

Fichier : 03 - mock_llm_chatgpt.mock.js

Tutoriel : Comprendre la fonction `reply` en Node.js

Dans ce tutoriel, nous allons decomposer une fonction Node.js appelee `reply`. Cette fonction est utilisee pour generer une reponse formatee en fonction des parametres recus.

Code

Voici le code de la fonction `reply` :

```
function reply(type, data) {
  const name = (data.name || 'Inconnu').replace('-', ' ');
  const style = data.style || 'neutral';
  const length = data.length || 'medium';
  const llm = data.llm || 'chatgpt';
  const validType = ['biography', 'filmography', 'summary'].includes(type) ? type : 'contenu';

  return `Mock Backend - Demande envoyee a ${llm} pour une ${validType} de "${name}", avec un
  style "${style}" et une longueur "${length}".`;
}

export default reply;
```

Explication

Initialisation des variables

La fonction `reply` prend deux parametres : `type` et `data`.

```
const name = (data.name || 'Inconnu').replace('-', ' ');
const style = data.style || 'neutral';
const length = data.length || 'medium';
const llm = data.llm || 'chatgpt';
```

Dans ce bloc de code, nous initialisons quatre variables (`name`, `style`, `length`, `llm`) avec les valeurs correspondantes de l'objet `data`. Si une valeur n'est pas fournie, nous utilisons une valeur par default.

Validation du type

```
const validType = ['biography', 'filmography', 'summary'].includes(type) ? type : 'contenu';
```

Ici, nous verifions si le `type` fourni est present dans notre tableau de types valides. Si c'est le cas, nous utilisons le `type` fourni, sinon nous utilisons `contenu` comme valeur par default.

Construction de la reponse

```
return `Mock Backend - Demande envoyee a ${llm} pour une ${validType} de "${name}", avec un style
```

```
"${style}" et une longueur "${length}".`;
```

Enfin, nous construisons et renvoyons une chaîne de caractères formatée qui contient toutes nos variables.

Conclusion

La fonction ``reply`` est un exemple simple de comment nous pouvons utiliser les paramètres et les valeurs par défaut pour générer une réponse dynamique en Node.js. En comprenant comment cette fonction fonctionne, vous pouvez créer des fonctions similaires qui répondent à vos propres besoins.