Playground

The DocETL Playground is an integrated development environment (IDE) for building and testing document processing pipelines. Built with Next.js and TypeScript, it provides a real-time interface to develop, test and refine your pipelines through a FastAPI backend.

Why a Playground? 🤔

This **interactive playground** streamlines development from prototype to production! **Our (in-progress) user studies show 100% of developers** found building pipelines significantly faster and easier with our playground vs traditional approaches.

Building complex LLM pipelines for your data often requires experimentation and iteration. The IDE lets you:

- 🚀 Test prompts and see results instantly
- ✨ Refine operations based on sample outputs
- 🔄 Build complex pipelines step-by-step

Public Playground

You can access our hosted playground at docetl.org/playground. You'll need to provide your own LLM API keys to use the service. The chatbot and prompt engineering assistants are powered by OpenAI models, so you'll need to provide an OpenAI API key.

Data Storage Notice

As this is a research project, we cache results and store data on our servers to improve the system. While we will never sell or release your data, if you have privacy concerns, we recommend running the playground locally using the installation instructions below.

Installation

There are two ways to run the playground:

1. Using Docker (Recommended for Quick Start)

The easiest way to get started is using Docker:

a) Create the required environment files:

Create `.env` in the root directory (for the FastAPI backend):

```
# Required: API key for your preferred LLM provider (OpenAI, Anthropic, etc)
# The key format will depend on your chosen provider (sk-..., anthro-...)
OPENAI_API_KEY=your_api_key_here
BACKEND_ALLOW_ORIGINS=http://localhost:3000,http://127.0.0.1:3000
BACKEND_HOST=localhost
BACKEND_PORT=8000
BACKEND_RELOAD=True
FRONTEND_HOST=localhost
FRONTEND_PORT=3000
```

Create `.env.local` in the `website` directory (for the frontend) **note that this must be in the `website` directory**:

```
# Optional: These are only needed if you want to use the AI assistant chatbot
# and prompt engineering tools. Must be OpenAI API keys specifically.
OPENAI_API_KEY=sk-xxx
OPENAI_API_BASE=https://api.openai.com/v1
MODEL_NAME=gpt-4o-mini

NEXT_PUBLIC_BACKEND_HOST=localhost
NEXT_PUBLIC_BACKEND_PORT=8000
```

b) Run Docker:

```
make docker
```

This will:

- Create a Docker volume for persistent data
- Build the DocETL image
- Run the container with the UI accessible at http://localhost:3000 and API at http://localhost:8000

To clean up Docker resources (note that this will delete the Docker volume):

```
make docker-clean
```

2. Running Locally (Development Setup)

For development or if you want to run the UI locally:

1. Clone the repository:

```
git clone https://github.com/ucbepic/docetl.git
cd docetl
```

2. Set up environment variables in `.env` in the root directory:

```
LLM_API_KEY=your_api_key_here
BACKEND_ALLOW_ORIGINS=http://localhost:3000,http://127.0.0.1:3000
BACKEND_HOST=localhost
BACKEND_PORT=8000
BACKEND_RELOAD=True
FRONTEND_HOST=0.0.0.0
FRONTEND_PORT=3000
```

Create `.env.local` in the `website` directory:

```
OPENAI_API_KEY=sk-xxx
OPENAI_API_BASE=https://api.openai.com/v1
MODEL_NAME=gpt-4o-mini

NEXT_PUBLIC_BACKEND_HOST=localhost
NEXT_PUBLIC_BACKEND_PORT=8000
```

Note

Note that the OpenAI API key, base, and model name in the `.env.local` file are only for the UI assistant functionality, not the DocETL pipeline execution engine.

1. Install dependencies:

```
make install      # Install Python package
make install-ui   # Install UI dependencies
```

2. Start the development server:

```
make run-ui-dev
```

3. Navigate to http://localhost:3000/playground to access the playground.

Setting up the AI Assistant

The UI offers an optional chat-based assistant that can help you iteratively develop your pipeline. It is currently very experimental. It can't write to your pipeline, but you can bounce ideas off of it and get it to help you iteratively develop your pipeline.

To use the assistant, you need to set your OpenAI API key in the `.env.local` file in the website directory. You can get an API key here. The API key should be in the following format: `sk-proj-...`. We only support the openai models for the assistant.

Self-hosting with UI API key management

If you want to host your own version of DocETL for your organization while allowing users to set their API keys through the UI, you'll need to set up encryption. Add the following to both `.env` and `website/.env.local`:

```
DOCETL_ENCRYPTION_KEY=your_secret_key_here
```

This shared encryption key allows API keys to be securely encrypted when sent to your server. Make sure to use the same value in both files.

Complex Tutorial

See this YouTube video for a more in depth tutorial on how to use the playground.