Backend Manual

Initial Draft Version 1.0

Author: *jd*
Last Reviewed: *December 26, 2022*

Table of Contents

1. Document Objectives and Purpose

This document contains information about installation and usage procedures related to the *backend* that is being developed in the context of this project and the web services it provides. It also contains information considered relevant covering other aspects related to the *backend*.

2. Restrictions

Commented [JD1]: Some restrictions may apply. This is the place where restrictions should be presented and discussed.

3. Assumptions

Commented [JD2]: Some assumptions may apply. This is the place where assumptions should be presented and discussed.

4. Architecture

- The backend web services were developed using FastAPI[1].
- Information about the database that the backend web services use is presented in Section 7.

5. Installation

- To install the web services provided by the backend read the *README.md* file located in the root directory of the source code of each service.
- The file *pyproject.toml*, that can be found in the same directory as the *README.md* file, specifies the dependencies of each service. At the time writing this document, python 3.10^{+} may not yet be compatible with some of the dependencies listed (tested python 3.11 with no success). Therefore, using python 3.9 is recommended to avoid incompatibility issues (used python 3.9.13 successfully).

[1] https://fastapi.tiangolo.com

6. Configuration

- The web services need some configuration, in particular, the external connections allowed must be configured. All other configurations can be left as is. The configuration of the web services can be found in the *config.py* files located in the *<service_src_dir>\app\core* directory.
- Explore the *config.py* files to find what can be configured in each case.

7. Database

- Currently the backend uses *SQLite*. The database can be found in the *<train_model_service_src_dir>\app* directory with the name: *sql_app.db*. This directory is where the database will be created, if it does not exist, by default.
- *Alembic* is used to manage the database initial creation, updates, and versioning.
- The communication with the database is performed using *SQLAlchemy*. *The SQLAlchemy* object-relational mapper (ORM) is also used. The idea is to be able to change the database being used by the backend, with minor changes in the source code.

8. Running/Starting The Backend

- To start the web services provided by the backend read the *README.md* file provided in the root directory of the source code of each service. The documentation uses *Poetry*, but using *Poetry* is not mandatory. Also, in production, the web services should start (be started/restarted) automatically when the server starts, restarts or when a service running crashes. These cases are not considered in the documentation. In principle, the operating system provides mechanisms to manage this cycle, or at least part of it. Automatic service allocation and provisioning are also not discussed and considered.

9. Security and Privacy

- A *reverse proxy* must be installed and running in front of the backend. The *reverse proxy* should be the only ‘external’ entity allowed to communicate directly with the backend. It is the responsibility of the *reverse proxy* to establish the HTTPS connection between the backend and a client and mediate the communication between the two parties.