

Fichier : 07 - services_llm_claude.service.js

Tutoriel : Utilisation de l'API Claude avec Node.js

Ce tutoriel vous guide à travers un exemple de code Node.js qui utilise l'API Claude pour générer des réponses stylisées et de longueur variable.

Code complet

Voici le code complet que nous allons décomposer et expliquer dans ce tutoriel.

```
import axios from 'axios';

const styleMap = {
  //...
};

const lengthMap = {
  //...
};

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

export default reply;
```

Importation de la bibliothèque Axios

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

Nous commençons par importer la bibliothèque Axios, qui est une bibliothèque JavaScript promise-based utilisée pour faire des requêtes HTTP.

Cartes de style et de longueur

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

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

Ces deux objets sont des cartes de style et de longueur. Ils associent des identifiants de style et de longueur à des descriptions textuelles. Ces descriptions seront utilisées pour construire l'invite que nous allons envoyer à l'API Claude.

Fonction de réponse

```

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

```

C'est ici que la magie opère. La fonction `reply` est une fonction asynchrone qui prend deux arguments : `type` et `input`. `type` peut être soit 'summary' soit autre chose, et `input` est un objet qui doit contenir les clés `name`, `style` et `length`.

Construction de l'invite

```

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 construisons l'invite à envoyer à l'API Claude. Nous utilisons les valeurs de `name`, `style` et `length` fournies dans l'objet `input`, ou des valeurs par défaut si elles ne sont pas fournies.

Envoi de la requête à l'API Claude

```

const response = await axios.post(
  'https://api.anthropic.com/v1/messages',
  {
    model: 'claude-3-5-sonnet-20240620',
    max_tokens: 1000,
    messages: [{ role: 'user', content: prompt }],
  },
  {
    headers: {
      'x-api-key': process.env.ANTHROPIC_API_KEY,
      'anthropic-version': '2023-06-01',
      'Content-Type': 'application/json',
    },
  },
);

```

Nous utilisons Axios pour envoyer une requête POST à l'API Claude. Nous passons l'invite que nous avons construite dans la section précédente dans le corps de la requête.

Gestion des erreurs

```

const result = response.data.content?.[0]?.text;
if (!result) { throw new Error('Reponse vide de Claude.')}

return result;

```

Enfin, nous vérifions si nous avons reçu une réponse de l'API Claude. Si ce n'est pas le cas, nous lançons une erreur. Si nous avons reçu une réponse, nous la renvoyons.

Exportation de la fonction de réponse

```
export default reply;
```

Nous exportons la fonction `reply` pour qu'elle puisse être utilisée dans d'autres parties de notre application.