
        etudiants.set(etudiants.indexOf(E), E);
    }

    @Override
    public Etudiant getEtudiantByCode(long code) {
        return etudiants.stream().filter(et -> et.getCode() == code).findFirst().get();
    }

    @Override
    public List<Etudiant> getAllEtudiants() {
        return etudiants;
    }

    public void initialiser() {
        System.out.println("Initialisation ...");
    }
}
```

## Couche Servie:

Contient une nouvelle interface EtudiantService:

Son implémentation par la classe ImplEtudiantService :

```
import java.util.List;

/**
 * @author Abdelghani
 */
public class ImplEtudiantService implements EtudiantService {
    private EtudiantDao dao;

    public EtudiantDao getDao() {
        return dao;
    }

    @Override
    public void addEtudiant(Etudiant E) {
        // TODO Auto-generated method stub
        dao.addEtudiant(E);
    }

    @Override
    public void deleteEtudiant(long code) {
        // TODO Auto-generated method stub
        dao.deleteEtudiant(code);
    }

    @Override
    public void updateEtudiant(Etudiant E) {
        // TODO Auto-generated method stub
        dao.updateEtudiant(E);
    }

    @Override
    public Etudiant getEtudiantByCode(long code) {
        // TODO Auto-generated method stub
        return dao.getEtudiantByCode(code);
    }

    @Override
    public List<Etudiant> getAllEtudiants() {
        // TODO Auto-generated method stub
        return dao.getAllEtudiants();
    }

    public void setDao(EtudiantDao dao) {
        this.dao = dao;
    }
}
```

## Couche Controlleur :

### EtudiantControlleur.java

```
/**
 * @author Abdelghani
 *
 */
@Controller
public class EtudiantController {

    @Autowired
    private EtudiantService service;

    @RequestMapping(value = "/index")
    private String index() {
        return "index";
    }

    @RequestMapping(value = "/saisir")
    private String saisir() {
        return "saisir";
    }

    @RequestMapping(value="/saisie")
    public String saveEtud(Model model, Etudiant e){
        service.addEtudiant(e);
        model.addAttribute("EtudiantList", service.getAllEtudiants());
        return "saisir";
    }

    @RequestMapping(value="/affiche")
    public String affiche(Model model){
        model.addAttribute("EtudiantList", service.getAllEtudiants());
        return "afficher";
    }

    @RequestMapping(value="/delete")
    public String delEtud(Model model, @RequestParam(value="code") int code){
        service.deleteEtudiant(code);
        model.addAttribute("EtudiantList", service.getAllEtudiants());
        return "afficher";
    }

    @RequestMapping(value="/Rechercher")
    public String RechEtud(Model model,
        @RequestParam(value="code") int code){

        List<Etudiant> etudiants = new ArrayList<>();
        etudiants.add(service.getEtudiantByCode(code));
        model.addAttribute("EtudiantList", etudiants);

        return "afficher";
    }
}
```

## Couche Présentation :

Index qui redirige vers la page de saisie ou à la page d'affichage :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@taglib prefix="sp" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta>
<title>Bienvenu</title>
</head>
<body>
    <a href="saisir">Ajouter un nouveau étudiant</a><br>
    <a href="affiche">Afficher la liste des étudiants</a>
</body>
</html>
```

Saisir qui permet d'ajouter les étudiants :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@taglib prefix="sp" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta>
<title>Saisie</title>
</head>
<body>
    <form action="saisie" method="post">
        <table>
            <tr>
                <td>Nom:</td>
                <td><input type="text" name="nom"></td>
            </tr>
            <tr>
                <td>Prénom:</td>
                <td><input type="text" name="prenom"></td>
            </tr>
            <tr>
                <td>Ville:</td>
                <td><input type="text" name="ville"></td>
            </tr>
            <tr>
                <td>Age:</td>
                <td><input type="text" name="age"></td>
            </tr>
        </table>
        <input type="submit" value="Enregistrer"><br><br>
        <a href="index">Revenir à la page d'accueil</a><br>
        <a href="affiche">Afficher la liste des étudiants</a>
    </form>
</body>
</html>
```

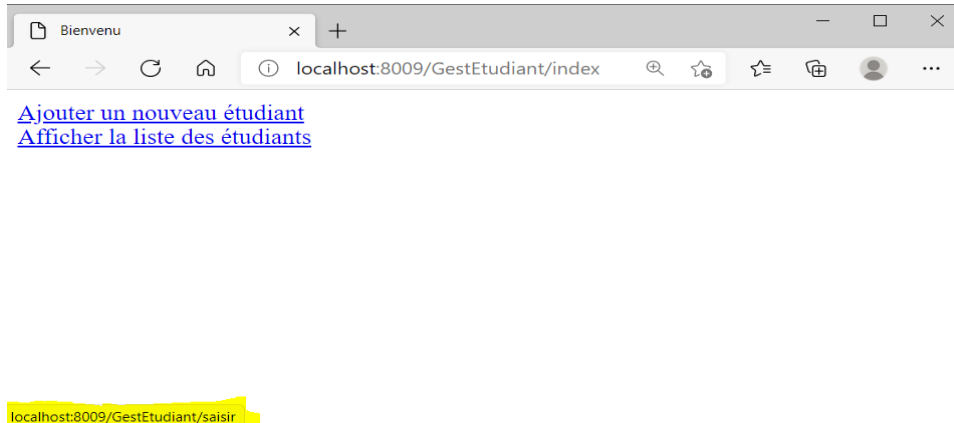
## Afficher et Rechercher les étudiants :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

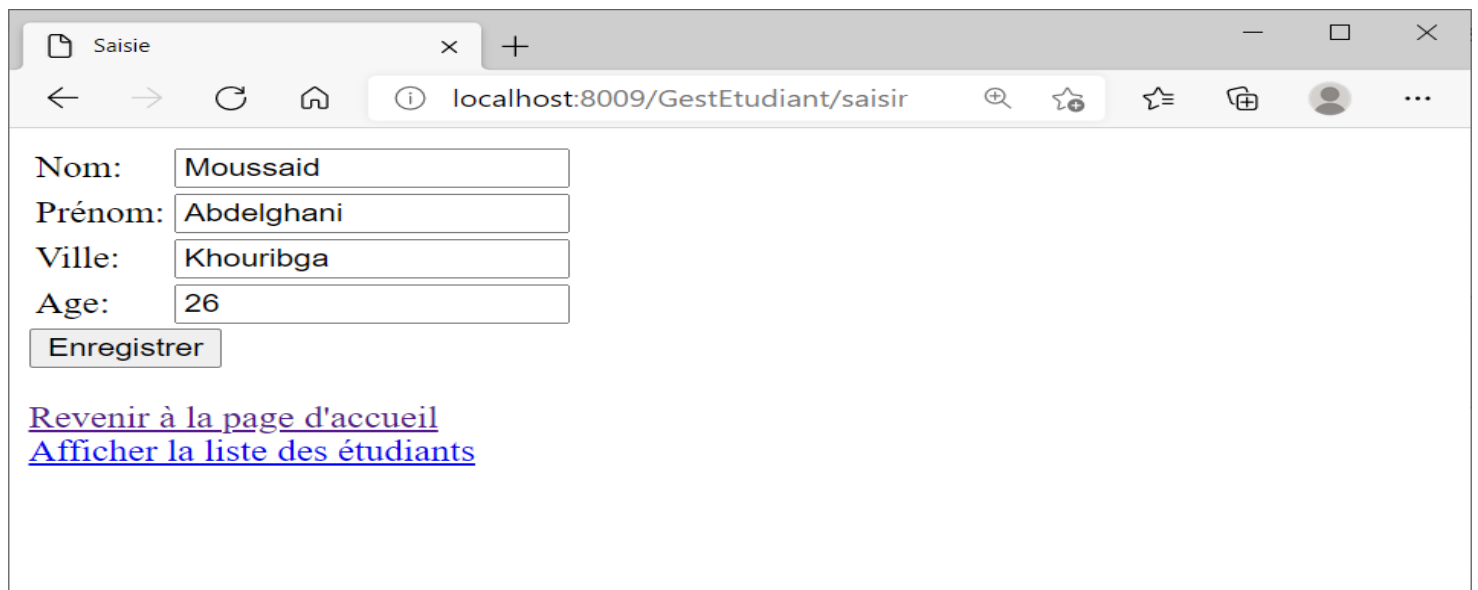
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="sp" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
<title>Lister et Rechercher</title>
</head>
<body>
<div>
<table border=1>
<caption>La liste des étudiants</caption>
<tr><th>code : </th> <th>Prenom: </th> <th>Nom: </th> <th>Age : </th> <th>Ville : </th> </tr>
<sp:forEach items="${EtudiantList}" var="listItem">
<tr><td>${listItem.code}</td><td>${listItem.nom}</td><td>${listItem.prenom}</td><td>${listItem.age}</td>
    <td>${listItem.ville}</td><td><a href="delete?code=${listItem.code}">supprimer</a></td>
</sp:forEach>
</table><br>
<form action="Rechercher">
    <label>code : </label>
    <input type="text" name="code">
    <input type="submit" value="Rechercher">
</form>
    <a href="index">Revenir à la page d'accueil</a><br>
    <a href="saisir">Ajouter un nouvel étudiant</a>
</div>
</body>
</html>
```

Resultats :

Page d'accueil:



La page de saisir :



The screenshot shows a web browser window with the title 'Saisie'. The address bar displays 'localhost:8009/GestEtudiant/saisir'. The page contains a form with the following fields and values:

Nom:	Moussaid
Prénom:	Abdelghani
Ville:	Khouribga
Age:	26

Below the form is a button labeled 'Enregistrer'. At the bottom of the page, there are two blue underlined links: 'Revenir à la page d'accueil' and 'Afficher la liste des étudiants'.



La page Afficher :

Lister et Rechercher
 
 localhost:8009/GestEtudiant/affiche

La liste des étudiants

code :	Prenom:	Nom:	Age :	Ville :	
1	Moussaid	Abdelghani	26	Khouribga	<a href="#">supprimer</a>
2	El Kerti	Mosaab	24	Settat	<a href="#">supprimer</a>
3	Katir	Yousra	23	Tanger	<a href="#">supprimer</a>

code :

[Revenir à la page d'accueil](#)  
[Ajouter un nouvel étudiant](#)

Recherche :

Lister et Rechercher
 
 localhost:8009/GestEtudiant/Rechercher?code=2

La liste des étudiants

code :	Prenom:	Nom:	Age :	Ville :	
2	El Kerti	Mosaab	24	Settat	<a href="#">supprimer</a>

code :

[Revenir à la page d'accueil](#)  
[Ajouter un nouvel étudiant](#)

Supprimer :

Lister et Rechercher

localhost:8009/GestEtudiant/delete?code=2

La liste des étudiants

code :	Prenom:	Nom:	Age :	Ville :	
1	Moussaid	Abdelghani	26	Khouribga	<a href="#">supprimer</a>

code :

[Revenir à la page d'accueil](#)

[Ajouter un nouvel étudiant](#)