01UDFOV/01TXYOV Applicazioni Web I / Web Applications I [2024/2025]

Lab 3: APIs with Express

In this lab, you will develop a simple back-end for your chosen exam topic using the Express framework. The back-end will consist of a set of APIs to support CRUD operations (Create, Read, Update, Delete) for your items. The data will continue to be stored in the SQLite database created in Lab 2.

1. API design

Design a set of APIs to support your application based on the objects identified and on the database created in the previous labs. Data passed to or received from the API should be in JSON format. The APIs should allow a web application to:

a. **Retrieve** the list of all items of the main collection.
b. **Retrieve** a list of items with specific characteristics.
c. **Retrieve** a specific item, i.e., given its "id".
d. **Create** a new item by providing all its information, except the "id" that will be automatically assigned by the back-end.
e. **Update** an existing item, by providing its information, i.e., all the properties except the "id".
f. **Update** specific attributes of a specific item.
g. **Delete** an existing item.

List the designed APIs, together with a short description of the parameters and the exchanged entities, in the README file of your group repository. Be sure to identify which are the collections and elements you are representing, as seen in class. 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]

2. API implementation

Implement the designed HTTP APIs with Express. Items are stored persistently in the SQLite database you created for Lab 2 and you can start from the methods implemented in that lab. You may want to create a back-up copy of the database before modifying it. New items may need to be assigned to a specific user, if so assign its user to an already existing user (e.g., user with id=1).

Remember to add proper validations to the implemented APIs. For instance, you should check that IDs are integer values, or you should check that any date format is valid.

3. API implementation

Test the realized API with the REST Client extension for Visual Studio Code[1]. To this end, you will have to write an API.http file according to the following syntax:

```
[HTTP Method] [URL, optionally with parameter(s)] HTTP/1.1
content-type: application/json (if needed)

[Sample request, with body (if any), in JSON format]
###
```

[1] Visual Studio Code REST Client extension is available at the following link: https://marketplace.visualstudio.com/items?itemName=humao.rest-client