

Fichier : 06 - services_llm_chatgpt.service.js

Tutoriel : Creation d'une fonction de reponse avec Node.js et l'API OpenAI

Dans ce tutoriel, nous allons examiner une fonction `reply` qui utilise l'API OpenAI pour generer des reponses en fonction de differents styles et longueurs de reponses.

Code complet

Voici le code complet que nous allons decomposer et expliquer :

```
import axios from 'axios';

const styleMap = {
  //...
};

const lengthMap = {
  //...
};

async function reply(type, input) {
  //...
}

export default reply;
```

Importation de la bibliotheque Axios

```
import axios from 'axios';
```

Nous importons la bibliotheque `axios` pour effectuer des requetes HTTP. Axios est une bibliotheque populaire qui offre une API facile a utiliser pour effectuer des requetes HTTP.

Definition des mappages de style et de longueur

```
const styleMap = {
  //...
};

const lengthMap = {
  //...
};
```

Ces deux objets mappent des chaines de caracteres a des descriptions de styles et de longueurs de reponses. `styleMap` definit differents styles de reponses, tandis que `lengthMap` definit differentes longueurs de reponses.

Fonction de reponse

```

async function reply(type, input) {
  //...
}

```

C'est la fonction principale qui genere la reponse. Elle est asynchrone car elle effectue une requete HTTP, qui est une operation asynchrone.

Preparation de la requete

```

const name = input.name || 'inconnu';
const rawStyle = input.style || 'neutral';
const rawLength = input.length || 'medium';

const style = styleMap[rawStyle] || styleMap.neutral;
const length = lengthMap[rawLength] || lengthMap.medium;

const prompt = type === 'summary'
  ? `Fais un resume du film "${name}" avec un style ${style}, ${length}.`
  : `Ecris une biographie de ${name} avec un style ${style}, ${length}.`;

```

Dans cette section, nous preparons les donnees pour la requete. Nous utilisons les valeurs par default si certaines donnees ne sont pas fournies.

Envoi de la requete

```

const response = await axios.post(
  'https://api.openai.com/v1/chat/completions',
  {
    model: 'gpt-4-turbo',
    messages: [{ role: 'user', content: prompt }],
  },
  {
    headers: {
      Authorization: `Bearer ${process.env.OPENAI_API_KEY}`,
      'Content-Type': 'application/json',
    },
  },
);

```

Ici, nous envoyons une requete POST a l'API OpenAI. Nous utilisons le modele `gpt-4-turbo` et nous passons notre message dans le corps de la requete.

Gestion des erreurs

```

try {
  //...
} catch (error) {
  //...
}

```

Nous utilisons un bloc `try/catch` pour gerer les erreurs qui peuvent se produire lors de l'envoi de la requete.

Si une erreur se produit, nous la consignons et la renvoyons pour qu'elle puisse être traitée par le code appelant.

Exportation de la fonction de réponse

```
export default reply;
```

Enfin, nous exportons la fonction `reply` pour qu'elle puisse être utilisée dans d'autres modules.