

# *Architecture JEE et Middlewares*

## **Compte-rendu**

### *Contrôle Architecture JEE et Middlemeware*

**Nom et Prénom :** BOUSSAIRI Ikram

**E-mail :** [ikramboussairi12@gmail.com](mailto:ikramboussairi12@gmail.com)

**ENSET GLSID – S4**

**Professeur :** Mohamed YOUSSEFI



Java EE™

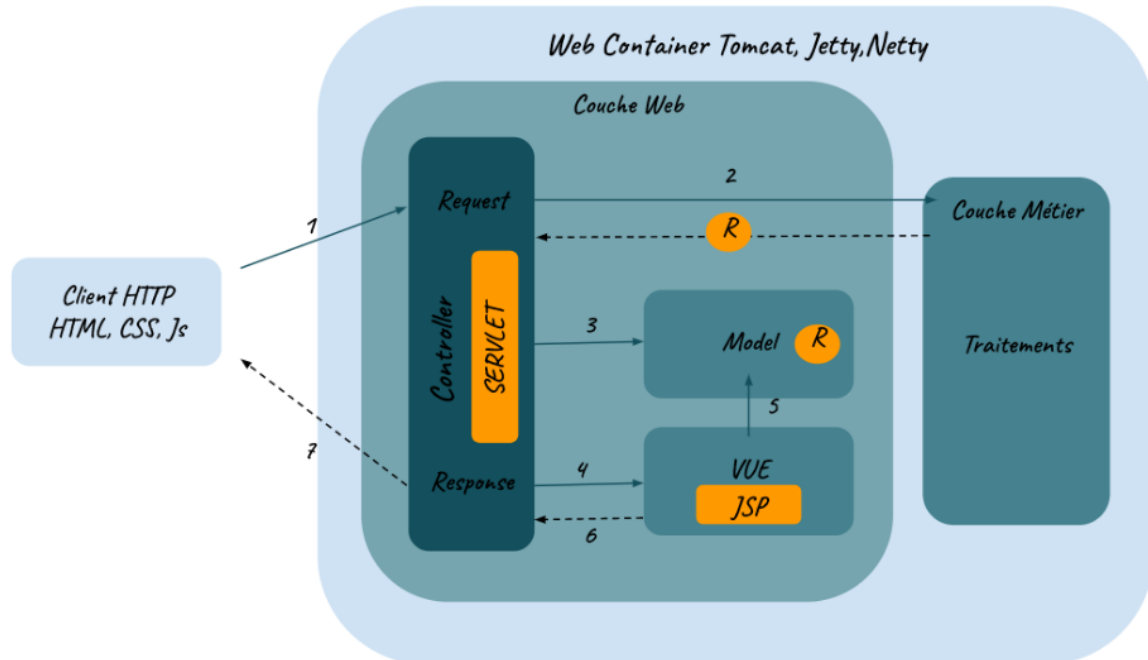


## Travail à faire

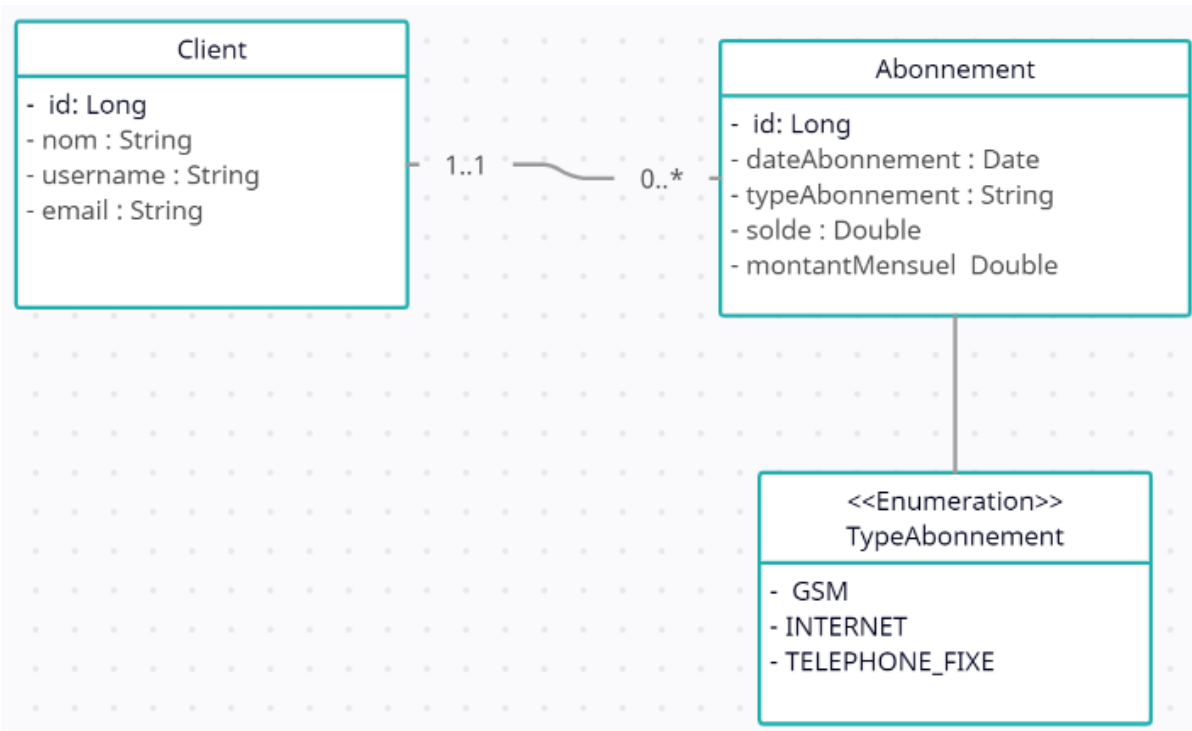
1. Schéma de l'architecture technique de l'application
2. Diagramme de classe représentant les données manipulées par l'application
3. Couche DAO :
  - a. Créer les entités JPA
  - b. Créer les interfaces JpaRepository basées sur Spring Data
  - c. Tester la couche DAO
4. Couche Web : Créer une applications Web qui permet de :
  - a. Gérer les clients (Chercher, Pagination, Ajout, Edition et Suppression)
  - b. Gérer les abonnements :
    - o Afficher les abonnements d'un clients
    - o Charger le solde de l'abonnement avec un montant
    - o Autres opérations de gestion des abonnements
5. Créer un web service RESTful qui permet de gérer les clients et les abonnements
6. Sécurité : Sécuriser l'accès à l'application en développant un système d'authentification statefull basé sur Spring Security avec deux rôles CLIENT et ADMIN. l'application doit répondre aux critères suivants :
  - a. Authentification avec le rôle CLIENT : le client ne peut voir que : son profile, ses abonnements et peut charger ses abonnements
  - b. Authentification avec le rôle ADMIN : l'administrateur peut gérer les clients et les abonnements avec tous les droits possibles. En plus il peut créer de nouveau utilisateurs et affecter des rôles aux utilisateurs.

## Réalisation

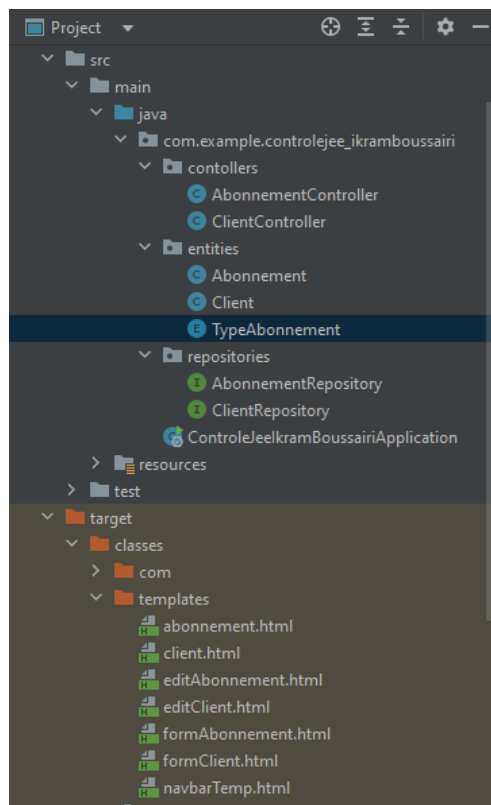
### 1. Schéma de l'architecture technique de l'application



### 2. Diagramme de classe représentant les données manipulées par l'application

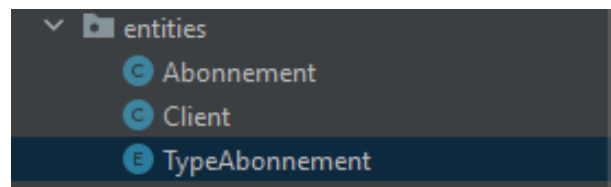


## Structure de projet



### 3. Couche DAO :

#### a. Créer les entités JPA



### Client :

```
Client.java x Abonnement.java x TypeAbonnement.java x
1 package com.example.controlejee_ikramboussairi.entities;
2
3
4 import jakarta.persistence.*;
5 import lombok.AllArgsConstructor;
6 import lombok.Data;
7 import lombok.NoArgsConstructor;
8
9 import java.util.Collection;
10
11 @Entity
12 @Data @AllArgsConstructor @NoArgsConstructor
13 public class Client {
14     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String nom;
17     private String username;
18     private String email;
19
20     @OneToMany(mappedBy = "client", fetch = FetchType.LAZY)
21     private Collection<Abonnement> abonnements;
22 }
```

## Abonnement :

```
Client.java x Abonnement.java x TypeAbonnement.java x
9
10 import java.util.Date;
11
12 14 usages
13 @Entity
14 @Data @AllArgsConstructor @NoArgsConstructor
15 public class Abonnement {
16     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Long id;
18     @DateTimeFormat(pattern = "yyyy-mm-dd")
19     @Temporal(TemporalType.DATE)
20     private Date dateAbonnement;
21     private double soldeAbonnement;
22     private double montantMensuel;
23     @Enumerated(EnumType.STRING)
24     private TypeAbonnement typeAbonnement;
25
26     @ManyToOne
27     @JoinColumn(name="clientId")
28     private Client client;
```

## TypeAbonnement :

```
Client.java x Abonnement.java x TypeAbonnement.java x
1 package com.example.controlejee_ikramboussairi.entities;
2
3 public enum TypeAbonnement {
4     GSM, INTERNET, TELEPHONE_FIXE
5 }
```

b. Créer les interfaces JpaRepository basées sur Spring Data

```
repositories
  I AbonnementRepository
  I ClientRepository
```

```
ClientRepository.java x AbonnementRepository.java x
1 package com.example.controlejee_ikramboussairi.repositories;
2
3 import com.example.controlejee_ikramboussairi.entities.Client;
4 import org.springframework.data.domain.Page;
5 import org.springframework.data.domain.Pageable;
6 import org.springframework.data.jpa.repository.JpaRepository;
7
8 6 usages
9 public interface ClientRepository extends JpaRepository<Client, Long> {
10     1 usage
11     Page<Client> findByNomContains(String keyword, Pageable pageable);
12 }
```

```

ClientRepository.java x AbonnementRepository.java x
1 package com.example.controlejee_ikramboussairi.repositories;
2
3 import com.example.controlejee_ikramboussairi.entities.Abonnement;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 import java.util.List;
7
8 public interface AbonnementRepository extends JpaRepository<Abonnement, Long> {
9     List<Abonnement> findAbonnementsByClient_Id(Long id);
10

```

### c. Tester la couche DAO

```

ControleJeeIkramBoussairiApplication.java x
15 @SpringBootApplication
16 public class ControleJeeIkramBoussairiApplication implements CommandLineRunner {
17
18     @Autowired
19     ClientRepository clientRepository;
20
21     @Autowired
22     AbonnementRepository abonnementRepository;
23
24     public static void main(String[] args) { SpringApplication.run(ControleJeeIkramBoussairiApplication.class, args); }
25
26
27     @Override
28     public void run(String... args) throws Exception {
29         clientRepository.save(new Client(id: null, nom: "ikram", username: "ikram@gmail.com", email: "ikramUser", abonnements: null));
30         clientRepository.save(new Client(id: null, nom: "imane", username: "imane@gmail.com", email: "imaneUser", abonnements: null));
31         clientRepository.save(new Client(id: null, nom: "afaf", username: "afaf@gmail.com", email: "afafUser", abonnements: null));
32         clientRepository.save(new Client(id: null, nom: "saad", username: "saad@gmail.com", email: "saadUser", abonnements: null));
33         clientRepository.save(new Client(id: null, nom: "ikram", username: "ikram@gmail.com", email: "ikramUser", abonnements: null));
34         clientRepository.save(new Client(id: null, nom: "imane", username: "imane@gmail.com", email: "imaneUser", abonnements: null));
35         clientRepository.save(new Client(id: null, nom: "afaf", username: "afaf@gmail.com", email: "afafUser", abonnements: null));
36         clientRepository.save(new Client(id: null, nom: "saad", username: "saad@gmail.com", email: "saadUser", abonnements: null));
37

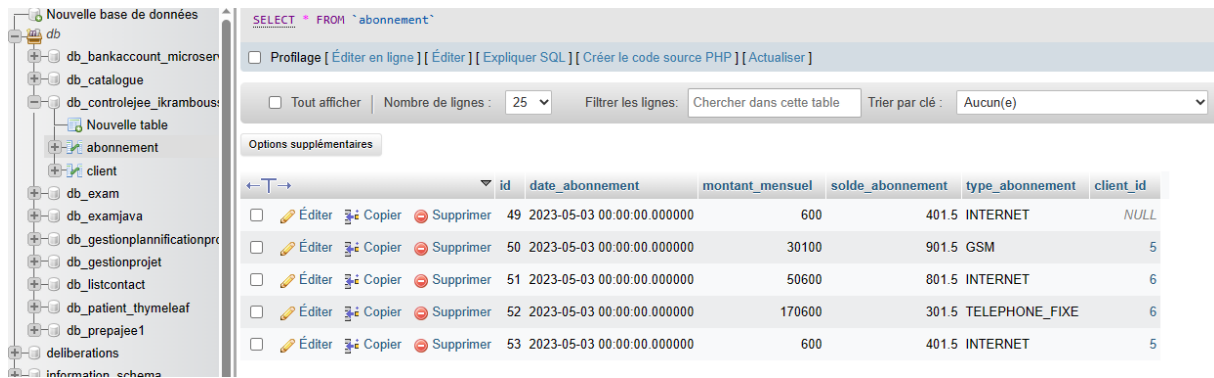
```

```

37
38
39 Client client1 = clientRepository.findById(3L).orElse( other: null);
40 Client client2 = clientRepository.findById(2L).orElse( other: null);
41
42 abonnementRepository.save(new Abonnement(id: null, new Date(), soldesAbonnement: 901.5, montantMensuel: 30100, TypeAbonnement.GSM, client1));
43 abonnementRepository.save(new Abonnement(id: null, new Date(), soldesAbonnement: 801.5, montantMensuel: 50600, TypeAbonnement.INTERNET, client2));
44 abonnementRepository.save(new Abonnement(id: null, new Date(), soldesAbonnement: 301.5, montantMensuel: 170600, TypeAbonnement.TELEPHONE_FIX, client1));
45 abonnementRepository.save(new Abonnement(id: null, new Date(), soldesAbonnement: 401.5, montantMensuel: 600, TypeAbonnement.INTERNET, client1));
46
47

```

<div>Nouvelle base de données</div> <ul style="list-style-type: none"> <li>db</li> <li>db_bankaccount_microsen</li> <li>db_catalogue</li> <li>db_controlejee_ikrambous:</li> <li>Nouvelle table</li> <li>abonnement</li> <li>client</li> <li>db_exam</li> <li>db_examjava</li> <li>db_gestionplannificationpr</li> <li>db_gestionprojet</li> <li>db_listcontact</li> <li>db_patient_thymeleaf</li> <li>db_prepajee1</li> <li>deliberations</li> <li>information_schema</li> </ul>	<input type="checkbox"/>	Éditer	Copier	Supprimer	4	saadUser	saad	saad@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	5	ikramUser	ikram	ikram@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	6	imaneUser	imane	imane@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	7	afafUser	afaf	afaf@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	8	saadUser	saad	saad@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	9	ikramUser	ikram	ikram@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	10	imaneUser	imane	imane@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	11	afafUser	afaf	afaf@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	12	saadUser	saad	saad@gmail.com
	<input type="checkbox"/>	Éditer	Copier	Supprimer	13	123@gmail.Com	boussairi	ikram
	<input type="checkbox"/>	Éditer	Copier	Supprimer	14	rayane123@gmail.com	rayane	rayaneUser
	<input type="checkbox"/>	Éditer	Copier	Supprimer	15	najwazraidi@gmail.com	najwa	zraidi



	id	date_abonnement	montant_mensuel	solde_abonnement	type_abonnement	client_id
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	49	2023-05-03 00:00:00.000000	600	401.5	INTERNET	NULL
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	50	2023-05-03 00:00:00.000000	30100	901.5	GSM	5
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	51	2023-05-03 00:00:00.000000	50600	801.5	INTERNET	6
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	52	2023-05-03 00:00:00.000000	170600	301.5	TELEPHONE_FIXE	6
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	53	2023-05-03 00:00:00.000000	600	401.5	INTERNET	5

#### 4. Couche Web : Créer une applications Web qui permet de :

- Gérer les clients (Chercher, Pagination, Ajout, Edition et Suppression)

#### Controller

```
package com.example.controlejee_ikramboussairi.controllers;

import com.example.controlejee_ikramboussairi.entities.Abonnement;
import com.example.controlejee_ikramboussairi.entities.Client;
import com.example.controlejee_ikramboussairi.repositories.AbonnementRepository;
import com.example.controlejee_ikramboussairi.repositories.ClientRepository;
import jakarta.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.util.List;

@Controller
public class ClientController {
    @Autowired
    private ClientRepository clientRepository;

    @Autowired
    private AbonnementRepository abonnementRepository;

    @GetMapping("/index")
    public String ressources(Model model,
                             @RequestParam(name="page", defaultValue = "0")
int page,
                             @RequestParam(name="size", defaultValue = "5")
int size,
                             @RequestParam(name="keyword", defaultValue =
"") String keyword){
        Page<Client> pageClient =
clientRepository.findByNomContains(keyword, PageRequest.of(page, size));
        model.addAttribute("listClient", pageClient.getContent());
    }
}
```



```

        model.addAttribute("pages", new int[pageClient.getTotalPages()]);
        model.addAttribute("currentPage",page);
        model.addAttribute("keyword",keyword);

        return "client";
    }

    @GetMapping("/")
    public String home(){
        return "redirect:/index";
    }

    @GetMapping("/delete")
    public String delete(@RequestParam Long id, String keyword, int page){

        List<Abonnement> abonnements =
abonnementRepository.findAbonnementsByClient_Id(id);
        for (Abonnement a : abonnements ) {
            abonnementRepository.delete(a);
        }

        clientRepository.deleteById(id);
        return "redirect:/index?page="+page+"&keyword="+keyword;
    }

    @GetMapping("/formClient")
    public String formRessources(Model model){
        model.addAttribute("client", new Client());
        return "formClient";
    }

    @PostMapping("/save")
    public String save(Model model, @Valid Client client, BindingResult
bindingResult,
                        @RequestParam(defaultValue = "0") int page,
                        @RequestParam(defaultValue = "") String keyword){
        if(bindingResult.hasErrors()) return "formPatients";

        clientRepository.save(client);
        return "redirect:/index?page="+page+"&keyword="+keyword;
    }

    @GetMapping("/editClient")
    public String editClient(Model model, Long id, String keyword, int
page){
        Client client = clientRepository.findById(id).orElse(null);
        if(client==null) throw new RuntimeException("Patient introuvable");

        model.addAttribute("client",client);
        model.addAttribute("page",page);
        model.addAttribute("keyword",keyword);

        return "editClient";
    }
}

```

## Templates

### - Client

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
      layout:decorate="navbarTemp">

<head>
  <meta charset="UTF-8">
  <title>Client</title>
  <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
  <Link rel="stylesheet" href="/webjars/bootstrap-
icons/1.10.3/font/bootstrap-icons.css">

</head>
<body>
<div layout:fragment="content1">
<div>
  <div class="container mt-4">
    <div class="card">
      <div class="card-header">
        liste des clients
      </div>
      <div class="card-body">
        <form method="get" th:action="@{/index}">
          <label>keyword</label>
          <input type="text" name="keyword"
th:value="${keyword}">
          <button type="submit" class="btn btn-primary"><i
class="bi bi-search"></i></button>
        </form>

        <table class="table">
          <thead>
            <tr>
              <th>Id</th>
              <th>Nom</th>
              <th>UserName</th>
              <th>email</th>
              <th>Abonnement</th>

            </tr>
          </thead>
          <tbo>
            <tr th:each="c:${listClient}">
              <td th:text="${c.getId()}"></td>
              <td th:text="${c.getNom()}"></td>
              <td th:text="${c.getEmail()}"></td>
              <td th:text="${c.getUsername()}"></td>
              <td>
                <a th:href="@{/abonnement(id=${c.id})}"
type="button" class="btn btn-outline-primary">Abonnement <i class="bi bi-
caret-right"></i></a>

              </td>

            <td>
              <a th:href="@{/editClient(id=${c.getId()},
keyword=${keyword}, page=${currentPage})}" class="btn btn-success"><i
```

```

class="bi bi-pencil"></i></a>
        </td>

        <td>
            <a onclick="return confirm('etes vous
sure?') "
                th:href="@{/delete(id=${c.id},
keyword=${keyword}, page=${currentPage})}" class="btn btn-danger"><i
class="bi bi-trash3"></i></a>
        </td>

    </tr>
</tbody>
</table>
<ul class="nav nav-pills">
    <li th:each="page,status:${pages}">
        <a th:class="${status.index==currentPage?'btn btn-
primary ms-1' : 'btn btn-outline-primary ms-1'}"
            th:text="${status.index}"
            th:href="@{index(page=${status.index},
keyword=${keyword})}"></a>
    </li>
</ul>
</div>
</div>
</div>
</div>
</body>
</html>

```

## - Edit Client

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
    xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
    layout:decorate="navbarTemp">

<head>
    <meta charset="UTF-8">
    <title>edit Client</title>
    <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
</head>
<body>

<div layout:fragment="content1">
    <div class="col-md-4 offset-4 mt-4">
        <form method="post" th:action="@{/save(page=${page},
keyword=${keyword})}">

            <div class="mt-4">
                <label>Id </label>
                <label th:text="${client.id}"> </label>
                <input class="form-control" type="hidden" name="id"
th:value="${client.id}">
            </div>

            <div class="mt-4">
                <label>Nom</label>

```

```

        <input class="form-control" type="text" name="nom"
th:value="${client.nom}">
        <span class="text-danger" th:errors="${client.nom}"></span>
    </div>
    <div class="mt-4">
        <label>UserName</label>
        <input class="form-control" type="text" name="username"
th:value="${client.username}">
        <span class="text-danger" th:errors="${client.username}"></span>
    </div>
    <div class="mt-4">
        <label>Email</label>
        <input class="form-control" type="text" name="email"
th:value="${client.email}">
        <span class="text-danger" th:errors="${client.email}"></span>
    </div>

    <div class="offset-5 mt-4">
        <button type="submit" class="btn btn-primary "> save</button>
    </div>
</form>
</div>
</div>
</body>
</html>

```

## - Form Client

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
    xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
    layout:decorate="navbarTemp">

<head>
    <meta charset="UTF-8">
    <title>Form Client</title>
    <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
</head>
<body>

<div layout:fragment="content1">
    <div class="col-md-4 offset-4 mt-4">
        <form method="post" th:action="@{/save}">
            <div class="mt-4">
                <label class="form-label">Nom</label>
                <input class="form-control" type="text" name="nom"
th:value="${client.nom}">
                <span class="text-danger form-text"
th:errors="${client.nom}"></span>
            </div>

            <div class="mt-4">
                <label class="form-label">userName</label>
                <input class="form-control" type="text" name="username"
th:value="${client.username}">
                <span class="text-danger form-text"
th:errors="${client.username}"></span>
            </div>

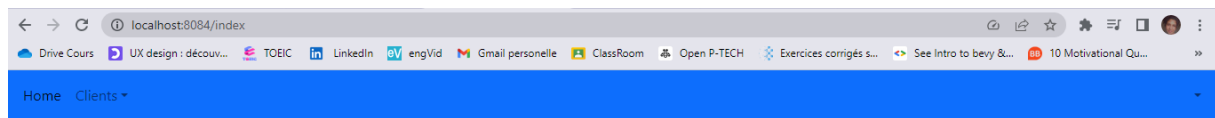
```

```

<div class="mt-4">
  <label class="form-label">email</label>
  <input class="form-control" type="email" name="email"
th:value="{client.email}">
  <span class="text-danger form-text"
th:errors="{client.email}"></span>
</div>

<div class="offset-5 mt-4">
  <button type="submit" class="btn btn-primary "> Submit</button>
</div>
</form>
</div>
</div>
</body>
</html>

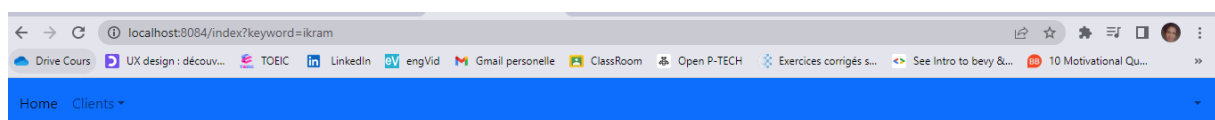
```



liste des clients

keyword

Id	Nom	UserName	email	Abonnement		
4	saad	saadUser	saad@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>
5	ikram	ikramUser	ikram@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>
6	imane	imaneUser	imane@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>
7	afaf	afafUser	afaf@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>
8	saad	saadUser	saad@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>



liste des clients

keyword

Id	Nom	UserName	email	Abonnement		
5	ikram	ikramUser	ikram@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>
9	ikram	ikramUser	ikram@gmail.com	<a href="#">Abonnement ▾</a>	<input type="button" value="✎"/>	<input type="button" value="🗑"/>

Id 9

Nom

ikram12

UserName

ikram12@gmail.com

Email

ikramUser12

save

liste des clients

keyword



Id	Nom	UserName	email	Abonnement
9	ikram12	ikramUser12	ikram12@gmail.com	Abonnement ▶



Nom

BOUSSAIRI

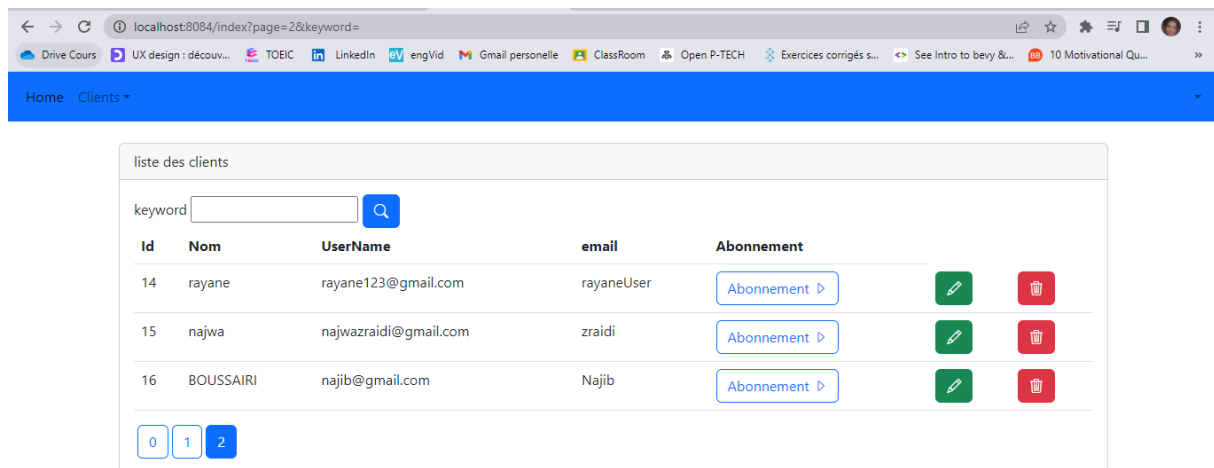
userName

Najib

email

najib@gmail.com

Submit



## b. Gérer les abonnements :

### Controller

```
package com.example.controlejee_ikramboussairi.controllers;

import com.example.controlejee_ikramboussairi.entities.Abonnement;
import com.example.controlejee_ikramboussairi.entities.Client;

import com.example.controlejee_ikramboussairi.repositories.AbonnementRepository;
import com.example.controlejee_ikramboussairi.repositories.ClientRepository;
import jakarta.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import java.util.Date;
import java.util.List;

@Controller
public class AbonnementController {

    Long id_Client;

    @Autowired
    private AbonnementRepository abonnementRepository;
    @Autowired
    private ClientRepository clientRepository;

    @GetMapping("/abonnement/{id}")
    public String abonnementRessources (Model model, @RequestParam Long id) {

        List<Abonnement> abonnements =
abonnementRepository.findAbonnementsByClient_Id(id);
```

```

        model.addAttribute("listAbonnement", abonnements);

        id_Client = id;

        return "abonnement";
    }

    @PostMapping("/saveAbonnement/{id}")
    public String saveAbonnement(Model model, @Valid Abonnement
abonnement, @RequestParam Long clientId, BindingResult bindingResult) {
        Client client = clientRepository.findById(clientId).orElse(null);

        if(client!=null) abonnement.setClient(client);

        abonnementRepository.save(abonnement);
        return "redirect:/abonnement?id="+clientId;
    }

    @GetMapping("/editAbonnement/{id}")
    public String editAbonnement(Model model, @RequestParam Long id) {
        Abonnement abonnement =
abonnementRepository.findById(id).orElse(null);
        if(abonnement==null) throw new RuntimeException("Abonnement
introuvable");

        model.addAttribute("abonnement", abonnement);

        return "editAbonnement";
    }

    @GetMapping("/formAbonnement")
    public String formRessources(Model model) {
        Abonnement abonnement = new Abonnement();

        abonnement.setDateAbonnement(new Date());

        Client client = clientRepository.findById(id_Client).orElse(null);
        abonnement.setClient(client);

        model.addAttribute("abonnement", abonnement);
        return "formAbonnement";
    }

    @GetMapping("/deleteAbonnement/{id}")
    public String deleteAbonnement(@RequestParam Long id) {

        abonnementRepository.deleteById(id);
        return "redirect:/abonnement?id=" + id_Client;
    }
}

```

## Templates

### - Abonnement



```

- <!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
      layout:decorate="navbarTemp">

<head>
  <meta charset="UTF-8">
  <title>Abonnement</title>
  <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
  <Link rel="stylesheet" href="/webjars/bootstrap-
icons/1.10.3/font/bootstrap-icons.css">

</head>
<body>
<div layout:fragment="content1">
<div>
  <div class="container mt-4">
    <div class="card">
      <div class="card-header">
        liste des Abonnements
      </div>

      <div class="card-body">

        <div>
          <a th:href="@{/formAbonnement}" type="button"
class="btn btn-primary">Nouvelle Abonnement <i class="bi bi-
plus"></i></a>
        </div>

        <table class="table">
          <thead>
            <tr>
              <th>Id</th>
              <th>Type </th>
              <th>Date Abonnement</th>
              <th>Solde</th>
              <th>Montant Mensuel</th>
              <th>Client Id </th>

            </tr>
          </thead>
          <tbo>
            <tr th:each="a:${listAbonnement}">
              <td th:text="${a.getId()}"></td>
              <td
th:text="${a.getTypeAbonnement().toString()}"></td>
              <td
th:text="${a.getDateAbonnement()}"></td>
              <td
th:text="${a.getSoldeAbonnement()}"></td>
              <td
th:text="${a.getMontantMensuel()}"></td>
              <td
th:text="${a.getClient().getId()}"></td>

              <td>
                <a
th:href="@{/editAbonnement(id=${a.getId()})}" class="btn btn-

```

```

success"><i class="bi bi-pencil"></i></a>
                </td>
                <td>
                    <a
th:href="@{/deleteAbonnement(id=${a.getId()})}" onclick="return
confirm('etes vous sure?') "
                    class="btn btn-danger"><i
class="bi bi-trash3"></i></a>
                </td>
            </tr>
        </tbody>
    </table>

    <!-- <ul class="nav nav-pills">
        <li th:each="page,status:${pages}">
            <a th:class="${status.index==currentPage?'btn
btn-primary ms-1' : 'btn btn-outline-primary ms-1'}"
                th:text="${status.index}"

th:href="@{abonnement(page=${status.index},
keyword=${keyword})}"></a>
            </li>
        </ul>-->

    </div>
</div>
</div>
</div>
</body>
</html>

```

## - Edit Abonnement

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
    xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
    layout:decorate="navbarTemp">

<head>
    <meta charset="UTF-8">
    <title>edit Abonnement</title>
    <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
</head>
<body>

<div layout:fragment="content1">
    <div class="col-md-4 offset-4 mt-4">
        <form method="post"
th:action="@{/saveAbonnement(clientId=${abonnement.getClient().getId()})}">

            <div class="mt-4">
                <label>Id : </label>
                <label th:text="${abonnement.id}"> </label>
                <input class="form-control" type="hidden" name="id"
th:value="${abonnement.id}">

```

```

        </div>

        <div class="mt-4">
            <label>Type Abonnement : </label>
            <label th:text="\${abonnement.typeAbonnement}"> </label>
            <input class="form-control" type="hidden" name="typeAbonnement"
th:value="\${abonnement.typeAbonnement}">
        </div>

        <div class="mt-4">
            <label>Date Abonnement : </label>
            <label th:text="\${abonnement.dateAbonnement.toString()}">
</label>
            <input class="form-control" type="hidden" name="dateAbonnement"
th:value="\${abonnement.dateAbonnement}">
        </div>

        <div class="mt-4">
            <label>Solde</label>
            <input class="form-control" type="number" step="any"
name="soldeAbonnement" th:value="\${abonnement.soldeAbonnement}">
            <span class="text-danger"
th:errors="\${abonnement.soldeAbonnement}"></span>
        </div>

        <div class="mt-4">
            <label>Montant Mensuel</label>
            <input class="form-control" type="number" step="any"
name="montantMensuel" th:value="\${abonnement.montantMensuel}">
            <span class="text-danger"
th:errors="\${abonnement.montantMensuel}"></span>
        </div>

        <div class="offset-5 mt-4">
            <button type="submit" class="btn btn-primary ">
saveAbonnement</button>
        </div>
    </form>
</div>
</div>
</body>
</html>

```

## - Form Abonnement

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
layout:decorate="navbarTemp">

<head>
    <meta charset="UTF-8">
    <title>Form Abonnement</title>
    <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
</head>
<body>

```

```

<div layout:fragment="content1">
  <div class="col-md-4 offset-4 mt-4">
    <form method="post"
th:action="@{/saveAbonnement(clientId=${abonnement.getClient().getId()})}">

      <div class="mt-4">
        <label>Date Abonnement </label>
        <input class="form-control" type="date" name="dateAbonnement"
th:value="${abonnement.dateAbonnement}">
      </div>

      <div class="mt-4">
        <label class="form-label">Type Abonnement</label>
        <select name="typeAbonnement" class="form-select" aria-
label="Default select example">
          <option selected value="GSM">GSM</option>
          <option th:value="INTERNET">INTERNET</option>
          <option th:value="TELEPHONE_FIXE">TELEPHONE_FIXE</option>
        </select>
        <span class="text-danger form-text"
th:errors="${abonnement.typeAbonnement}"></span>
      </div>

      <div class="mt-4">
        <label>Type Abonnement : </label>
        <label th:text="${abonnement.typeAbonnement}"> </label>
        <input class="form-control" type="hidden" name="typeAbonnement"
th:value="${abonnement.typeAbonnement}">
      </div>

      <div class="mt-4">
        <label class="form-label">Solde</label>
        <input class="form-control" type="number" step="any"
name="soldeAbonnement" th:value="${abonnement.soldeAbonnement}">
        <span class="text-danger form-text"
th:errors="${abonnement.soldeAbonnement}"></span>
      </div>

      <div class="mt-4">
        <label class="form-label">Montant Mensuel</label>
        <input class="form-control" type="number" step="any"
name="montantMensuel" th:value="${abonnement.montantMensuel}">
        <span class="text-danger form-text"
th:errors="${abonnement.montantMensuel}"></span>
      </div>

      <div class="offset-5 mt-4">
        <button type="submit" class="btn btn-primary "> Submit</button>
      </div>
    </form>
  </div>
</div>
</body>
</html>

```

liste des Abonnements

[Nouvelle Abonnement +](#)

Id	Type	Date Abonnement	Solde	Montant Mensuel	Client Id		
51	INTERNET	2023-05-03	801.5	50600.0	6		
52	TELEPHONE_FIXE	2023-05-03	301.5	170600.0	6		

Date Abonnement  
03/05/2023

Type Abonnement  
GSM

Type Abonnement :

Solde  
200

Montant Mensuel  
200

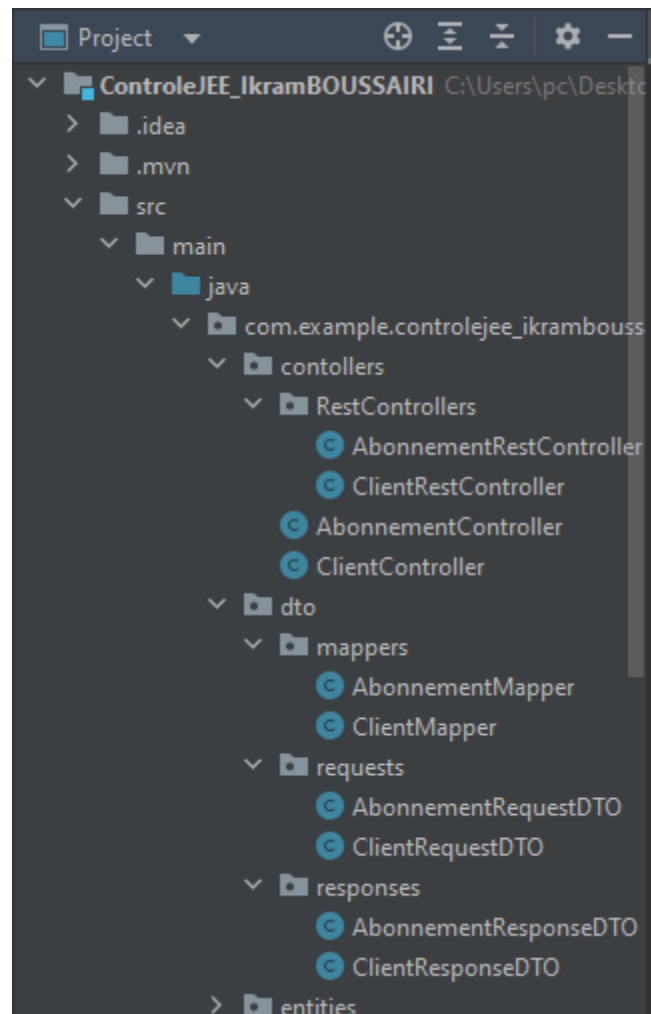
[Submit](#)

liste des Abonnements

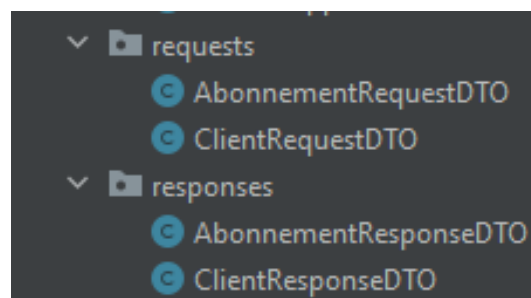
[Nouvelle Abonnement +](#)

Id	Type	Date Abonnement	Solde	Montant Mensuel	Client Id		
51	INTERNET	2023-05-03	801.5	50600.0	6		
52	TELEPHONE_FIXE	2023-05-03	301.5	170600.0	6		
62	GSM	2023-01-05	200.0	200.0	6		

5. Créer un web service RESTful qui permet de gérer les clients et les abonnements



- DTOs



```

import com.example.controlejee_ikramboussairi.entities.Client;
import com.example.controlejee_ikramboussairi.entities.TypeAbonnement;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.format.annotation.DateTimeFormat;
import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class AbonnementRequestDTO {

    private Date dateAbonnement;
    private Double soldeAbonnement;
    private Double montantMensuel;
    private TypeAbonnement typeAbonnement;
}

```

```

package com.example.controlejee_ikramboussairi.dto.response;

import com.example.controlejee_ikramboussairi.entities.Abonnement;
import jakarta.persistence.FetchType;
import jakarta.persistence.OneToMany;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.util.Collection;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class ClientRequestDTO {

    private String nom;
    private String username;
    private String email;
}

```

```

import com.example.controlejee_ikramboussairi.entities.Client;
import com.example.controlejee_ikramboussairi.entities.TypeAbonnement;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class AbonnementResponseDTO {

    private Long id;
    private Date dateAbonnement;
    private Double soldeAbonnement;
    private Double montantMensuel;
    private TypeAbonnement typeAbonnement;
    private Long clientId;
}

```

```

package com.example.controlejee_ikramboussairi.dto.response;

import com.example.controlejee_ikramboussairi.entities.Abonnement;
import jakarta.persistence.FetchType;
import jakarta.persistence.OneToMany;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class ClientResponseDTO {

    private Long id;
    private String nom;
    private String username;
    private String email;
}

```

## - Mappers

```

package com.example.controlejee_ikramboussairi.dto.mappers;

import com.example.controlejee_ikramboussairi.dto.responses.ClientResponseDTO;
import com.example.controlejee_ikramboussairi.entities.Client;
import org.springframework.beans.BeanUtils;
import org.springframework.stereotype.Component;

@Component
public class ClientMapper {

    public ClientResponseDTO fromClient(Client client){
        ClientResponseDTO clientResponseDTO = new ClientResponseDTO();
        BeanUtils.copyProperties(client, clientResponseDTO);

        return clientResponseDTO;
    }
}

```

```

package com.example.controlejee_ikramboussairi.dto.mappers;

import com.example.controlejee_ikramboussairi.dto.responses.AbonnementResponseDTO;
import com.example.controlejee_ikramboussairi.entities.Abonnement;
import org.springframework.beans.BeanUtils;
import org.springframework.stereotype.Component;

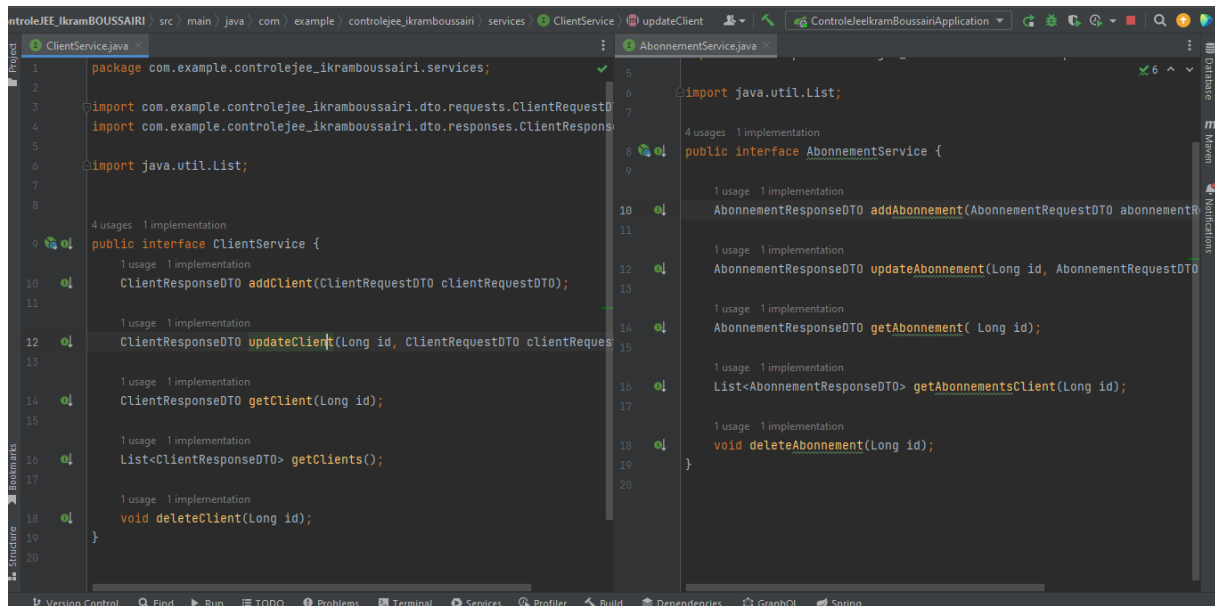
@Component
public class AbonnementMapper {

    public AbonnementResponseDTO fromAbonnement(Abonnement abonnement, Long id){
        AbonnementResponseDTO abonnementResponseDTO = new AbonnementResponseDTO();
        abonnementResponseDTO.setClientId(id);
        BeanUtils.copyProperties(abonnement, abonnementResponseDTO);

        return abonnementResponseDTO;
    }
}

```

## - Interface Service



## - Services Implémentation

### Client

```
package com.example.controlejee_ikramboussairi.services.serviceImp;

import com.example.controlejee_ikramboussairi.dto.mappers.ClientMapper;
import com.example.controlejee_ikramboussairi.dto.requests.ClientRequestDTO;
import com.example.controlejee_ikramboussairi.dto.responses.ClientResponseDTO;
import com.example.controlejee_ikramboussairi.entities.Abonnement;
import com.example.controlejee_ikramboussairi.entities.Client;
import com.example.controlejee_ikramboussairi.repositories.AbonnementRepository;
import com.example.controlejee_ikramboussairi.repositories.ClientRepository;
import com.example.controlejee_ikramboussairi.services.ClientService;
import lombok.AllArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.ArrayList;
import java.util.List;

@Service
@Transactional
@AllArgsConstructor
public class ClientServiceImp implements ClientService {

    private ClientRepository clientRepository;
    private ClientMapper clientMapper;

    private AbonnementRepository abonnementRepository;

    @Override
    public ClientResponseDTO addClient(ClientRequestDTO clientRequestDTO) {
        Client client = new Client();
        client.setNom(clientRequestDTO.getNom());
        client.setUsername(clientRequestDTO.getUsername());
        client.setEmail(clientRequestDTO.getEmail());
    }
}
```



```

        Client saveClient = clientRepository.save(client);

        return clientMapper.fromClient(saveClient);
    }

    @Override
    public ClientResponseDTO updateClient(Long id, ClientRequestDTO clientRequestDTO) {

        Client client = clientRepository.findById(id).orElse(null);
        if(client!=null){
            if(clientRequestDTO.getNom()!=null)
client.setNom(clientRequestDTO.getNom());
            if(clientRequestDTO.getUsername()!=null)
client.setUsername(clientRequestDTO.getUsername());
            if(clientRequestDTO.getEmail()!=null)
client.setEmail(clientRequestDTO.getEmail());
        }

        Client clientUpdated = clientRepository.save(client);

        return clientMapper.fromClient(clientUpdated);
    }

    @Override
    public ClientResponseDTO getClient(Long id) {
        Client client = clientRepository.findById(id).orElse(null);

        return clientMapper.fromClient(client);
    }

    @Override
    public List<ClientResponseDTO> getClients() {
        List<Client> clients = clientRepository.findAll();

        List<ClientResponseDTO> responseDTOS = new ArrayList<>();

        for(Client c : clients){
            ClientResponseDTO clientResponseDTO = new ClientResponseDTO();
            clientResponseDTO = clientMapper.fromClient(c);
            responseDTOS.add(clientResponseDTO);
        }
        return responseDTOS;
    }

    @Override
    public void deleteClient(Long id) {
        List<Abonnement> abonnements =
abonnementRepository.findAbonnementsByClient_Id(id);
        for (Abonnement a : abonnements ) {
            abonnementRepository.delete(a);
        }
        clientRepository.deleteById(id);
    }
}

```

## Abonnement

```

package com.example.controlejee_ikramboussairi.services.serviceImp;

import com.example.controlejee_ikramboussairi.dto.mappers.AbonnementMapper;
import com.example.controlejee_ikramboussairi.dto.requests.AbonnementRequestDTO;
import com.example.controlejee_ikramboussairi.dto.requests.ClientRequestDTO;
import com.example.controlejee_ikramboussairi.dto.responses.AbonnementResponseDTO;
import com.example.controlejee_ikramboussairi.dto.responses.ClientResponseDTO;
import com.example.controlejee_ikramboussairi.entities.Abonnement;
import com.example.controlejee_ikramboussairi.entities.Client;
import com.example.controlejee_ikramboussairi.repositories.AbonnementRepository;
import com.example.controlejee_ikramboussairi.repositories.ClientRepository;
import com.example.controlejee_ikramboussairi.services.AbonnementService;
import lombok.AllArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.ArrayList;
import java.util.List;

@Service
@AllArgsConstructor @Transactional
public class AbonnementServiceImp implements AbonnementService {

    private AbonnementRepository abonnementRepository;
    private AbonnementMapper abonnementMapper;
    private ClientRepository clientRepository;

    @Override
    public AbonnementResponseDTO addAbonnement (AbonnementRequestDTO
abonnementRequestDTO, Long id) {
        Abonnement abonnement = new Abonnement();

abonnement.setDateAbonnement (abonnementRequestDTO.getDateAbonnement());

abonnement.setTypeAbonnement (abonnementRequestDTO.getTypeAbonnement());

abonnement.setSoldeAbonnement (abonnementRequestDTO.getSoldeAbonnement());

abonnement.setMontantMensuel (abonnementRequestDTO.getMontantMensuel());
        abonnement.setClient (clientRepository.findById(id).orElse(null));

        Abonnement abonnementsaved = abonnementRepository.save(abonnement);

        return abonnementMapper.fromAbonnement (abonnementsaved, id);
    }

    @Override
    public AbonnementResponseDTO updateAbonnement (Long id,
AbonnementRequestDTO abonnementRequestDTO) {
        Abonnement abonnement =
abonnementRepository.findById(id).orElse(null);
        if (abonnement!=null) {
            if (abonnementRequestDTO.getDateAbonnement() !=null)
abonnement.setDateAbonnement (abonnementRequestDTO.getDateAbonnement());
            if (abonnementRequestDTO.getTypeAbonnement() !=null)

```

```

abonnement.setTypeAbonnement (abonnementRequestDTO.getTypeAbonnement () );
        if (abonnementRequestDTO.getSoldeAbonnement () !=null)
abonnement.setSoldeAbonnement (abonnementRequestDTO.getSoldeAbonnement () );
        if (abonnementRequestDTO.getMontantMensuel () !=null)
abonnement.setMontantMensuel (abonnementRequestDTO.getMontantMensuel () );
    }

    Abonnement abonnementUpdated =
abonnementRepository.save (abonnement);
    return abonnementMapper.fromAbonnement (abonnementUpdated,
abonnement.getClient ().getId ());
}

@Override
public AbonnementResponseDTO getAbonnement (Long id) {
    Abonnement abonnement =
abonnementRepository.findById (id).orElse (null);

    return
abonnementMapper.fromAbonnement (abonnement, abonnement.getClient ().getId ());
}

@Override
public List<AbonnementResponseDTO> getAbonnementsClient (Long idClient)
{
    List<Abonnement> abonnements =
abonnementRepository.findAbonnementsByClient_Id (idClient);
    List<AbonnementResponseDTO> ResponseDTOS = new ArrayList<> ();

    for (Abonnement a:abonnements){
        AbonnementResponseDTO abonnementResponseDTO = new
AbonnementResponseDTO ();
        abonnementResponseDTO = abonnementMapper.fromAbonnement (a,
a.getClient ().getId ());
        ResponseDTOS.add (abonnementResponseDTO);
    }

    return ResponseDTOS;
}

@Override
public void deleteAbonnement (Long id) {
    abonnementRepository.deleteById (id);
}
}

```

## - Rest Controllers

### Client

```

package com.example.controlejee_ikramboussairi.controllers.RestControllers;

import
com.example.controlejee_ikramboussairi.dto.requests.ClientRequestDTO;
import
com.example.controlejee_ikramboussairi.dto.responses.ClientResponseDTO;

import com.example.controlejee_ikramboussairi.services.ClientService;

```

```

import lombok.AllArgsConstructor;

import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/rest")
@AllArgsConstructor
public class ClientRestController {
    private ClientService clientService;

    @PostMapping("/clients")
    public ClientResponseDTO saveClient(@RequestBody ClientRequestDTO clientRequestDTO) {

        return clientService.addClient(clientRequestDTO);
    }

    @GetMapping("/clients")
    public List<ClientResponseDTO> getclients() {
        List<ClientResponseDTO> clients = clientService.getClients();
        return clients;
    }

    @GetMapping("/clients/{id}")
    public ClientResponseDTO getClient(@PathVariable Long id) {
        ClientResponseDTO client = clientService.getClient(id);
        return client;
    }

    @PostMapping("/clients/{id}")
    public ClientResponseDTO updateClient(@PathVariable Long id, @RequestBody ClientRequestDTO clientRequestDTO) {

        return clientService.updateClient(id, clientRequestDTO);
    }

    @DeleteMapping("/clients/{id}")
    public void deleteClient(@PathVariable Long id) {
        clientService.deleteClient(id);
    }
}

```

## Abonnement

```

package com.example.controlejee_ikramboussairi.controllers.RestControllers;

import com.example.controlejee_ikramboussairi.dto.requests.AbonnementRequestDTO;
import com.example.controlejee_ikramboussairi.dto.responses.AbonnementResponseDTO;
import com.example.controlejee_ikramboussairi.services.AbonnementService;
import lombok.AllArgsConstructor;
import org.springframework.web.bind.annotation.*;
import java.util.List;

```

```

@RestController
@RequestMapping("/rest")
@AllArgsConstructor
public class AbonnementRestController {

    private AbonnementService abonnementService;

    @PostMapping("/abonnements/{id}")
    public AbonnementResponseDTO saveAbonnement(@RequestBody
AbonnementRequestDTO abonnementRequestDTO, @PathVariable Long id){

        return abonnementService.addAbonnement(abonnementRequestDTO, id);

    }

    @GetMapping("/abonnementsClient/{idClient}")
    public List<AbonnementResponseDTO> getAbonnementsClient(@PathVariable
Long idClient){
        List<AbonnementResponseDTO> abonnements =
abonnementService.getAbonnementsClient(idClient);
        return abonnements;
    }

    @GetMapping("/abonnement/{id}")
    public AbonnementResponseDTO getAbonnement(@PathVariable Long id){
        AbonnementResponseDTO abonnement =
abonnementService.getAbonnement(id);
        return abonnement;
    }

    @PostMapping("/abonnements/update/{id}")
    public AbonnementResponseDTO updateAbonnement(@PathVariable Long
id, @RequestBody AbonnementRequestDTO abonnementRequestDTO) {

        return abonnementService.updateAbonnement(id, abonnementRequestDTO)
;

    }

    @DeleteMapping("/abonnements/{id}")
    public void deleteAbonnement(@PathVariable Long id){
        abonnementService.deleteAbonnement(id);
    }

}

```

GET

http://localhost:8084/rest/clients

Send

200 OK

18.7 ms

723 B

JSON

Basic

Query

Headers1

Docs

Preview

Headers3

Cookies

Timeline

1 ...

```
1 [
2   {
3     "id": 5,
4     "nom": "ikram",
5     "username": "ikram@gmail.com",
6     "email": "ikramUser"
7   },
8   {
9     "id": 7,
10    "nom": "afaf",
11    "username": "afaf@gmail.com",
12    "email": "afaUser"
13  },
14  {
15    "id": 8,
16    "nom": "saad",
17    "username": "saad@gmail.com",
18    "email": "saadUser"
19  },
20  {
21    "id": 9,
22    "nom": "ikram12",
23    "username": "ikram12@gmail.com",
24    "email": "ikramUser12"
25  },
26  {
27    "id": 10,
28    "nom": "imane",
29    "username": "imane@gmail.com",
30    "email": "imaneUser"
31  },
32  ]
```

GET

http://localhost:8084/rest/clients/5

Send

200 OK

10.9 ms

71 B

JSON

Basic

Query

Headers1

Docs

Preview

Headers3

Cookies

Timeline

1 ...

```
1 {
2   "id": 5,
3   "nom": "ikram",
4   "username": "ikram@gmail.com",
5   "email": "ikramUser"
6 }
```

POST

http://localhost:8084/rest/clients/5

Send

200 OK

67.9 ms

88 B

JSON

Basic

Query

Headers1

Docs

Preview

Headers3




Cookies

Timeline

```
1 {
2   "id": 5,
3   "nom": "ikrampdate",
4   "username": "ikramUpdate@gmail.com",
5   "email": "ikramUserUpdate"
6 }
```

```
1 {
2   "id": 5,
3   "nom": "ikrampdate",
4   "username": "ikramUpdate@gmail.com",
5   "email": "ikramUserUpdate"
6 }
```

Options supplémentaires

	id	email	nom	username
<div><input type="checkbox"/></div> <div> Éditer</div> <div> Copier</div> <div> Supprimer</div>	5	ikramUserUpdate	ikrampdate	ikramUpdate@gmail.com

DELETE http://localhost:8084/rest/clients/5 Send 200 OK 41.4 ms 0 B

JSON Basic Query Headers 1 Docs Preview Headers 2 Cookies Timeline

```
1 {
2   "id": 5,
3   "nom": "ikramupdate",
4   "username": "ikramUpdate@gmail.com",
5   "email": "ikramUserUpdate"
6 }
```

No body returned for response

Nouvelle base de données db

- db\_bankaccount\_microser
- Nouvelle table
- bank\_account
- customer
- db\_catalogue
- db\_controlejee\_ikrambous
- Nouvelle table
- abonnement
- client
- db\_exam
- db\_examjava
- db\_gestionplannificationpro
- db\_gestionprojet
- db\_listcontact
- db\_patient\_thymeleaf

Options supplémentaires

					id	email	nom	username
<input type="checkbox"/>	Éditer	Copier	Supprimer		7	afaUser	afaf	afaf@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		8	saadUser	saad	saad@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		9	ikramUser12	ikram12	ikram12@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		10	imaneUser	imane	imane@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		11	afaUser	afaf	afaf@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		12	saadUser	saad	saad@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		13	123@gmail.Com	boussairi	ikram
<input type="checkbox"/>	Éditer	Copier	Supprimer		14	rayane123@gmail.com	rayane	rayaneUser
<input type="checkbox"/>	Éditer	Copier	Supprimer		16	najib@gmail.com	BOUSSAIRI	Najib

POST http://localhost:8084/rest/clients Send 200 OK 24.6 ms 72 B

JSON Basic Query Headers 1 Docs Preview Headers 3 Cookies Timeline

```
1 {
2   "nom": "Hamza",
3   "username": "hamza@gmail.com",
4   "email": "hamzaUser"
5 }
```

```
1 {
2   "id": 18,
3   "nom": "Hamza",
4   "username": "hamza@gmail.com",
5   "email": "hamzaUser"
6 }
```

Nouvelle base de données db

- db\_bankaccount\_microser
- Nouvelle table
- bank\_account
- customer
- db\_catalogue
- db\_controlejee\_ikrambous
- Nouvelle table
- abonnement
- client
- db\_exam
- db\_examjava
- db\_gestionplannificationpro
- db\_gestionprojet
- db\_listcontact
- db\_patient\_thymeleaf
- db\_prepajee1

Options supplémentaires

					id	email	nom	username
<input type="checkbox"/>	Éditer	Copier	Supprimer		7	afaUser	afaf	afaf@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		8	saadUser	saad	saad@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		9	ikramUser12	ikram12	ikram12@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		10	imaneUser	imane	imane@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		11	afaUser	afaf	afaf@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		12	saadUser	saad	saad@gmail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer		13	123@gmail.Com	boussairi	ikram
<input type="checkbox"/>	Éditer	Copier	Supprimer		14	rayane123@gmail.com	rayane	rayaneUser
<input type="checkbox"/>	Éditer	Copier	Supprimer		16	najib@gmail.com	BOUSSAIRI	Najib
<input type="checkbox"/>	Éditer	Copier	Supprimer		18	hamzaUser	Hamza	hamza@gmail.com

POST http://localhost:8084/rest/abonnements/18 Send 200 OK 52 ms 143 B

JSON Basic Query Headers 1 Docs Preview Headers 3 Cookies Timeline

```
1 {
2   "dateAbonnement": "2023-05-04",
3   "soldeAbonnement": "123.4",
4   "montantMensuel": "1000.0",
5   "typeAbonnement": "GSM"
6 }
```

```
1 {
2   "id": 65,
3   "dateAbonnement": "2023-05-04T00:00:00.000+00:00",
4   "soldeAbonnement": 123.4,
5   "montantMensuel": 1000.0,
6   "typeAbonnement": "GSM",
7   "clientId": 18
8 }
```

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

	id	date_abonnement	montant_mensuel	solde_abonnement	type_abonnement	client_id
<input type="checkbox"/> Éditer Copier Supprimer	65	2023-05-04 00:00:00.000000	1000	123.4	GSM	18

GET http://localhost:8084/rest/abonnementsClient/7 Send 200 OK 9.74 ms 250 B

JSON Basic Query Headers 1 Docs Preview Headers 3 Cookies Timeline

```
1 [
2   {
3     "id": 67,
4     "dateAbonnement": "2023-05-04",
5     "soldeAbonnement": 80.4,
6     "montantMensuel": 100.0,
7     "typeAbonnement": "INTERNET",
8     "clientId": 7
9   },
10  {
11    "id": 68,
12    "dateAbonnement": "2023-05-04",
13    "soldeAbonnement": 80.4,
14    "montantMensuel": 100.0,
15    "typeAbonnement": "GSM",
16    "clientId": 7
17  }
18 ]
```

POST http://localhost:8084/rest/abonnements/update/67 Send 200 OK 20.3 ms 142 B

JSON Basic Query Headers 1 Docs Preview Headers 3 Cookies Timeline

```
1 {
2   "id": 67,
3   "dateAbonnement": "2023-02-28",
4   "soldeAbonnement": 0.4,
5   "montantMensuel": 0.0,
6   "typeAbonnement": "INTERNET"
7 }
```

```
1 {
2   "id": 67,
3   "dateAbonnement": "2023-02-28T00:00:00.000+00:00",
4   "soldeAbonnement": 0.4,
5   "montantMensuel": 0.0,
6   "typeAbonnement": "INTERNET",
7   "clientId": 7
8 }
```

DELETE http://localhost:8084/rest/abonnements/67 Send 200 OK 13.7 ms 0 B

JSON Basic Query Headers 1 Docs Preview Headers 2 Cookies

No body returned for response

	id	date_abonnement	montant_mensuel	solde_abonnement	type_abonnement	client_id
<input type="checkbox"/> Éditer Copier Supprimer	65	2023-05-04 00:00:00.000000	1000	123.4	GSM	18
<input type="checkbox"/> Éditer Copier Supprimer	66	2023-05-04 00:00:00.000000	100	80.4	INTERNET	18
<input type="checkbox"/> Éditer Copier Supprimer	68	2023-05-04 00:00:00.000000	100	80.4	GSM	7



## 6. Sécurité :

### - Security Config

```
package com.example.controlejee_ikramboussairi.security;

import lombok.AllArgsConstructor;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
@AllArgsConstructor
public class SecurityConfig {

    private PasswordEncoder passwordEncoder;

    @Bean
    public InMemoryUserDetailsManager inMemoryUserDetailsManager() {
        return new InMemoryUserDetailsManager(
            User.withUsername("ikram").password(passwordEncoder.encode("ikram")).roles("USER").build(),
            User.withUsername("rayane").password(passwordEncoder.encode("rayane")).roles("USER").build(),
            User.withUsername("afaf").password(passwordEncoder.encode("afaf")).roles("USER", "ADMIN").build()
        );
    }

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) throws Exception {
        //httpSecurity.formLogin();

        httpSecurity.authorizeHttpRequests().requestMatchers("/webjars/**").permitAll();
        httpSecurity.formLogin().loginPage("/login").permitAll();

        httpSecurity.authorizeHttpRequests().requestMatchers("/user/**").hasRole("USER");

        httpSecurity.authorizeHttpRequests().requestMatchers("/admin/**").hasRole("ADMIN");
        httpSecurity.authorizeHttpRequests().anyRequest().authenticated();

        httpSecurity.exceptionHandling().accessDeniedPage("/notAuthorized");
    }
}
```

```

        return httpSecurity.build();
    }
}

```

## - Security Controller

```

package com.example.controlejee_ikramboussairi.security;

import org.springframework.stereotype.Component;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class SecurityController {
    @GetMapping("/notAuthorized")
    public String notAuthorized(){
        return "notAuthorized";
    }

    @GetMapping("/login")
    public String login(){
        return "login";
    }
}

```

## Templates

### - Login

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="UTF-8">
    <title>Form Ressource</title>
    <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
</head>
<body>

<div class=" mt-4">
    <div class="col-md-4 offset-4 ">
        <div class="card">
            <div class="card-header">Authentication</div>
            <div class="card-body">
                <form method="post" th:action="@{/login}">
                    <div class="mt-4">
                        <label for="username" class="form-
label">UserName</label>
                        <input id="username" class="form-control"
type="text" name="username" placeholder="username">
                    </div>

                    <div class="mt-4">
                        <label for="password" class="form-
label">Password</label>
                        <input id="password" class="form-control"

```

```

type="password" name="password" >
    </div>

    <div class="mt-4">
        <button type="submit" class="btn btn-primary ">
Login</button>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

- Not Authaurized

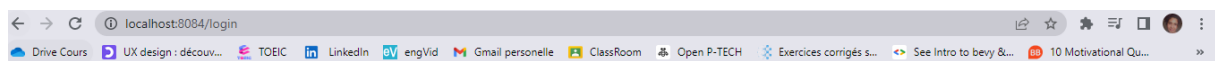
```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
    xmlns:layout="http://www.ultrap.net.nz/thymeleaf/layout"
    layout:decorate="navbarTemp">

<head>
    <meta charset="UTF-8">
    <title>not Authorized</title>
    <Link rel="stylesheet"
href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css">
</head>
<body>

<div layout:fragment="content1">
    <div class=" alert alert-danger mt-3">
        <h2>Not Authorized</h2>
    </div>
</div>
</body>
</html>

```



Authentication

UserName

Password

Login

← → ↻

localhost:8084/user/index

Drive Cours UX design : découv... TOEIC LinkedIn engVid Gmail personnelle Classroom Open P-TECH Exercices corrigés s... See Intro to bevy &... 10 Motivational Qu... >>











Home Clients ▾

afaf ▾

Profile  
logout

liste des clients

keyword  🔍

Id	Nom	UserName	email	Abonnement		
7	afaf	afaUser	afaf@gmail.com	<a href="#">Abonnement ▾</a>		
8	saad	saadUser	saad@gmail.com	<a href="#">Abonnement ▾</a>		
9	ikram12	ikramUser12	ikram12@gmail.com	<a href="#">Abonnement ▾</a>		
10	imane	imaneUser	imane@gmail.com	<a href="#">Abonnement ▾</a>		
11	afaf	afaUser	afaf@gmail.com	<a href="#">Abonnement ▾</a>		

0

1

2

← → ↻

localhost:8084/user/index

Drive Cours UX design : découv... TOEIC LinkedIn engVid Gmail personnelle Classroom Open P-TECH Exercices corrigés s... See Intro to bevy &... 10 Motivational Qu... >>

Home Clients ▾

ikram ▾

liste des clients

keyword  🔍

Id	Nom	UserName	email	Abonnement
7	afaf	afaUser	afaf@gmail.com	<a href="#">Abonnement ▾</a>
8	saad	saadUser	saad@gmail.com	<a href="#">Abonnement ▾</a>
9	ikram12	ikramUser12	ikram12@gmail.com	<a href="#">Abonnement ▾</a>
10	imane	imaneUser	imane@gmail.com	<a href="#">Abonnement ▾</a>
11	afaf	afaUser	afaf@gmail.com	<a href="#">Abonnement ▾</a>

0

1

2

← → ↻

localhost:8084/user/abonnement?id=7

Drive Cours UX design : découv... TOEIC LinkedIn engVid Gmail personnelle Classroom Open P-TECH Exercices corrigés s... See Intro to bevy &... 10 Motivational Qu... >>

Home Clients ▾

ikram ▾

liste des Abonnements

Id	Type	Date Abonnement	Solde	Montant Mensuel	Client Id
68	GSM	2023-05-04	80.4	100.0	7