**SCOPE**

Backend Module Scope Document  
ToDo Application - Asynchronous FastAPI Implementation

### 1. main.py - Application Launcher

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

* Hosts Uvicorn ASGI 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 routers
* Declares root endpoint health checks
* Manages global dependency injections
* Configures exception handlers  
  Boundaries:  
  ❌ No database operations  
  ❌ No endpoint implementations  
  ✅ Pure architectural orchestration

**All Database scripts will be 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.

### 3. 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**

**4. crud.py – Database – Core**

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

**Responsibilities:**

* Inserts new task
* Reads individual and all information in the bd
* updates the current information of the bd
* deletes individual registries.

Boundaries**:  
❌ Contains no API awareness  
❌ Implements no business rules  
✅ Pure database interaction**