Luigi De Russis, Alberto Monge Roffarello

LAB 6 – REST IN PYTHON

GETTING STARTED

The goal of this set of exercises is to design and implement a set of REST APIs for the todo list manager storing tasks with id, description, and urgent.

Recap:

1. *Fork your own copy of the Git repository associated with this lab (https://github.com/AmI-2019/python-lab6) to your personal GitHub space*
2. *Open PyCharm Professional and select Checkout from Version Control > Git in the "Welcome to PyCharm" window, to clone your (forked) repository*
3. *Fill the requested fields (repository URL, location on disk, …) and press the "Clone" button*
4. *Once the project is open, you can create a new Python file by right clicking on the project name (Project tab, on the left) and selecting New > Python File*
5. *Commit and push the changes you made back to GitHub, from the VCS menu in PyCharm*

EXERCISE 1 – API DESIGN

Design a set of APIs for the todo list manager developed in the previous laboratories.

N.B.: differently from the in-class example (https://github.com/AmI-2019/todolist-REST), a task for this todo list manager is composed by the following fields:

- id: the unique identifier of the task, modeled as an integer number
- description: the task content, modeled as a string
- urgent: whether the task is urgent or not, modeled as a boolea

The APIs should allow to:

a) Retrieve the list of available tasks
b) Create a new task
c) Retrieve the task identified by the given task id
d) Update an existing task
e) Delete an existing task

You are strongly encouraged to follow the design method presented in the lesson "REST services with Python and Flask: a case study" (https://elite.polito.it/files/courses/01QZP/2019/slide/Web-02-restapi.pdf) and inspired on the Google API Design Guide (https://cloud.google.com/apis/design/)

EXERCISE 2 – IMPLEMENT THE DESIGNED API

Implement the designed APIs in a REST server. As in the previous laboratories, the server should use a database to store the task information. You can find a working version of a database to store tasks with an id, a description, and a urgent field in the solution of the previous lab (https://github.com/AmI-2019/python-lab5/tree/solution). In the repository, you will also find a library to interact with the database.

EXERCISE 3 – IMPLEMENT A SIMPLE CLIENT FOR GETTING TASKS

Develop a python script to test **all** the implemented APIs. For this purpose, you can use the *request* module, as shown in the in-class example (https://github.com/AmI-2019/todolist-REST).