



Davy BALMINE

Paris, 14 juin 2018

Titre Professionnel niveau III “Développeur web”

Présentation

Reconversion professionnelle en alternance (9 mois)

Formation Développeur web

Stage en entreprise : La Poste



Application web de mise en relation

Projet réalisé par Davy BALMINE depuis avril 2018

Sommaire

Introduction

Cahier des charges

Organisation

Conception de l'application

Mise en place de la **base de données**

Développement du **Backend**

Développement du **Frontend**

Recette et **Livraison**

Conclusion

Introduction

Simplon'click est une **application de mise en relation** dont le but est de :

- ⇒ **Monter en compétence** (définir ses **besoins**, apprendre en enseignant ([teaching by teaching](#)))
- ⇒ **Favoriser le travail collaboratif** (partager ses connaissances)

Membre = Utilisateur (Membre de la communauté Simplon)

Savoir = Regroupe les notions de compétences, de langages / concepts informatiques...

Passeur = Le membre qui apporte un enseignement (donneur de savoir)

Receveur = Le membre qui exprime le besoin d'approfondir ses connaissances (receveur de savoir)

Ressource = Lien hypertexte vers une source du web

Application web : Java (Spring Boot) + Angular V

Cahier des charges

Application de **mise en relation**

- ⇒ Chaque membre est décrit principalement par ses **besoins** (receveur), ses **spécialités** (passeur), sa **disponibilité** et son **lieu de présence**.
- ⇒ Chaque besoin y est **détaillé** et laisse place à la notion de “savoir”.
- ⇒ Chaque savoir peut prétendre à des **ressources libres** (lien hypertexte).

Un **membre** peut :

- ⇒ **Consulter**, **ajouter**, **modifier** ou **supprimer** : *son compte, ses besoins, ses spécialités, un savoir ou une ressource*

Lien automatique entre les membres et les savoirs

Interface utilisateur simple et conviviale - Responsive

Organisation

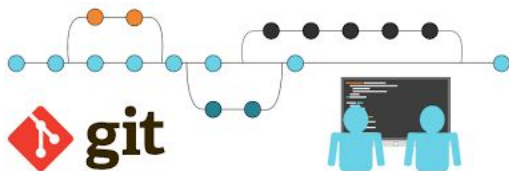
Organisation des tâches



Communication instantanée



Versionning



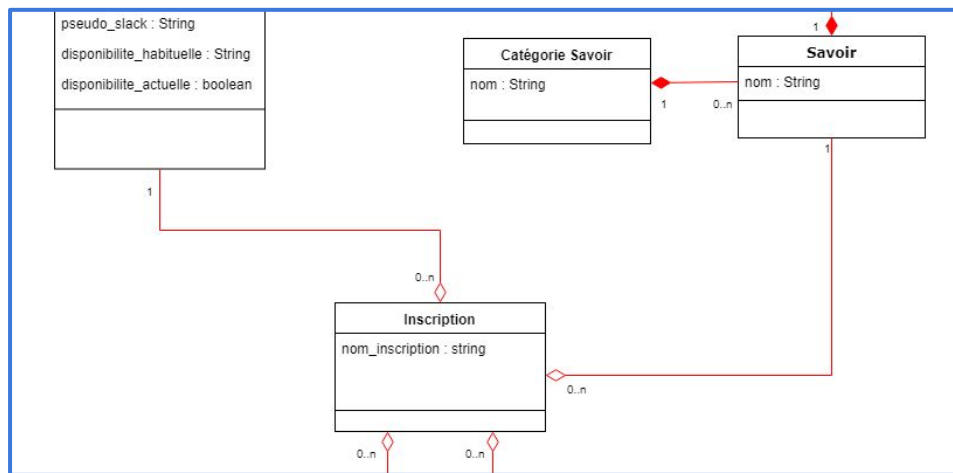
Dépôt en ligne



Conception (Diagrammes UML)

Diagrammes : Cas d'utilisation ➤ Séquence ➤ Classes

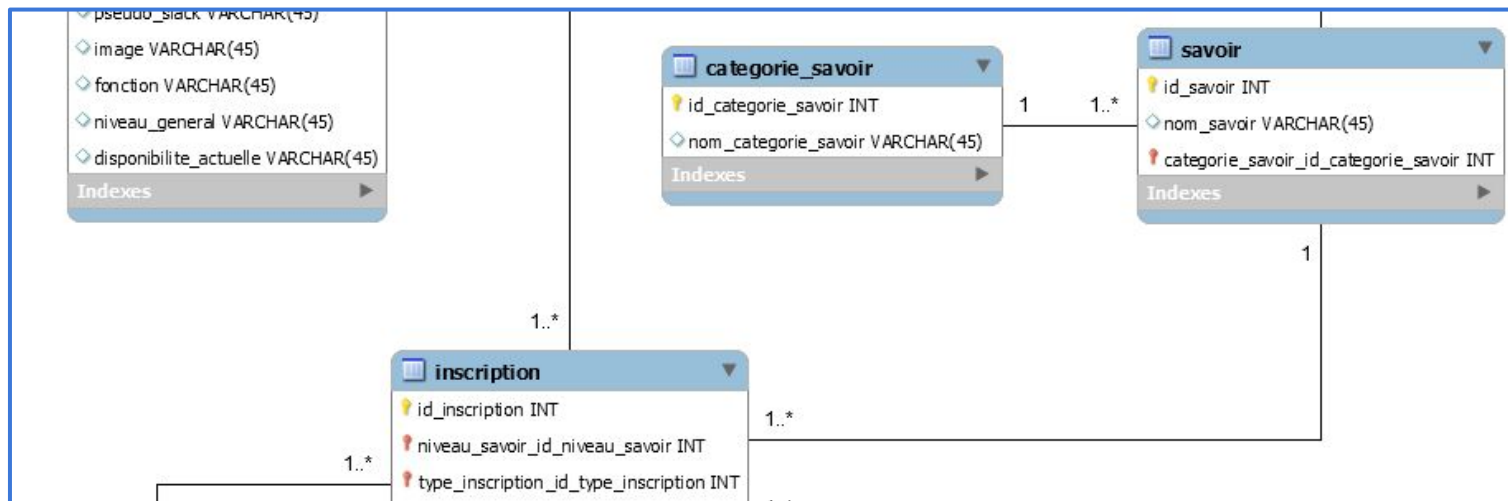
Diagramme des classes



Base de données (MySQL)



Base de données relationnelle



Base de données (MySQL)

Extrait du dump de la base

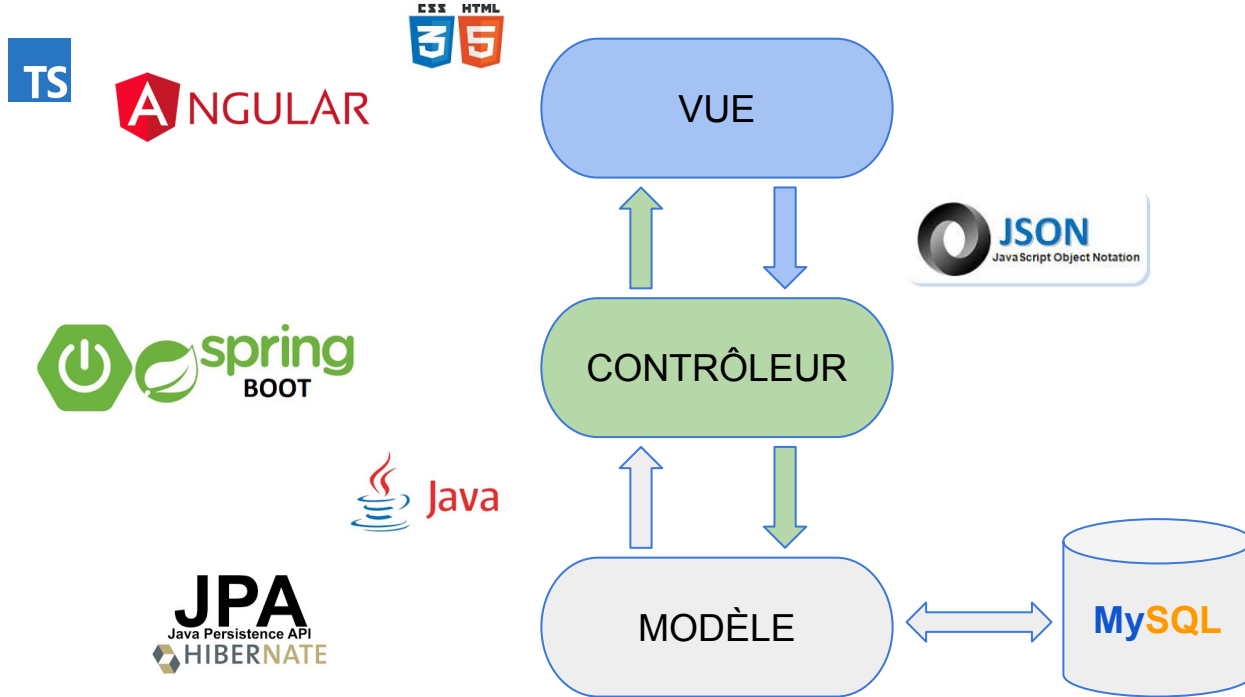
```
DROP TABLE IF EXISTS `membre`;
CREATE TABLE `membre` (
  `id_membre` bigint(20) NOT NULL AUTO_INCREMENT,
  `admin` bit(1) NOT NULL,
  `disponibilite_actuelle` bit(1) NOT NULL,
  `disponibilite_habituelle` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `fonction` varchar(255) DEFAULT NULL,
  `image` varchar(255) DEFAULT NULL,
  `niveau_general` varchar(255) DEFAULT NULL,
  `nom` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `prenom` varchar(255) DEFAULT NULL,
  `pseudo` varchar(255) DEFAULT NULL,
  `pseudo_slack` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id_membre`),
  UNIQUE KEY `UKjuh4as83jp2nka9svt1umenyv` (`nom`,`prenom`,`email`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

LOCK TABLES `membre` WRITE;
/*!40000 ALTER TABLE `membre` DISABLE KEYS */;
INSERT INTO `membre` VALUES (1, TRUE, TRUE, 'Le lundi à partir de 18 heures', 'gafa@hotmail.fr', 'apprenant',
  'https://i.redd.it/7if8lz40vo5z.jpg', 'Novice', 'Gallan', 'TEST', 'Fabien', 'Gafa', 'Gafan');
/*!40000 ALTER TABLE `membre` ENABLE KEYS */;
UNLOCK TABLES;
```

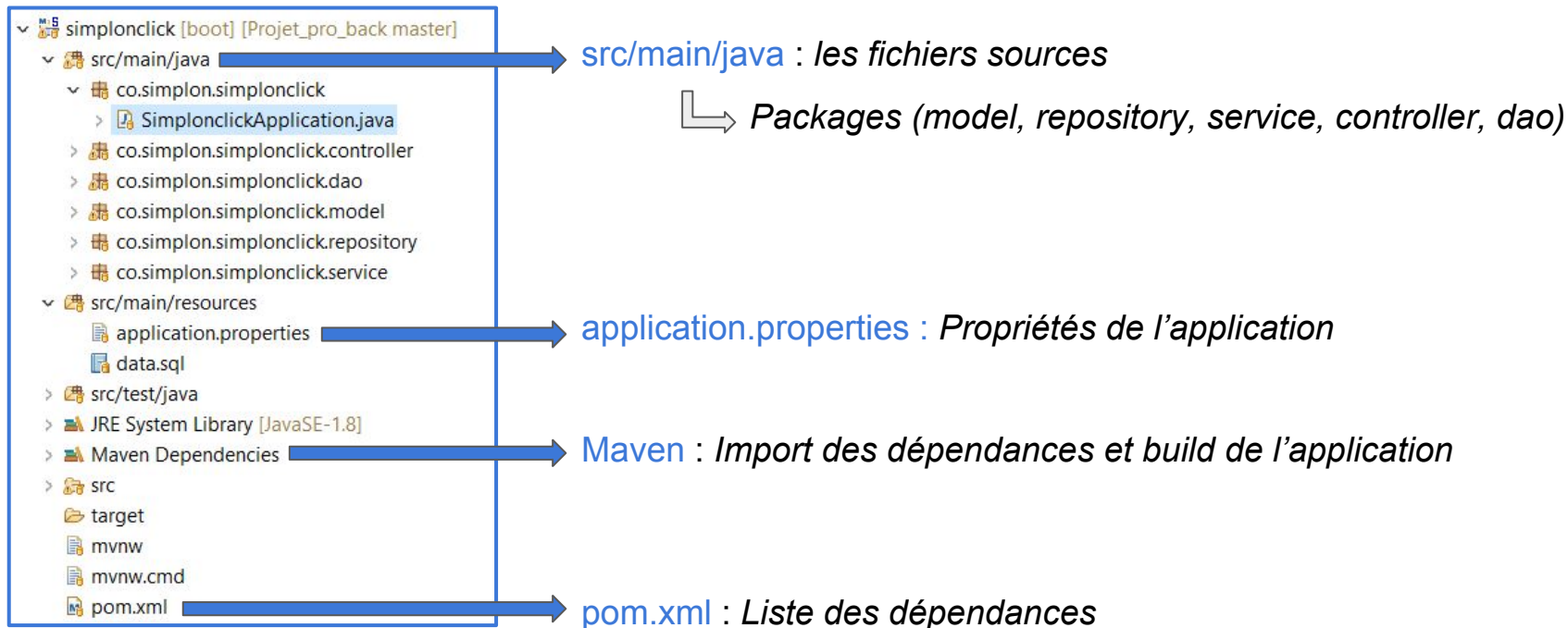
Requête SQL

```
SELECT membre.* FROM membre WHERE id_membre IN
(SELECT membre_id_membre FROM inscription WHERE savoir_id_savoir = ? AND type_inscription_id_type_inscription = 2 AND niveau_savoir_id_niveau_savoir = 3);
```

Architecture (REST & MVC)

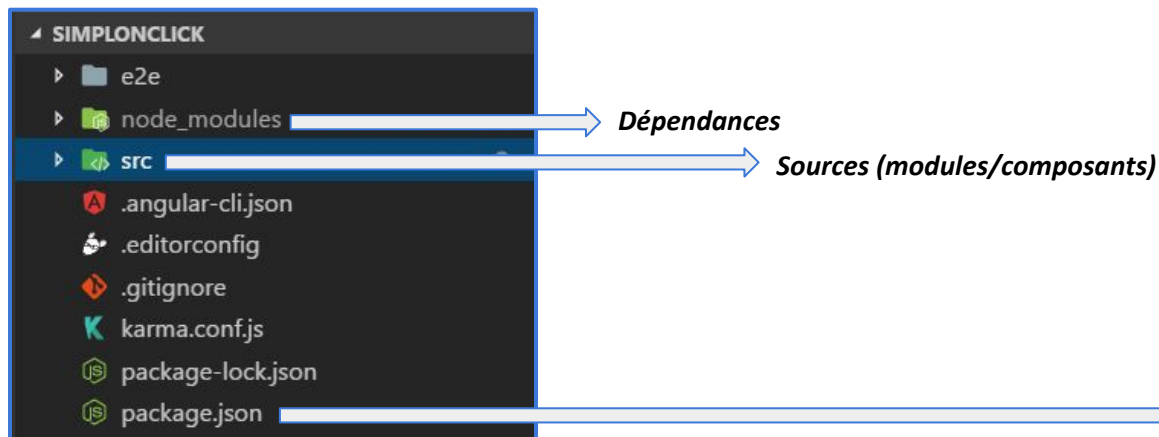


Backend (Spring Boot)



Frontend (Angular & Angular Material)

DOM



Configuration

```
1 {
2   "name": "simplonclick",
3   "version": "0.0.0",
4   "license": "MIT",
5   "scripts": {
6     "ng": "ng",
7     "start": "ng serve",
8     "build": "ng build --prod",
9     "test": "ng test",
10    "lint": "ng lint",
11    "e2e": "ng e2e"
12  },
13  "private": true,
14  "dependencies": {
15    "@angular/animations": "^5.2.9",
16    "@angular/cdk": "^5.2.4",
```

Recette & Livraison

Ateliers utilisateurs

Prérequis

⇒ Backend (*modification du fichier application.properties*) :


- ❖ Java 1,8
- ❖ Maven (+ commandes exécuter/déployer)
- ❖ SGBDR MySQL
- ❖ schema.sql (création de la base et des tables)
- ❖ data.sql (obtention des données)

⇒ Frontend :

- ❖ NodeJS

Lancement via Github (*ou fichier .jar pour le backend*)

Conclusion

 **Application** REST **complète** avec base de données (*réalisation de la conception à la livraison*)
Expérience très enrichissante

Évolutions :

- ⇒ *Forum intégré*
- ⇒ *Signalement par mail (chaque nouvelle inscription)*
- ⇒ *Restriction des habilitations*



MERCI POUR VOTRE ATTENTION

Davy BALMINE

Paris, 14 juin 2018

Titre Professionnel niveau III "Développeur web"