

## SCOPE

### Backend Module Scope Document

ToDo Application - FastAPI Implementation

#### 1. main.py - Application Launcher

**Core Purpose:** System initialization and production-grade server orchestration

**Responsibilities:**

- Hosts Uvicorn server configuration
- Sets runtime parameters (port, workers, reload behavior)
- Initializes application lifespan events
- Triggers api.py application bootstrap sequence

#### 2. api.py - Application Core

**Core Purpose:** Central FastAPI instance and router federation

**Responsibilities:**

- Instantiates primary FastAPI() object
- Mounts sub-routers from back packages
- Configures exception handlers

**Boundaries:**

- ✗ No database operations
- ✗ No endpoint implementations
- ✓ Pure architectural orchestration

### Database – Core

All Database scripts are allocated in the folder ToDoBD. Inside this folder, there should be an `__init__.py` file in order to set the entire folder as a package.

#### 1. database.py - Database Core

**Core Purpose:** Creates and connects to the database and sets the “task” table

**Responsibilities:**

- validates and creates the folder to locate the BD
- Connects to the BD
- Validate and create the Task table
- The task table has the following configuration:
  - ID integer primary key

- TASK -Text can not be null
- Description -Text
- creation\_time Text is set automatically
- end\_time -Text
- finished -Text possible values ('Y' or 'N'), default 'N'

**Boundaries:**

- ✗ Contains no API awareness
- ✗ Implements no business rules
- ✓ Pure database interaction

## 2. crud.py – Database – Core

**Core Purpose:** Performs de CRUD operations directly to the data base for the “tasks” entity

**Responsibilities:**

- Inserts new tasks
- Reads individual and all information in the database from “tasks”
- Updates the current information of the database
- Deletes individual registries.

**Boundaries:**

- ✗ Contains no API awareness
- ✗ Implements no business rules
- ✓ Pure database interaction

## 3. models.py – Database-Core

**Core Purpose:** Set the data model of the responses needed in the CRUD operations

**Responsibilities:**

- Only Classes in this package

## 4. crud\_router.py – Database Router

**Core Purpose:** Creates the API router to handle the CRUD operations, this method uses crud.py and models.py to handle the logic; it also uses JSONResponse from fastapi for the answers of each operations.

**Responsibilities:**

- POST operation in the creation of new tasks
- GET operation for the get all information query.
- PUT operation for update tasks
- DELETE operation to delete information from the database.

## 5. Validation.py – Database – Core

**Core Purpose:** Creates the classes and functions to validate and sanitizes that data in every CRUD operation in the app usage. This validation is meant to be used in crud.py

### Responsibilities:

- Validate the data in order to allow operations, if not raise an error
- Validate the data for any SQL injection
- Create a meaningful error message for each case if needed
- Clean every text from HTML tags and scape the characters

## LLM – Core

All Database scripts are allocated in the folder ToDoBD. Inside this folder, there should be an `__init__.py` file in order to set the entire folder as a package.

### 1. AI\_model\_exp.ipynb – LLM experiments

**Core Purpose:** This notebook has the different alternatives to call a CodeGPT agent in order to receive a description or summary of the “tasks” entity in the database.

### Responsibilities:

- Test the connection via request POST to the CodeGPT agent
- Test the alternatives to connect or take the information of the database into the agent
- Test different prompts in order to compare the accuracy of the prompt and the information in the database
- Tests the change in the Agent response after an immediate change in the information in the database

### 2. Llm.py – LLM Core

### Core Purpose:

- Class CodeGPTLLM, based in CustomLLM from LLaMaindex library to set the methods of connection to the CodeGPT Agent and in an object recognizable for the library logic.

- A method that calls on CodeGPTLLM to set up the object in order to create a Query Engine that is going to connect the todo.db to the CodeGPT Agent.

**Responsibilities:**

- Connection via request POST to the CodeGPT Agent
- Connect todo.db to the CodeGPT Agent

**3. llm\_router.py****Core Purpose:**

- This package is responsible to assemble the routers to be called as API of the llm section
- Contains a constant that is the principal PROMPT to be send to the LLM in order to receive the summary of the current information of the table “tasks” in the database.

**Responsibilities:**

- POST operation in the summary of the current database information
- GET operation of the confirmation of the llm API