# TD n°3

### SpringBoot - VueJS



### Notions abordées

- Back-end :
    1. Service Rest
    2. Génération SPA

1. Front-end :
    - SPA vueJS
    - Vuetify

## Application messagerie (suite)

Rester sur le même projet :

- Ajouter la dépendance **Maven SpringBoot-VueJS**

```
<dependency>
    <groupId>io.github.jeemv.springboot.vuejs</groupId>
    <artifactId>springboot-vuejs</artifactId>
    <version>[1.0,)</version>
</dependency>
```

Ajouter VueJS et Vuetify dans les templates du projet (Webjars déconseillés) :

```
<!doctype html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Messagerie</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/vuetify/1.5.3/vuetify.min.css" />
    <link
href='https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900|Materi
al+Icons' rel="stylesheet">
</head>
<body>
```

```
<v-app>
  <v-content>
    <v-container fluid>
```

```
        </v-container>
    </v-content>
</v-app>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.18.0/axios.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.7/vue.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/vuetify/1.5.3/vuetify.min.js"></script>
</body>
</html>
```

# Service REST

Contrôleur **RestOrgasController** :

| **Action** | **methode** | **datas** | **Route** |
|---|---|---|---|
| read() | GET | | /rest/orgas/ |
| read(id) | GET | | /rest/orgas/{id} |
| create()* | POST | orga | /rest/orgas/create |
| update()* | PUT | orga | /rest/orgas/update |
| delete()* | DELETE | orga | /rest/orgas/delete |
| get(id,member) | GET | | /rest/orgas/{member}/{id} |

N'implémenter dans un premier temps que les méthodes marquées d'une étoile*.

# Application

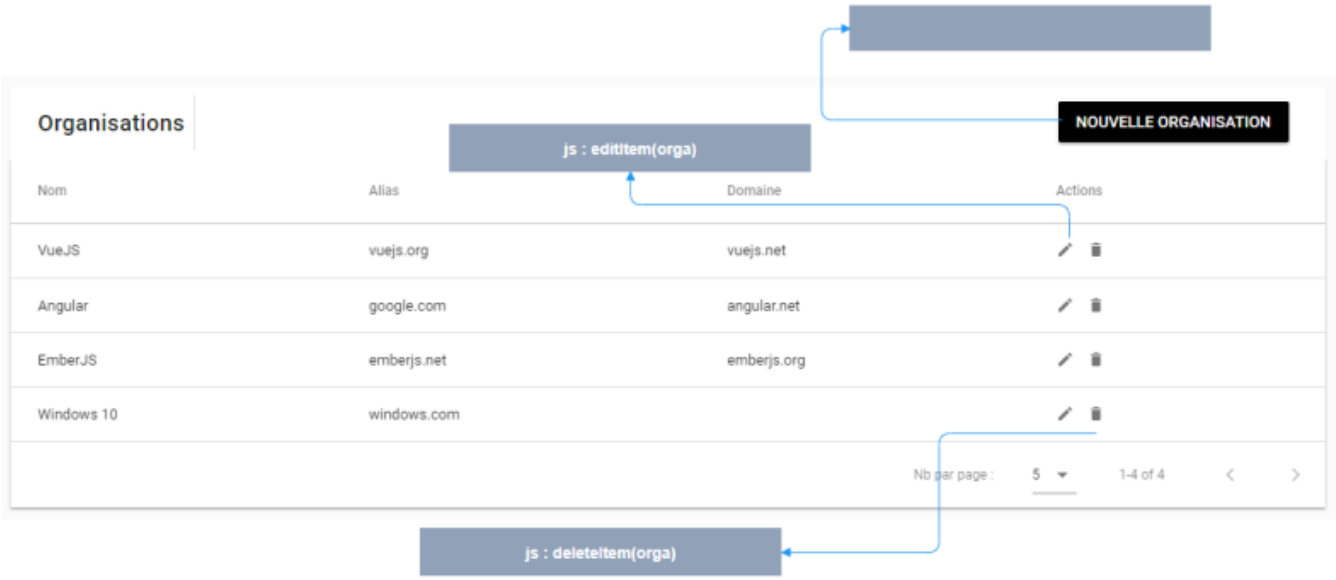
## Back-end

Contrôleur **OrgasController** :

La route **/orgas/** doit afficher les organisations et charger le module (SPA) de gestion des organisations.

## Front-end, généré en back

On utilisera le composant Data-table de Vuetify, exemple destiné au CRUD, pour obtenir le résultat suivant :







## SPA

| | | **Defaut** |
|---|---|---|
| **data** | orgas<br>Liste des organisations | Arraylist |
| | headers<br>En-têtes du data-table | Array JSON |
| | dialog<br>booléen pour affichage dialogue | false |
| | editedItem<br>orga sélectionnée | {} |
| | editedIndex<br>index de l'orga sélectionnée | -1 |
| **computed** | formTitle<br>Titre du dialogue | Varie en fonction de **editedIndex** |

| | | **Description** |
|---|---|---|
| **method** | close()<br>Ferme le dialogue | Réinitialise editedItem et editedIndex |
| | save()<br>Valide l'ajout ou la modification | POST/PUT vers REST + maj JS si succès |
| | deleteItem(orga)<br>Supprime l'organisation | DELETE vers REST + maj JS si succès |
| | editItem(orga)<br>Edite l'orga | maj JS + affichage dialogue |

## SPA (suite)

Sur sélection d'une organisation : - Afficher la liste des utilisateurs dans un composant List

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

Permanent link:
**http://slamwiki2.kobject.net/framework-web/spring/td3?rev=1551410167**

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