

Application Edster

Edster est une application basée sur JHipster, conçue pour créer et générer des présentations. Elle exploite la plateforme JHipster pour développer et déployer rapidement des applications modernes de génération de présentations.

Configuration de Edster

Génération du code source

Le code source d'Edster peut être généré à l'aide du script `jdl.sh`. Ce script traite un fichier JDL (JHipster Domain Language) pour définir les entités, les relations et les configurations de l'application.

Note : Avant d'exécuter le script `jdl.sh` pour la première fois, vous devez le rendre exécutable à l'aide de la commande suivante :

```
chmod u+x scripts/jdl.sh
```

Pour générer le code source :

1. Exécutez le script `jdl.sh` :

```
./scripts/jdl.sh
```

Cette commande va :

- Analyser le fichier JDL.
- Générer le code d'application nécessaire en fonction de la définition JDL.

Assurez-vous que le script `jdl.sh` est exécutable et que vous avez l'environnement JHipster nécessaire configuré.

Nettoyage du code généré

Si vous devez nettoyer le code généré, vous pouvez utiliser le script `clean.sh`.

Note : Avant d'exécuter le script `clean.sh` pour la première fois, vous devez le rendre exécutable à l'aide de la commande suivante :

```
chmod u+x scripts/clean.sh
```

Pour nettoyer le code généré :

1. Exécutez le script `clean.sh` :

```
./scripts/clean.sh
```

Cette commande supprimera le code généré, vous permettant de le régénérer à l'aide du script `jdl.sh`. == Execution de Edster === Mode Développement

Pour exécuter l'application en mode développement, suivez les étapes suivantes :

1. Démarrez l'environnement de développement :

```
npm run app:start;
```

Cette commande démarrera le serveur backend avec le rechargement à chaud activé. Le frontend est servi par le serveur de développement webpack.

1. Accédez à l'application dans votre navigateur :

<http://localhost:8080/>

2. Configurations spécifiques au développement :

- Profil : `dev`
- Rechargement à chaud pour les modifications du backend
- Serveur de développement Webpack pour le frontend
- Base de données H2 (si elle n'est pas configurée différemment)

Mode Production

Pour exécuter l'application en mode production, suivez les étapes suivantes :

1. Construisez le package prêt pour la production :

```
./gradlew -Pprod clean build
```

Cette commande va :

- Compiler le code Java
- Exécuter les tests
- Empaqueter l'application dans une archive déployable (`.war` ou `.jar` ou `.gradle`)
- Optimiser les ressources du frontend

1. Exécutez l'application :

```
java -jar build/libs/edster-0.0.1.jar
```

CAUTION

La version de edster-0.0.1.jar peut varier en fonction de la version de l'application.

Cette commande démarrera l'application en utilisant le profil de production.

1. Accédez à l'application dans votre navigateur :

<http://localhost:8080/>

2. Configurations spécifiques à la production :

- Profil : **prod**
- Ressources frontend optimisées et minifiées
- Pas de rechargement à chaud
- Base de données de production (par exemple, PostgreSQL, MySQL)

Itération de développement

Pendant le développement, vous pouvez exécuter le backend et le frontend séparément pour une itération plus rapide.

1. Pour exécuter uniquement le backend :

```
./npmw run backend:start;
```

Cette commande démarrera le serveur backend à :

+ <http://localhost:8080/>

1. Pour exécuter le frontend en mode live-reload :

```
npm run start;
```

Cette commande démarrera le serveur de développement frontend avec le rechargement à chaud, vous permettant de voir les modifications en temps réel à :

+ <http://localhost:9001/>

Synchronisation du code source

Le script **sync.sh** permet de synchroniser les modifications manuelles du code avec le code source généré par JDL. Cela garantit que les personnalisations sont conservées lors de la régénération de l'application à l'aide de **jdl.sh**.

Note : Avant d'exécuter le script `sync.sh` pour la première fois, vous devez le rendre exécutable à l'aide de la commande suivante :

```
chmod u+x scripts/sync.sh
```

Pour synchroniser le code source :

1. Exécutez le script `sync.sh` :

```
./scripts/sync.sh
```

Ce script identifiera et fusionnera vos modifications manuelles dans le code nouvellement généré, minimisant les conflits et préservant vos personnalisations.

Ajouter une entrée au fichier `.gitignore`

Ajouter le dossier `.goose` au fichier `.gitignore`

Afin d'ajouter le dossier `.goose` au fichier `.gitignore`, vous devez ajouter ce code à la fin du fichier `jdkl.sh`.

```
echo ".goose" >> .gitignore;
```

Exécution séparée de l'application front-end et back-end en mode développement sur un seul terminal

Pour exécuter séparément l'application front-end et back-end en mode développement sur un seul terminal, vous pouvez suivre les étapes suivantes :

```
./npmw run backend:start & npm run start;
```

Un seul Ctrl+C arrêtera uniquement le front-end. Pour arrêter le back-end, vous devrez trouver l'identifiant du processus et le tuer, comme ceci :

```
killall -9 java
```

Pour relancer le front-end :

```
npm run start;
```

Pousser Edster sur dockerhub

blabla