01UDFOV/01TXYOV – WEB APPLICATIONS I

BIGLAB 2A: APIS WITH EXPRESS

WHAT ARE WE BUILDING THIS WEEK?

We will create a **basic back-end for the task manager**. To do so, we will use the Express framework. The back-end has to implement a series of **APIs** to support the main features of the web-based task manager we developed in BigLab1: **create**, **read**, **update** and **delete** the tasks. The data will be persistently stored in an **SQLite database**.

STEP-BY-STEP INSTRUCTIONS:

- Design a set of APIs to support the main features of our web-based task manager. The APIs should allow the application to:
  - **retrieve** the list of all the available tasks;
  - **retrieve** a list of all the tasks that fulfill a given filter (e.g., all the important tasks, all the tasks with a given deadline, etc.);
  - **retrieve** a task, given its "id";
  - **create** a new task, by providing all relevant information – except the "id" that will be automatically assigned by the back-end;
  - **update** an existing task, by providing all relevant information (all the properties except the "id" will overwrite the current properties of the existing task. The "id" will not change after the update);
  - **mark** an existing task as completed/uncompleted;
  - **delete** an existing task, given its "id".

  Data *passed to* or *received from* the API should be in **JSON format**. You must list the designed APIs, together with a short description of the parameters and the exchanged entities, in the README.md file that is included in your repository. Be sure to identify which are the collections and elements you are representing. You might want to follow this structure for reporting each API:

  ```
  [HTTP Method] [URL, optionally with parameter(s)]
  [One-line about what this API is doing]
  [Sample request, with body (if any)]
  [Sample response, with body (if any)]
  [Error response(s), if any]
  ```

- Implement the designed HTTP APIs with Express. The tasks must be **stored persistently** in an SQLite database. To this end, you can use the database "tasks.db" that is included in your group's BigLab2 repository.
  The database contains two tables, "users" and "tasks", with each user that may have one or more tasks. For the moment, consider the information of the "tasks" table, only (the "user" column is always set to 1, the only user available). Besides the usual information (*id, description, important,*

private, deadline), the table also contains a column to mark a task as completed. The information about users will be used in the last week of this BigLab, to implement the login functionality.

- Test the realized API with a tool such as PostMan (https://www.postman.com/), ReqBin (https://reqbin.com) for Google Chrome or RestClient for Firefox (https://addons.mozilla.org/en-US/firefox/addon/restclient/).

Hints:

1. *The general specification of the second BigLab can be found at: https://github.com/polito-WA1-AW1-2021/course-materials/blob/main/labs/BigLab2/BigLab2.pdf*
2. *To visualize and edit the database, you can exploit the "DB Browser for SQLite" application, downloadable from https://sqlitebrowser.org/ or SQLiteStudio, available at https://sqlitestudio.pl/*