AIConfig - the open-source framework for building production-grade AI applications

Documentation

AIConfig is a framework that makes it easy to build generative AI applications for production. It manages generative AI prompts, models and model parameters as JSON-serializable configs that can be version controlled, evaluated, monitored and opened in a local editor for rapid prototyping.

It allows you to store and iterate on generative AI behavior *separately from your application code*, offering a streamlined AI development workflow.

More context here.

Quickstart

For VS Code Users:

- Install the AIConfig Editor VS Code Extension

If you're not using VS Code, follow these steps:

1. `pip3 install python-aiconfig`
2. `export OPENAI_API_KEY='your-key'`
3. `aiconfig edit`

Getting Started Tutorial

Check out the full Getting Started tutorial.

Install

```
# for python installation:
pip3 install python-aiconfig
# or using poetry: poetry add python-aiconfig

# for node.js installation:
npm install aiconfig
# or using yarn: yarn add aiconfig
```

Note: You need to install the python AIConfig package to use AIConfig Editor to create and iterate on prompts even if you plan to use the Node SDK to interact with your aiconfig in your application code.

You must specify your OpenAI API Key. Open your Terminal and add this line, replacing 'your-api-key-here' with your API key: `export OPENAI_API_KEY='your-api-key-here'`.

Open AIConfig Editor in VS Code

AIConfig Editor helps you visually create and edit the prompts and model parameters stored as AIConfigs.

1. Install the AIConfig Editor for VS Code
2. Open the `travel.aiconfig.json` file in VS Code. This will automatically open the AIConfig Editor in VS Code.

Run Prompts in the Editor

With AIConfig Editor, you can create and run prompts with complex chaining and variables. The editor auto-saves every 15 seconds and you can manually save with the Save button. Your updates will be reflected in the AIConfig JSON file. See this example of a prompt chain created with the editor:

Corresponding AIConfig JSON file:

travel.aiconfig.json

Use the AIConfig SDK

You can run the prompts from the aiconfig generated from AIConfig Editor in your application code using either python or Node SDK. We've shown the python SDK below.

```
# load your AIConfig
from aiconfig import AIConfigRuntime, InferenceOptions
import asyncio

config = AIConfigRuntime.load("travel.aiconfig.json")

# setup streaming
inference_options = InferenceOptions(stream=True)

# run a prompt
async def gen_nyc_itinerary():
  gen_itinerary_response = await config.run("gen_itinerary", params = {"order_by" : "location"}, options=inference_options,
        run_with_dependencies=True)

asyncio.run(gen_nyc_itinerary())

# save the aiconfig to disk and serialize outputs from the model run
config.save('updated_travel.aiconfig.json', include_outputs=True)
```

Edit your AIConfig

You can quickly iterate and edit your aiconfig using AIConfig Editor.

1. Open your Terminal
2. Run this command: `aiconfig edit --aiconfig-path=travel.aiconfig.json`

A new tab with AIConfig Editor opens in your default browser at http://localhost:8080/ with the prompts, chaining logic, and settings from `travel.aiconfig.json`. The editor auto-saves every 15 seconds and you can manually save with the Save button. Your updates will be reflected in the AIConfig file.

Why is this important?

Today, application code is tightly coupled with the gen AI settings for the application – prompts, parameters, and model-specific logic is all jumbled in with app code.

- results in increased complexity
- makes it hard to iterate on the prompts or try different models easily
- makes it hard to evaluate prompt/model performance

AIConfig helps unwind complexity by separating prompts, model parameters, and model-specific logic from your application.

- simplifies application code – simply call `config.run()`
- open the `aiconfig` in a playground to iterate quickly
- version control and evaluate the `aiconfig` - it's the AI artifact for your application.

Features

- **Prompts as Configs**: standardized JSON format to store prompts and model settings in source control.
- **Editor for Prompts**: Prototype and quickly iterate on your prompts and model settings with AIConfig Editor.
- **Model-agnostic and multimodal SDK**: Python & Node SDKs to use `aiconfig` in your application code. AIConfig is designed to be **model-agnostic** and **multi-modal**, so you can extend it to work with any generative AI model, including text, image and audio.
- **Extensible**: Extend AIConfig to work with any model and your own endpoints.
- **Collaborative Development**: AIConfig enables different people to work on prompts and app development, and collaborate together by sharing the `aiconfig` artifact.

Use cases

AIConfig makes it easy to work with complex prompt chains, various models, and advanced generative AI workflows. Start with these recipes and access more in `/cookbooks`:

- RAG with AIConfig
- Function Calling with OpenAI
- CLI Chatbot
- Prompt Routing
- Multi-LLM Consistency
- Safety Guardrails for LLMs - LLama Guard
- Chain-of-Verification

Schema

- AIConfig Schema

Supported Models

AIConfig supports the following models out of the box. See examples:

- OpenAI models (GPT-3, GPT-3.5, GPT-4, DALLE3)
- Gemini
- LLaMA
- LLaMA Guard
- Google PaLM models (PaLM chat)
- Hugging Face Text Generation Task models (Ex. Mistral-7B)

If you need to use a model that isn't provided out of the box, you can implement a `ModelParser` for it. See instructions on how to support a new model in AIConfig.

Extensibility

AIConfig is designed to be customized and extended for your use-case. The Extensibility guide goes into more detail.

Currently, there are 3 core ways to extend AIConfig:

1. Supporting other models - define a ModelParser extension
2. Callback event handlers - tracing and monitoring
3. Custom metadata - save custom fields in `aiconfig`

Contribute to AIConfig

We are rapidly developing AIConfig! We welcome PR contributions and ideas for how to improve the project.

- Join the conversation on Discord - #`aiconfig` channel
- Open an issue for feature requests
- Read our contributing guide

Latest Updates

We currently release new tagged versions of the `pypi` and `npm` packages every week. Hotfixes go out when completed.

- Changelog: Check out our latest updates.
- Roadmap: What we're building next - please feel free to contribute. See our contributing guide here.