

Module IA Provider - Version Simplifiée

Package Python pour interagir avec les modèles d'IA GPT-4.1 (OpenAI) et Claude Sonnet 4 (Anthropic).

Structure du Package

```
ia_provider/  
├── __init__.py    # Point d'entrée, expose les éléments publics  
├── core.py        # Classes de base, ProviderManager, exceptions  
├── openai.py      # Implémentation pour GPT-4.1  
└── anthropic.py   # Implémentation pour Claude Sonnet 4
```

Installation

1. Créer le dossier du package :

```
bash  
  
mkdir ia_provider
```

2. Copier les fichiers dans le dossier `ia_provider/` :

- `__init__.py`
- `core.py`
- `openai.py`
- `anthropic.py`

3. Installer les dépendances :

```
bash  
  
pip install openai anthropic python-dotenv pyyaml
```

4. Configurer les clés API dans un fichier `.env` :

```
env  
  
OPENAI_API_KEY=sk-...  
ANTHROPIC_API_KEY=sk-ant-...
```

Utilisation

Import Simple

```
python

from ia_provider import manager, APIError

# Obtenir un provider
provider = manager.get_provider("gpt-4.1") # ou "claude-sonnet-4"

# Générer une réponse
response = provider.generer_reponse(
    "Explique Python en 2 lignes",
    temperature=0.5,
    max_tokens=150
)
print(response)
```

Conversation

```
python

messages = [
    {"role": "user", "content": "Bonjour!"},
    {"role": "assistant", "content": "Bonjour! Comment puis-je aider?"},
    {"role": "user", "content": "Qu'est-ce que Python?"}
]

response = provider.chatter(messages, temperature=0.7)
print(response)
```

Fonctionnalités Spécifiques

GPT-4.1 - Conversion automatique max_tokens

```
python

provider = manager.get_provider("gpt-4.1")

# max_tokens est automatiquement converti en max_completion_tokens
response = provider.generer_reponse(
    "Test",
    max_tokens=200 # → max_completion_tokens=200
)
```

Claude Sonnet 4 - Mode Thinking

```
python

provider = manager.get_provider("claude-sonnet-4")

# Avec mode thinking pour raisonnement approfondi
response = provider.generer_reponse(
    "Résous:  $x + 2y = 10$  et  $3x - y = 5$ ",
    thinking_budget=300, # Tokens pour le raisonnement
    max_tokens=500
)
```

Modèles Disponibles

Modèle	Provider	Particularité
gpt-4.1	OpenAI	Utilise max_completion_tokens
claude-sonnet-4	Anthropic	Support du mode thinking

API Publique

Classes Principales

- BaseProvider : Classe abstraite de base
- ProviderManager : Gestionnaire des providers
- manager : Instance globale du gestionnaire

Exceptions

- APIError : Erreur lors des appels API
- UnknownModelError : Modèle non reconnu

Fonctions Utilitaires

- load_config() : Charge la configuration
- load_api_key(provider_name) : Charge une clé API

Tests

Exécuter les tests :

```
bash
```

```
pytest test_ia_provider.py -v
```

Migration depuis stubs.py

Pour migrer une application utilisant l'ancien module `stubs.py` :

1. Changer l'import :

```
python

# Ancien
from stubs import manager, APIError

# Nouveau
from ia_provider import manager, APIError
```

2. Adapter la liste des modèles : Seuls `gpt-4.1` et `claude-sonnet-4` sont disponibles

Configuration

Créer un fichier `config.yaml` pour les paramètres par défaut :

```
yaml

temperature: 0.7
max_tokens: 1000
top_p: 0.95
frequency_penalty: 0
presence_penalty: 0
```

Exemple Complet

```
python
```

```

from ia_provider import manager

# Lister les modèles disponibles
print(f"Modèles: {manager.get_available_models()}")
# Output: ['claude-sonnet-4', 'gpt-4.1']

# Utiliser GPT-4.1
provider = manager.get_provider("gpt-4.1", api_key="sk-...")
response = provider.generer_reponse(
    "Qu'est-ce que la récursivité?",
    max_tokens=200 # Converti en max_completion_tokens
)

# Utiliser Claude avec thinking
provider = manager.get_provider("claude-sonnet-4", api_key="sk-ant-...")
response = provider.generer_reponse(
    "Problème complexe...",
    thinking_budget=200,
    max_tokens=300
)

```

Licence

MIT

Support

Pour ajouter de nouveaux modèles, modifier le fichier `__init__.py` :

```

python

# Dans ia_provider/__init__.py
manager.register_provider(OpenAIProvider, [
    "gpt-4.1",
    "gpt-4.1-mini", # Nouveau modèle
])

```