



Première mini application Backend avec JAVA

Vous allez créer votre première application Java qui vous permettra de visualiser et de créer facilement des posts.

- Java 8
- Spring Boot,
- PostgreSQL,
- Maven
- * JUnit
- * Plugin Sonar

Référentiels :

Développer la partie backend d'une application :

- Application des principes de programmation orientée objet
- Développement des fonctionnalités de l'application, dans un style défensif, et éventuellement en asynchrone.
- Utilisation des composants d'accès aux données.
- Utilisation des composants tiers et en vérifier l'efficacité et la sécurité à l'aide d'un gestionnaire de dépendances.
- Utilisation de Framework et bibliothèques avancées, couramment utilisés dans le domaine.

Concevoir avec agilité :

- Gérer un projet de manière efficace et itérative.
- Pratiquer la méthodologie Scrum.
- Appliquer des formats de facilitation dans la conduite de projets et de réunions.
- S'organiser, individuellement ou en groupe.
- Communiquer clairement avec des collaborateurs dans le cadre d'un projet.
- S'adapter au changement et être flexible sur un projet soumis à changements fréquents.

Concevoir et optimiser une base de données :

- Réalisation et exécution d'un script de gestion de base de données à l'aide de l'environnement intégré de développement
- Sécurisation des accès à la base de données
- Programmer les scripts d'alimentation de la base
- Création des requêtes de recherches

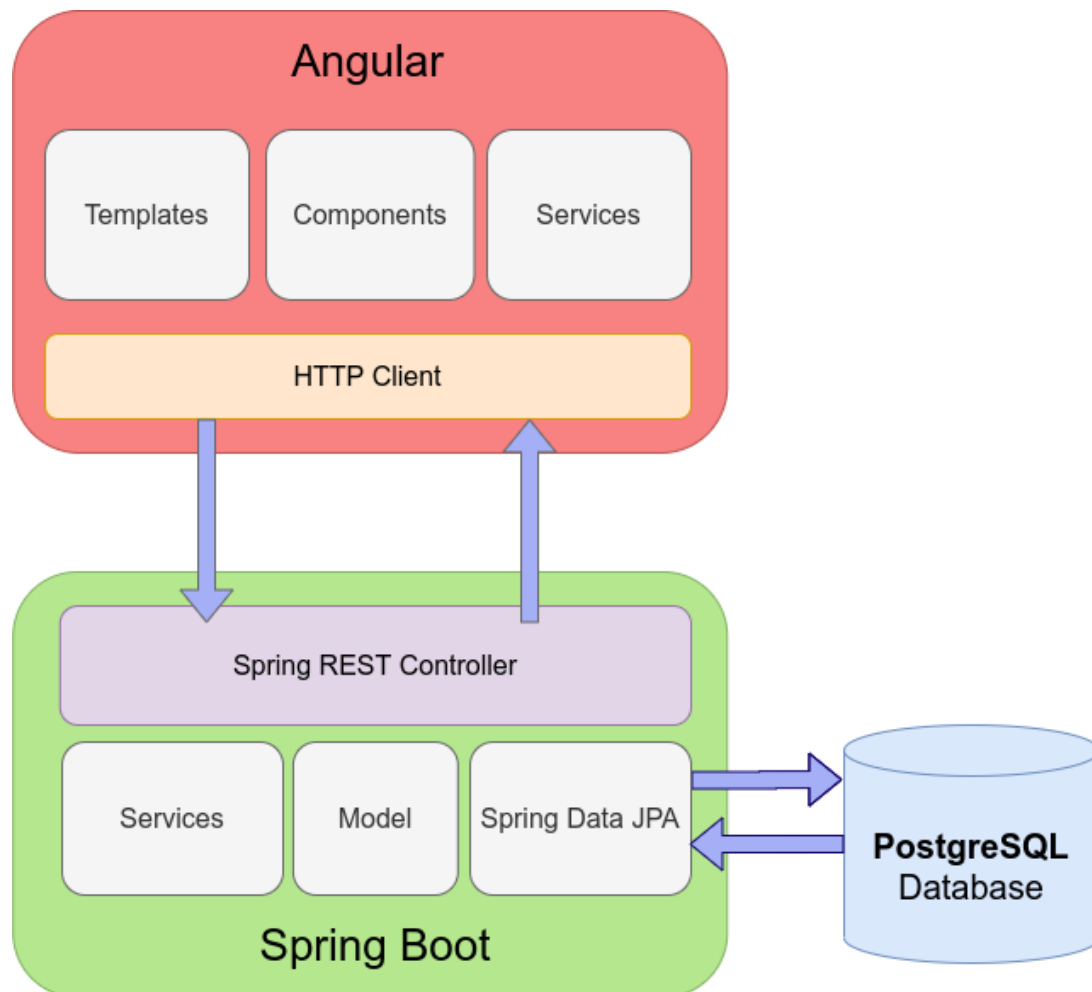
Activités et tâches transversales :

- Documentation du code
- Respect des règles orthographiques et grammaticales des documents produits en français ou en anglais
- Utilisation d'un environnement de développement intégré y compris en anglais
- Utilisation d'un outil collaboratif de partage de fichiers
- Planification et suivi des tâches de développement
- Recherche d'une réponse pertinente à une difficulté technique de développement ou à une vulnérabilité identifiée.
- Recherche des informations sur des sites Internet, des forums et des FAQ francophones ou anglophones
- Contribution à la mise à jour des bases de connaissances francophones ou anglophones accessibles par Internet

Contexte du projet :

Architecture

L'application sera construite à l'aide de l'architecture suivante :



Pour le Projet on peut distinguer trois couches principales :

- Front-end avec Angular 11 comme framework principal, avec des modèles, des composants et un client HTTP pour la communication avec le serveur d'applications REST-full ().
- Back-end - côté serveur avec Spring Boot, Spring REST Controllers et Spring Data Jpa pour la communication avec la base de données,
- Base de données - les données soumises seront stockées dans la base de données PostgreSQL.

Structure du projet :

```
├─ package.json
├─ package-lock.json
├─ README.md
├─ src
│   └─ app
│       ├── app.component.css
│       ├── app.component.html
│       ├── app.component.ts
│       ├── app.module.ts
│       ├── app-routing.module.ts
│       ├── components
│       │   ├── post-form
│       │   │   ├── post-form.component.html
│       │   │   └─ post-form.component.ts
│       │   └─ post-list
│       │       ├── post-list.component.html
│       │       └─ post-list.component.ts
│       ├── model
│       │   └─ post.model.ts
│       └─ services
│           └─ post.service.ts
├─ favicon.ico
├─ index.html
├─ main.ts
├─ polyfills.ts
└─ styles.css
```

Le projet Front-end en Angular 11 contient deux composants : **post-form** et **post-list**,

- **post.service** est responsable de la communication avec REST api,
- **post.model** est la représentation Post dans l'objet JavaScript (avec des champs comme le titre (title), le contenu (content), les balises (tags), l'identifiant (id)),
- **App.module** est un module principal angular où nous avons importé les modules comme **Forms** et **HttpClient**,
- **app.routing.module** contient URL composant angular correspondants.

Présentation de l'API REST

Le serveur d'applications Spring Boot fournira l'API avec les points de terminaison suivants :

URL	Method	Action
/posts	GET	Obtenir la liste de tous les messages créés
/posts?title={title}	GET	Obtenez une liste de messages filtrée par titre
/posts	POST	Créer un nouveau message
/posts/{id}	GET	Obtenir la publication par identifiant spécifique
/posts/{id}	DELETE	Supprimer le message par identifiant spécifique
/posts/{id}	PUT	Mettre à jour le message

Critères de performance :

Le Front-end vous est fourni en ressource.

Pour aller plus loin vous pouvez faire une veille sur les Test Unitaires au sein de votre appli

Modalités d'évaluation :

Restitution Lundi 25/10 au matin devant la promo en groupe.

Livrables :

Livré sur le dépôt Github (Pusher vos code Dimanche 24/10 à minuit au plus tard).