



# TP1: Utilisez React et Spring Boot pour créer une application CRUD simple

Du 06 au 10 septembre 2021



- \* L'objectif de ce TP est de développer une simple application en fullstack qui permet de gérer le personnel.
- \* Une partie back end/api en Spring boot permettra de gérer avec une base de données in Memory H2. Spring data JPA et Hibernate seront utilisés pour la partie accès aux données
- \* Une partie front end permettra de gérer via le Framework React JS. NodeJs, TypeScript, Material... seront utilisés avec React.

## Partie1:Créer un projet Spring Boot dans Eclipse STS IDE

Dans cette partie 1, nous allons créer un projet de démarrage Spring dans Eclipse STS IDE et développer une API CRUD RESTFul à l'aide de [Spring boot](#) .

# \* Les points essentiels du back end

- ❖ Configuration de la base de données MySQL
- ❖ Créer une entité JPA - Employee.java
- ❖ Créez un référentiel de données Spring - EmployeeRepository.java
- ❖ Créer un contrôleur Spring Rest - EmployeeController.java

# Exécution de l'application

vous pouvez démarrer l'application  
Spring Boot

Ceci termine le développement des  
API Spring boot CRUD Rest.

# React App Frontend - Partie 2

- \* Au regard de ce qui précède, nous avons créé le projet de démarrage Spring backend, configuré la base de données MySQL et écrit les API CRUD Rest pour l'application.
- \* Dans cette partie 2, nous allons créer une application React et utiliser les API CRUD Restful développées.

# \* Create a React UI with Create React App

- \* Pour créer une nouvelle application, vous pouvez choisir l'une des méthodes suivantes :
  - Utiliser npx: `npx create-react-app react-frontend`
  - Utiliser npm: `npm init react-app react-frontend`
  - Utiliser du fil: `yarn create react-app react-frontend`

# \* Ajout de Bootstrap dans React à l'aide de NPM

\* `$ npm install bootstrap -save`

\* Après avoir installé le package d'amorçage, vous devrez l'importer dans votre fichier d'entrée d'application React.

\* Ouvrez le fichier `src/index.js` et ajoutez le code suivant :

```
import 'bootstrap/dist/css/bootstrap.min.css';
```



## \* EmployeeService - Consommer l'appel d'API CRUD REST

\* Pour nos appels API, nous utiliserons Axios . Vous trouverez ci-dessous la commande npm pour installer Axios

**npm add axios**

Créons un dossier de *services* dans le dossier **src** et créons un fichier Javascript nommé **EmployeeService.js** . Dans ce fichier, créez la classe **EmployeeService** avec les méthodes suivantes pour effectuer notre appel HTTP REST via **Axios**.

# \* Création du dossier composants

- \* Dans cette section, nous allons créer un nouveau dossier appelé *components* à l'intérieur du dossier *src* .
- \* Créez ensuite un nouveau fichier appelé *ListUserComponent.jsx* . Dans ce fichier, créez un composant de classe React nommé *ListUserComponent* .
- \* Créons un nouveau fichier nommé *HeaderComponent.js* et dans ce fichier, créons un composant nommé *HeaderComponent* .
- \* Créons un nouveau fichier nommé *FooterComponent.js* et dans ce fichier, créons un composant nommé *FooterComponent* .

# \* Configurer le routage

- \* Pour utiliser React Router, vous devez d'abord l'installer à l'aide de NPM :

`npm install react-router-dom`

Importer *BrowserRouter* , *Route* et *Switch* depuis le package **react-router-dom** :

```
import React, { Component } from 'react';
```

```
import { BrowserRouter, Route, Switch } from 'react-router-dom';
```

Ouvrons le composant **App** et configurons le routage.

# \* Ajouter et mettre à jour le composant employee

- \* Dans cette section, nous allons implémenter Ajouter un employé et mettre à jour la fonctionnalité d'employé. Nous utiliserons le même composant React pour effectuer à la fois l'opération d'ajout et de mise à jour des employés.
- \* Créons un nouveau fichier appelé *CreateEmployeeComponent.jsx* . Dans ce fichier, créez un composant de classe React nommé *CreateEmployeeComponent*.

# Afficher le composant employé

- Créons un nouveau fichier appelé *View **EmployeeComponent.jsx*** .
- Dans ce fichier, créez un composant de classeReact nommé ***ViewEmployeeComponent***

# \* Exécutez l'application React

- Avant d'exécuter React App, assurez-vous que votre projet de démarrage Spring est opérationnel.
- Utilisez la commande ci-dessous pour démarrer le projet : **npm start** ou **yarn start**

# Les résultats

WhatsApp React App Extensions +

localhost:3000/employees Réduire S

Gestion du personnel

## Liste du personnel

Add Employee

Employee First Name	Employee Last Name	Employee Email Id	Actions
Eric	Abidal	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Fatma	Fatma	Abid@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Tuomo	Thomas	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Zidane	ZZZ	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Eric	Abidal	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Fatma	Fatma	Abid@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Tuomo	Thomas	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Zidane	ZZZ	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Eric	Abidal	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Fatma	Fatma	Abid@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>
Tuomo	Thomas	Ad@yahoo.fr	<button>Update</button> <button>Delete</button> <button>View</button>

Taper ici pour rechercher

25°C 04:55 05/09/2021

# Les résultats

Gestion du personnel

### Add Employee

First Name:

Last Name:

Email Id:

All Rights Reserved 2021 @Tuemo