

# AdvandEB Development Environment

AdvandEB Project

December 8, 2025

## 1 Overview

This document provides a printable guide to the shared development environment used by the AdvandEB project. All services share a single Conda environment named **advandeb**.

The main repositories are:

- **advandeb-knowledge-builder** – knowledge ingestion, processing, and exploration.
- **advandeb-modeling-assistant** – modeling-focused retrieval and reasoning (planned).
- **advandeb-architecture** – system-level documentation, diagrams, and environment definition.

## 2 Conda Environment

### 2.1 Creating the Environment

The canonical `environment.yml` file lives in the **advandeb-architecture** repository. To create the environment:

```
conda env create -f environment.yml
```

If the file changes and you need to update an existing environment:

```
conda env update -f environment.yml --prune
```

### 2.2 Activating the Environment

To activate the shared environment:

```
conda activate advandeb
```

This environment is used for:

- Back-end services (FastAPI applications).
- Command-line tools (e.g., `uvicorn`, `plantuml` if installed via Conda).
- Any Python-based utilities for data processing and analysis.

## 3 Core External Services

### 3.1 MongoDB

AdvandEB services use MongoDB for persistent storage.

- Default URI: `mongodb://localhost:27017`
- Typical environment variable: `MONGODB_URL`

Start MongoDB using your preferred method (system service, Docker, or local binary).

### 3.2 Ollama

Ollama provides local hosting for large language models used by the Knowledge Builder and, in the future, the Modeling Assistant.

- Default base URL: `http://localhost:11434`
- Typical environment variable: `OLLAMA_BASE_URL`

Install Ollama following its official instructions and ensure that it is running before starting any LLM-dependent services.

## 4 Running the Knowledge Builder

### 4.1 Back-end

From the `advandeb-knowledge-builder/backend` directory:

```
conda activate advandeb
cp .env.example .env # edit as needed
uvicorn main:app --host 0.0.0.0 --port 8000 --reload
```

Important environment variables in `backend/.env`:

- `MONGODB_URL` – MongoDB connection string.
- `DATABASE_NAME` – database name (e.g., `advandeb_knowledge_builder_kb`).
- `OLLAMA_BASE_URL` – Ollama endpoint.

### 4.2 Front-end

From the `advandeb-knowledge-builder/frontend` directory:

```
npm install
npm run dev
```

The front-end development server typically runs at `http://localhost:3000`.

## 5 Running the Modeling Assistant (Planned)

The Modeling Assistant will also use the shared `advandeb` environment. A likely pattern for the future back-end service is:

```
conda activate advandeb
# from advandeb-modeling-assistant/backend (future)
uvicorn main:app --host 0.0.0.0 --port 8001 --reload
```

It will communicate with:

- The Knowledge Builder’s HTTP APIs for knowledge and agent operations.
- MongoDB for its own collections (e.g., scenarios, models, runs).

## 6 Diagram Rendering

Architecture diagrams are stored as PlantUML files in `advandeb-architecture/diagrams`.

To render all diagrams (assuming `plantuml` is installed and available):

```
conda activate advandeb
cd advandeb-architecture/diagrams
plantuml *.puml
```

This generates image files (for example, PNGs) that can be included in LaTeX documents such as the main architecture specification.

## 7 Troubleshooting

- If Python imports such as `fastapi` or `uvicorn` cannot be resolved, ensure that the `advandeb` environment is active and that it was created from the canonical `environment.yml`.
- If diagrams fail to render with “command not found”, install PlantUML via Conda or your preferred method and re-run the rendering commands.
- For MongoDB or Ollama connection issues, verify that the services are running and that the URLs configured in your `.env` files match the actual hosts and ports.