

# TD n°2

Notions abordées :

- Models/Entities
- JPA + Hibernate
- H2
- Persistence

## Création

Créer le projet td2 :

- Group : s4.spring
- Artifact : td2
- Packaging : War
- Description : Gestion de messagerie
- Dependencies : Web, Devtools, Mustache, H2, JPA

Configurer le projet dans **application.properties**, pour que le contextPath soit /, configurer Mustache, H2

```
spring.datasource.url=jdbc:h2:file:./data/messagerie;DB_CLOSE_ON_EXIT=FALSE
spring.datasource.username=sa
spring.datasource.password=
spring.datasource.driverClassName=org.h2.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true
spring.h2.console.path=/h2-console

server.servlet.context-path=/

# Mustache Template engine
spring.mustache.prefix=classpath:/templates/
spring.mustache.suffix=.html
```

Ajouter webjars à pom.xml pour intégrer Semantic-UI :

```
<!-- https://mvnrepository.com/artifact/org.webjars/Semantic-UI -->
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>Semantic-UI</artifactId>
  <version>2.4.1</version>
</dependency>
```

Ajouter la dépendance pour jQuery (indispensable avec Semantic)

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>jquery</artifactId>
  <version>3.3.1-1</version>
</dependency>
```

Créer une classe de configuration de l'application pour définir des mappings vers les webJars :

```
@Configuration
@EnableWebMvc
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry
            .addResourceHandler("/webjars/**")
            .addResourceLocations("/webjars/");
    }
}
```

Créer les templates :

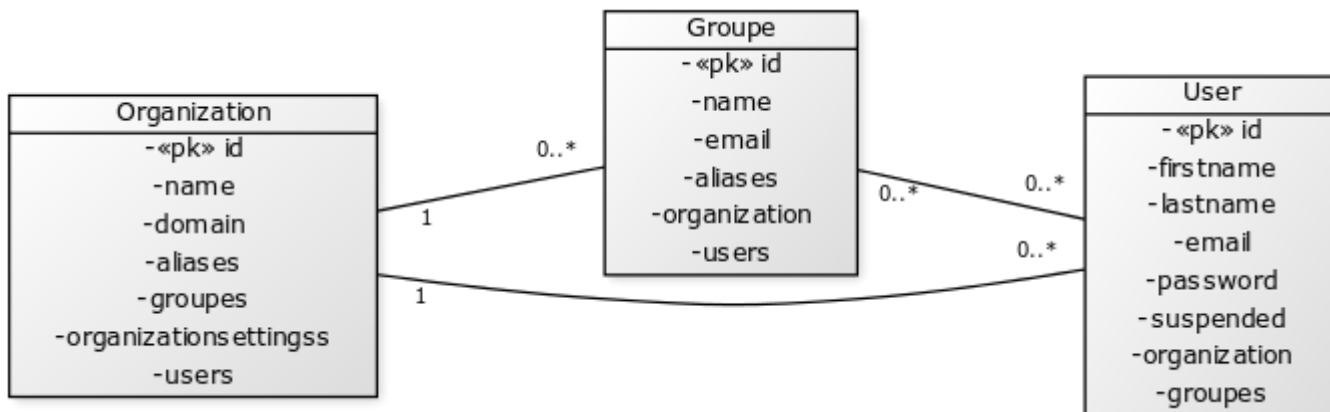
```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Messagerie</title>
  <link rel="stylesheet"
    href="/webjars/Semantic-UI/2.4.1/semantic.min.css" />
</head>
<body>
<div class="ui container">
```

```
</div>
<script src="/webjars/jquery/3.3.1-1/jquery.min.js"></script>
<script src="/webjars/Semantic-UI/2.4.1/semantic.min.js"></script>
</body>
</html>
```

## Création des Entities

Créer les classes métier correspondant au diagramme de classes suivant dans **s4.spring.td2.entities** :

- penser à ajouter les relations :
  - oneToMany
  - manyToOne
  - manyToMany

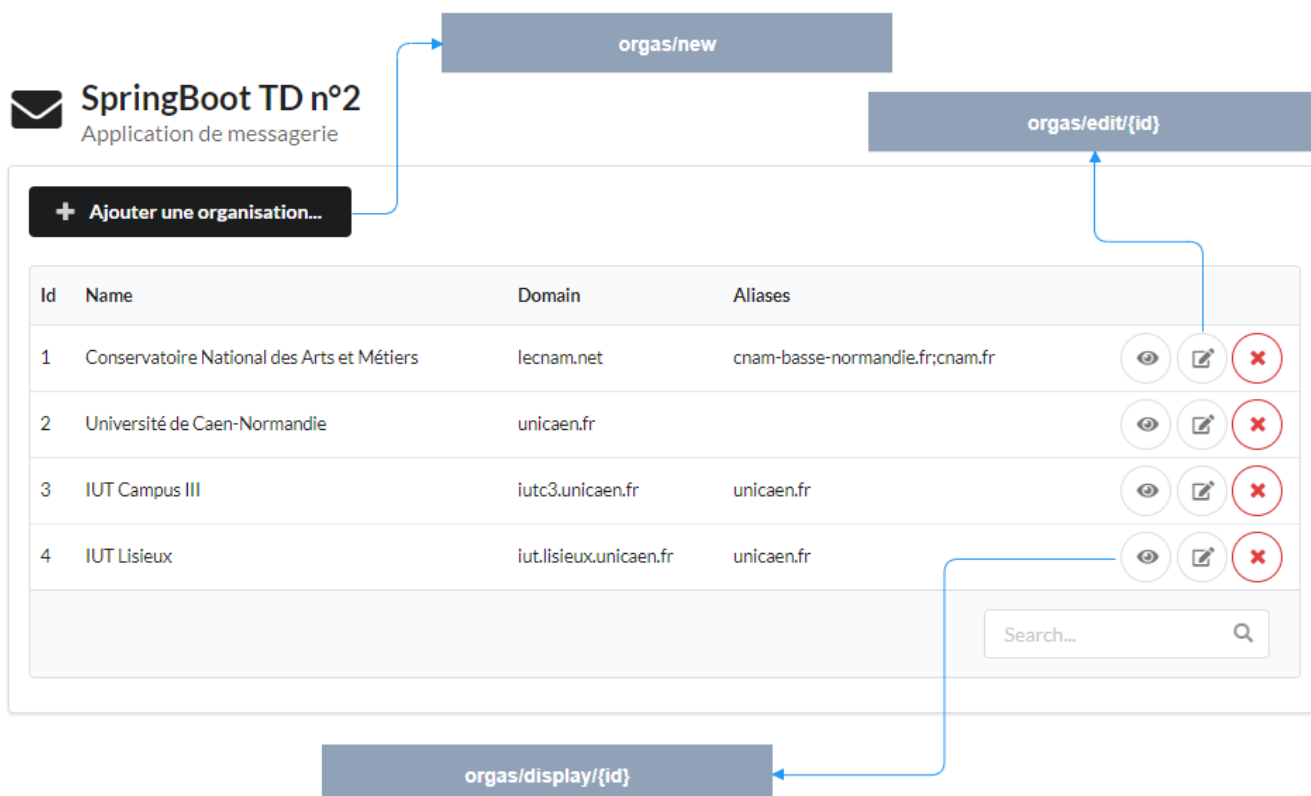


## Création des Repositories

Créer les repositories correspondant à chacune des classes dans `s4.spring.td2.repositories`.

## Création des routes

`/orgas/(index)?`



`/orgas/new`


**models\Organization**  
New object creation

Name

Domain

Aliases

### /orgas/display/{id}

 **SpringBoot TD n°2**  
Application de messagerie

**orgas/edit/{id}**

<b>Id</b>	2
<b>Name</b>	Université de Caen-Normandie
<b>Domain</b>	unicaen.fr
<b>Aliases</b>	
<b>Groupes</b>	Personnels Etudiants Enseignants Vacataires
<b>Organizationsettings</b>	: {{firstname}},{{lastname}}
<b>Users</b>	benjamin.sherman@gmail.com acton.carrillo@gmail.com zorita.rodriquez henry.beasley kelsey.weber carolyn.pace levi.bishop lionel.mccray jeremy.bryan ava.pollard ...

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:  
<http://slamwiki2.kobject.net/framework-web/spring/td2?rev=1549589541>

Last update: **2019/08/31 14:43**

