

Fichier : 02 - config_ai-services.js

Tutoriel : Comprendre le code Node.js pour les services d'IA

Ce tutoriel va vous aider a comprendre le code Node.js utilise pour definir une collection de services d'IA. Nous allons examiner le code, puis expliquer les concepts cles et les roles des differentes sections du code.

Code

```
const aiServices = {
  llm: [
    { type: 'chatgpt', label: 'OpenAI', purpose: 'Text generation, summarization, Q&A, code completion' },
    { type: 'claude', label: 'Claude', purpose: 'Structured reasoning, content writing, safe dialogue' },
    { type: 'gemini', label: 'Gemini', purpose: 'Multimodal LLM for text and image understanding' },
    { type: 'mistral', label: 'Mistral', purpose: 'Open-source LLM for high-performance text/code tasks' },
    { type: 'perplexity', label: 'Perplexity AI', purpose: 'Web-augmented search engine powered by LLM' },
    { type: 'deepseek', label: 'DeepSeek', purpose: 'Code generation, explanation and debugging assistant' },
  ],

  tts: [
    { type: 'elevenlabs', label: 'ElevenLabs', purpose: 'High-quality voice synthesis from text, multilingual' },
  ],

  avatar: [
    { type: 'did', label: 'D-ID', purpose: 'Animate a still photo with audio or text' },
    { type: 'heygen', label: 'Heygen', purpose: 'Generate talking avatar videos from script' },
    { type: 'jogg', label: 'Jogg AI', purpose: 'Create realistic talking avatars from custom photos' },
  ],

  image: [
    { type: 'leonardo', label: 'Leonardo AI', purpose: 'Create illustrations, concept art and product visuals' },
    { type: 'midjourney', label: 'MidJourney', purpose: 'Stylized artistic image generation from prompt' },
    { type: 'kling', label: 'Kling AI', purpose: 'Future video generation from text (Sora-level quality)' },
  ],

  agent: [
    { type: 'langchain', label: 'LangChain', purpose: 'Chain tools, memory, and LLMs into intelligent agents' },
    { type: 'llamaindex', label: 'LlamaIndex', purpose: 'Connect LLMs to data sources, documents,
```

```

and files' }},
  ],

  music: [
    { type: 'suno', label: 'Suno AI', purpose: 'Generate full songs with lyrics, melody, and vocals' },
    { type: 'udio', label: 'Udio AI', purpose: 'Generate high-quality vocal music tracks from prompt' },
  ],
};

export default aiServices;

```

Explication du code

Structure de base

Le code definit une constante `aiServices` qui est un objet JavaScript. Cet objet contient plusieurs proprietes qui correspondent a differentes categories de services d'IA : `llm`, `tts`, `avatar`, `image`, `agent` et `music`. Chaque categorie contient un tableau d'objets, ou chaque objet represente un service d'IA specifique.

Description des services d'IA

Chaque service d'IA est defini par un objet avec trois proprietes : `type`, `label` et `purpose`.

- `type` : Il s'agit d'une chaine de caracteres unique qui identifie le service d'IA.
- `label` : Il s'agit d'un nom plus descriptif ou d'une etiquette pour le service d'IA.
- `purpose` : Il s'agit d'une description de ce que le service d'IA fait ou de son utilisation prevue.

Exportation du module

A la fin du code, nous exportons l'objet `aiServices` en tant que module par default. Cela signifie que lorsque ce fichier est importe dans un autre fichier, l'objet `aiServices` sera directement accessible.