




Objectifs de la séance 8

(22-24/02/2022)

Les objectifs de la séance d'aujourd'hui:

Objectif 8.1 :

Consommer les services REST par une application front:

-  Utiliser les starters spring boot : thymeleaf, web
-  Utiliser RestTemplate pour consommer les services web REST.
-  Utiliser Bootstrap pour les styles CSS

Dans ce TP on suppose que :

- ☒ ~~Vous avez réalisé totalement le TP4.~~
- ☒ ~~Vous avez réalisé totalement le TP5.~~
- ☒ ~~Vous avez réalisé totalement le TP6.~~
- ☒ ~~Vous avez réalisé totalement le TP7.~~

SI CE N'EST PAS LE CAS : FAIRE D'ABORD LE TP4, TP5, TP6 ET LE TP7 D'URGENCE

I. UTILISATION DE SPRING BOOT FRAMEWORK

1. ALLER AU SITE SPRING INITIALIZR <https://start.spring.io/>

REPLIR LES PARAMÈTRES DU PROJET COMME MENTIONNÉ DANS L'ÉCRAN CI-DESSOUS.

Project		Language		
<input checked="" type="radio"/> Maven Project	<input type="radio"/> Gradle Project	<input checked="" type="radio"/> Java	<input type="radio"/> Kotlin	<input type="radio"/> Groovy
Spring Boot				
<input type="radio"/> 3.0.0 (SNAPSHOT)	<input type="radio"/> 3.0.0 (M1)	<input type="radio"/> 2.7.0 (SNAPSHOT)	<input type="radio"/> 2.7.0 (M1)	
<input type="radio"/> 2.6.4 (SNAPSHOT)	<input checked="" type="radio"/> 2.6.3	<input type="radio"/> 2.5.10 (SNAPSHOT)	<input type="radio"/> 2.5.9	

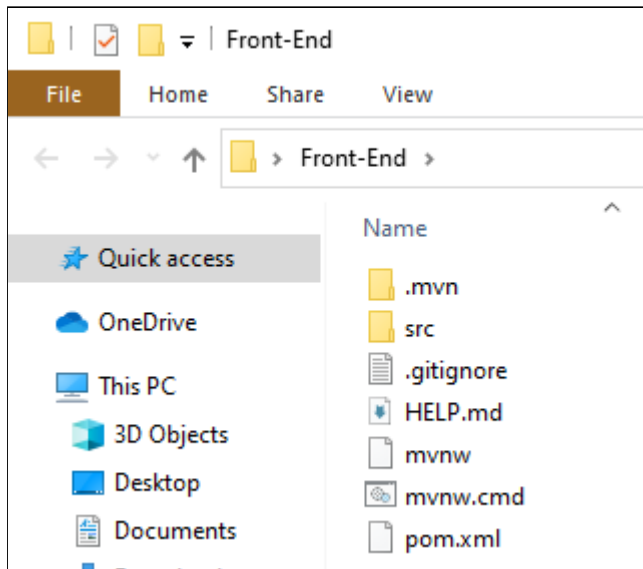
REPLIR LES META-DATA DU PROJET COMME MENTIONNÉ DANS L'ÉCRAN CI-DESSOUS.

Project Metadata	
Group	ma.cigma
Artifact	Front-End
Name	Front-End
Description	Application Front pour consommer les services REST du TP7
Package name	ma.cigma
Packaging	<input checked="" type="radio"/> Jar <input type="radio"/> War
Java	<input type="radio"/> 17 <input checked="" type="radio"/> 11 <input type="radio"/> 8

CHOISIR LES DEPENDENCIES DU PROJET CI-APRÈS : WEB , THYMELEAF ET LOMBOK .

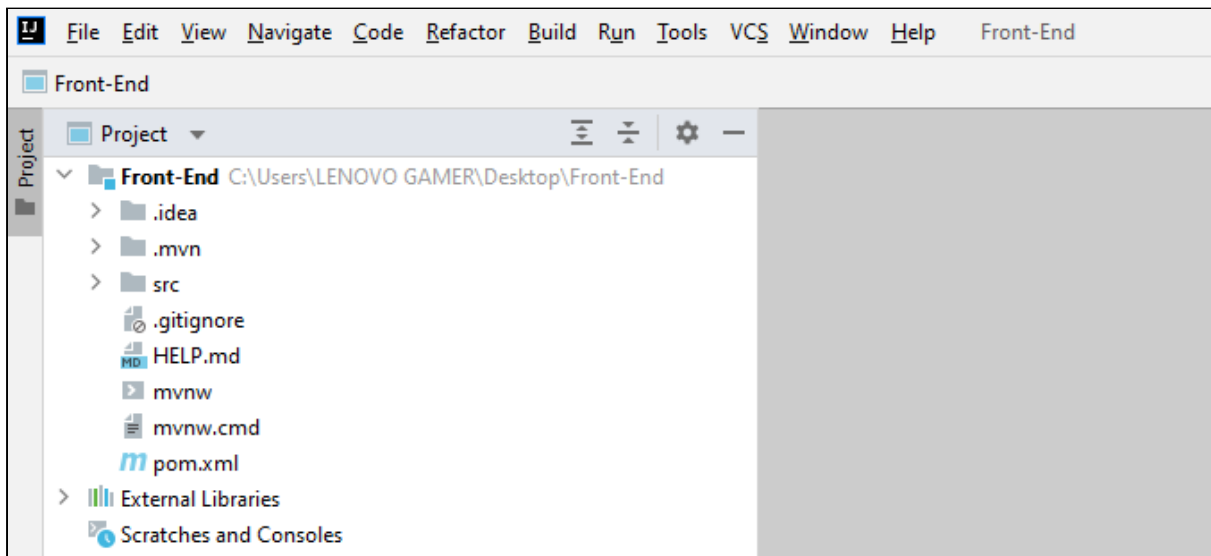
Dependencies	ADD DEPENDENCIES... CTRL + B
Spring Web WEB Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.	
Thymeleaf TEMPLATE ENGINES A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.	
Lombok DEVELOPER TOOLS Java annotation library which helps to reduce boilerplate code.	

PUIS CLIQUER SUR **GENERATE**. SAUVEGARDER LE FICHIER GENERÉ **FRONT-END.ZIP** SUR VOTRE DISQUE PUIS DÉCOMPRESSER LE.



2. OUVRIR CE PROJET DANS INTELLIJ:

FILE->OPEN->CHOISIR FRONT-END PROJECT->TRUST->NEW WINDOW



3. DANS LE POM.XML SUPPRIMER LE STARTER TOMCAT DU SPRING BOOT POUR UTILISER LE TOMCAT EMBEDDED.

4. DANS LE PROJET FRONT-END CRÉER LE PACKAGE : **ma.cigma.models**. PUIS CRÉER LA CLASSE **CLIENT** SUIVANTE:

```
package ma.cigma.models;

import lombok.Getter;
import lombok.Setter;
import lombok.ToString;

@Getter
```

```
@Setter
@ToString
public class Client {
    private long id;
    private String name;

    public Client() {
    }

    public Client(long id, String name) {
        this.id = id;
        this.name = name;
    }
}
```

5. MODIFIER LE FICHIER DE CONFIGURATION APPLICATION.PROPERTIES À APPLICATION.YML ET AJOUTER LA CONFIGURATION SUIVANTE :

```
server:
  port: 8085
spring:
  thymeleaf:
    cache: false
    enabled: true
    prefix: classpath:/templates/
    suffix: .html
api:
  url: http://localhost:8083/api
```

6. DANS LE PROJET FRONT-END CRÉER LE PACKAGE : **ma.cigma.controllers**. PUIS CRÉER LA CLASSE **CLIENTCONTROLLER** SUIVANTE :

```
package ma.cigma.controllers;

import ma.cigma.models.Client;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.client.RestTemplate;

import java.util.List;

@Controller
public class ClientController {
```

```
@Value("${api.url}")
private String apiUrl;

@Autowired
private RestTemplate restTemplate;

@GetMapping(value = {"/", "/clients"})
public String home(Model model) {
    model.addAttribute("client", new Client());
    List<Client> clients = restTemplate.getForObject(
        apiUrl + "/clients/all",
        List.class);
    model.addAttribute("clients", clients);

    return "index-client";
}
```

7. CRÉER LE BEAN RESTTEMPLATE DANS LA CLASSE DE DÉMARRAGE :

```
package ma.cigma;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
public class FrontEndApplication
{
    public static void main(String[] args) {
        SpringApplication.run(FrontEndApplication.class, args);
    }

    @Bean
    public RestTemplate getRestTemplate() {
        return new RestTemplate();
    }
}
```

8. CRÉER LE TEMPLATE `index-client.html` DANS RESOURCES/TEMPLATES.

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8"/>
    <link th:rel="stylesheet"
```

```

th:href="@{/webjars/bootstrap/4.0.0-2/css/bootstrap.min.css} " />
<link rel="stylesheet" type="text/css" th:href="@{/css/style.css}" />
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"

integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ
8lWUE00s/" crossorigin="anonymous">
</head>
<body>

<!-- Navigation -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark static-top">
  <div class="container">
    <a class="navbar-brand" href="/">Front Application</a>
    <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarResponsive"
aria-controls="navbarResponsive"
aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
          <a class="nav-link" href="#">Home
            <span class="sr-only">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Services</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Contact</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <table>
        <thead>
          <tr>
            <th scope="col">Photo</th>
            <th scope="col">Id</th>
            <th scope="col">Name</th>
            <th scope="col">Actions</th>
          </tr>
        </thead>
        <tbody>
          <tr th:each="client : ${clients}">
            <td>

```

```

        <div class="event-img">
            
        </div>
    </td>
    <td th:text="${client.id}"></td>
    <td th:text="${client.name}"></td>
    <td>
        <a class="btn btn-primary"
th:href="@{||/show-client/${client.id}||}">
            <i class="fa fa-edit fa-lg"></i> Edit</a>
        <a class="btn btn-danger"
th:href="@{||/delete-client/${client.id}||}">
            <i class="fa fa-trash fa-lg"></i> Delete</a>
    </td>
</tr>
</tbody>
</table>
</div>
</div>
</div>

<script th:src="@{/webjars/jquery/3.0.0/jquery.min.js}"></script>
<script
th:src="@{/webjars/popper.js/1.12.9-1/umd/popper.min.js}"></script>
<script
th:src="@{/webjars/bootstrap/4.0.0-2/js/bootstrap.min.js}"></script>

</body>

```

9. POUR QUE LE STYLE DE CE TEMPLATE S’AFFICHE CORRECTEMENT IL FAUT AJOUTER LA DÉPENDANCE DE BOOTSTRAP SUIVANTE DANS LE POM.XML

```

<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>bootstrap</artifactId>
    <version>4.0.0-2</version>
</dependency>

```

AJOUTER LE FICHIER style.css DANS resources/static/css

```

img {
    width: 50px;
    height: 50px;
}

```

10. DÉMARRER L’API (LE BACK-END) CRÉER DANS LE TP7 POUR LANCER À L’ÉCOUTE SUR LE PORT 8083 ET S’ASSURER QUE VOUS DES CLIENTS DANS LA BASE.

11. DÉMARRER L’APPLICATION FRONT-END ET ACCÉDER À L’URL SUIVANTE : <http://localhost:8085/>

12. LE RÉSULTAT D’ACCÈS À L’APPLICATION FRONT-END DEVRAIT RESSEMBLER À L’ÉCRAN :
Dossier des travaux pratiques. Module 1 : JEE and Fwks .Années scolaire 2021/2022. Niveau : Master FST Settat
Professeur : M. Boulchahoub Hassan hboulchahoub@gmail.com

Mise à jour 16 Feb 2022

Front Application				Home	About	Services	Contact
Photo	Id	Name	Actions				
	66	AHMED	Edit Delete				
	67	AHMED	Edit Delete				
	68	AHMED	Edit Delete				
	69	AHMED	Edit Delete				
	70	AHMED	Edit Delete				

13. AJOUTER UN FORMULAIRE HTML DANS LE TEMPLATE "TEMPLATES/INDEX-CLIENT.HTML"

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8"/>
  <link th:rel="stylesheet"
th:href="@{/webjars/bootstrap/4.0.0-2/css/bootstrap.min.css}" />
  <link rel="stylesheet" type="text/css" th:href="@{/css/style.css}" />
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtigrqD80Kn4Z8NTSRyMA2Fd33n5dQ
8lWUE00s/" crossorigin="anonymous">
</head>
<body>

<!-- Navigation -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark static-top">
  <div class="container">
    <a class="navbar-brand" href="/">Front Application</a>
    <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarResponsive"
aria-controls="navbarResponsive"
aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
          <a class="nav-link" href="#">Home
            <span class="sr-only">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Services</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Contact</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```



```

    </ul>
  </div>
</div>
</nav>

<div class="container">

  <div class="row">
    <div class="col-sm-8">
      <form action="#" th:action="@{/add-client}"
th:object="${client}" method="post">
        <div class="form-group">
          <input type="text" th:field="*{id}" id="id"
placeholder="Id" readonly class="form-control">
          <span th:if="${#fields.hasErrors('id')}"
th:errors="*{id}"></span>
        </div>
        <div class="form-group">
          <label for="name">Name</label>
          <input type="text" th:field="*{name}" id="name"
placeholder="Name" class="form-control">
        </div>
        <button type="submit" class="btn btn-success">Add
Client</button>
      </form>
    </div>
  </div>

  <!-- row end-->
  <div class="row">
    <div class="col-lg-12">
      <table>
        <thead>
          <tr>
            <th scope="col">Photo</th>
            <th scope="col">Id</th>
            <th scope="col">Name</th>
            <th scope="col">Actions</th>
          </tr>
        </thead>
        <tbody>
          <tr th:each="client : ${clients}">
            <td>
              <div class="event-img">
                
              </div>
            </td>
            <td th:text="${client.id}"></td>
            <td th:text="${client.name}"></td>
            <td>
              <a class="btn btn-primary"
th:href="@{||/show-client/{client.id}||}">
                <i class="fa fa-edit fa-lg"></i> Edit</a>
              <a class="btn btn-danger"
th:href="@{||/delete-client/{client.id}||}">

```

```

                <i class="fa fa-trash fa-lg"></i> Delete</a>
            </td>
        </tr>
    </tbody>
</table>
</div>
<!-- /col end-->
</div>
</div>

<script th:src="@{/webjars/jquery/3.0.0/jquery.min.js}"></script>
<script
th:src="@{/webjars/popper.js/1.12.9-1/umd/popper.min.js}"></script>
<script
th:src="@{/webjars/bootstrap/4.0.0-2/js/bootstrap.min.js}"></script>

</body>

```

14. AJOUTER LES MÉTHODES NÉCESSAIRES AU NIVEAU DU CONTROLLER :

- ★ AJOUTER LA MÉTHODE QUI FAIT L'AJOUT D'UN CLIENT AU NIVEAU DU CONTROLLER CLIENTCONTROLLER.
- ★ AJOUTER LA MÉTHODE QUI FAIT LA SUPPRESSION D'UN CLIENT AU NIVEAU DU CONTROLLER CLIENTCONTROLLER.

```

package ma.cigma.controllers;

import ma.cigma.models.Client;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.client.RestTemplate;

import java.util.List;

@Controller
public class ClientController {

    @Value("${api.url}")
    private String apiUrl;

    @Autowired
    private RestTemplate restTemplate;

```

```

@GetMapping(value = {"/", "/clients"})
public String home(Model model) {
    model.addAttribute("client", new Client());
    List<Client> clients = restTemplate.getForObject(
        apiUrl+"/clients/all",
        List.class);
    model.addAttribute("clients", clients);

    return "index-client";
}

@PostMapping(value = "/add-client")
public String addClient(Model model,
    @ModelAttribute("client") Client
client) {
    restTemplate.postForObject(
        apiUrl+"/clients/add",
        client,
        Client.class);
    return "redirect:/clients";
}

@GetMapping(value = {"/delete-client/{id}"})
public String deleteClientById(
    Model model, @PathVariable long id) {
    restTemplate.delete(apiUrl+"/clients/"+id);
    return "redirect:/clients";
}
}

```

15. DÉMARRER L'APPLICATION FRONT-END ET ACCÉDER À L'URL :
<http://localhost:8085/>

Front Application

0

Name

Name

Add Client

Photo	Id	Name	Actions
	66	AHMED	<div>Edit</div> <div>Delete</div>
	67	AHMED	<div>Edit</div> <div>Delete</div>
	68	AHMED	<div>Edit</div> <div>Delete</div>
	69	AHMED	<div>Edit</div> <div>Delete</div>
	73	dfdfdfdf	<div>Edit</div> <div>Delete</div>
	74		<div>Edit</div> <div>Delete</div>