

Interface Graphique de Contrôle Optimal

1. Description:

Une interface web pour résoudre des problèmes de contrôle optimal en utilisant une application backend et une interface utilisateur frontend.

2. Installation et Configuration

○ Prérequis

- Node.js (version 14 ou supérieure)
- NPM (version 6 ou supérieure)

○ Installation

- Clonez le dépôt :

```
git clone https://github.com/youssef1969/Composant-Editeur
```

- Naviguez dans le répertoire du projet :

```
cd front
```

- Installez les dépendances :

```
npm install
```

○ Démarrage de l'application

- Pour démarrer le serveur de développement :

```
npm start
```

- L'application sera disponible sur `http://localhost:3000`.

3. Structure du Projet

○ Backend

- **Contrôleur** : `OptimalControlController.cs`

- Endpoint GET `/api/OptimalControl` pour vérifier la connexion.
- Endpoint POST `/api/OptimalControl` pour recevoir les données du frontend et retourner une solution.

- **Données** : `ProblemData.cs`

- Classe définissant la structure des données envoyées du frontend.

○ Frontend

- **Composant principal** : `App.js`

- Utilise `axios` pour la communication avec le backend.
- Utilise `CodeMirror` pour l'édition du code.
- Contient la logique d'affichage de l'interface utilisateur et de soumission des données.

- **Styles** : `App.css`

- Contient les styles CSS pour l'interface utilisateur.

4. Utilisation

- **Interface Utilisateur**

- L'application affiche un message de bienvenue récupéré depuis le backend.
- Les users peuvent saisir des paramètres et une description de problème dans des éditeurs de code.
- Les users peuvent cliquer sur un bouton pour soumettre leurs données.
- Une solution est affichée après soumission.

- **Fonctionnalités Avancées**

- Afficher/masquer la description du problème à l'aide d'un bouton toggle.
- Les résultats sont affichés avec une taille de police plus petite pour une meilleure lisibilité.

5. Fonctionnalités

- **Éditeurs de Code**

- Utilisation de `CodeMirror` pour offrir une expérience d'édition riche.

- **Toggle Description**

- Utilisation de `FontAwesome` pour afficher une icône plus/moins et permettre à l'utilisateur d'afficher ou de masquer la section de description du problème.

- **Soumission des Données**

- Envoi des données saisies au backend pour traitement et affichage des résultats.

6. Dépendances

- **Backend**

- ASP.NET Core

- **Frontend**

- React
- Axios
- CodeMirror
- FontAwesome