Distributed Systems Programming

A.Y. 2021/22

Exam Assignment for the Exam Call on 03/02/2022

Deadline for submission: 31/01/2022 EOD

Modify the solution of Laboratory 5, by extending the use of MQTT, to improve the consistency of the task information displayed in the Public Tasks pages of the clients, according to the following specifications:

1. Whenever any modification in a public task is performed by means of the REST APIs (create, delete, update, including modification from non-public to public and vice versa), the server must promptly inform all clients by means of MQTT messages. These messages must be JSON objects that carry the full information about the occurred change (e.g., if a new task is created, the message must carry the full information of the new task).
2. Each client that is displaying the Public Tasks page must receive, via MQTT, the information about the changes occurred in what is being displayed, and it must update its view and local information accordingly. For example, if the deadline of one of the tasks being displayed changes, the deadline must be updated in the view. Or, if a new task is created or becomes public, the new task must appear in the view.

Design the MQTT topics and messages according to the guidelines studied in the course. If you want, you can extend/modify the design proposed for Lab 5, making sure that, in the end, everything works (i.e., not only the Public Tasks view works according to these specifications, but also the My Tasks view works according to the Lab 5 specifications).

Implement the designed solution and test it. To facilitate testing, the application must be configured so that the pagination mechanism shows at least 10 tasks at the same time (rather than just 2 tasks). Showing more than two tasks allows you to better understand if the view if correctly updated according to the information received via MQTT.

Submit the updated solution, which must have the same folder organization as the Lab 5 solution, including all the following additional items:

- A README.md file in the main folder. The file must include:
    - An explanation of the design choices made about the MQTT topics, messages, and configurations (e.g., retention policy, QoS, etc.).

- The files containing the schema(s) of the messages used over MQTT (in the Server/json_schemas folder). Please refer to these files in the explanation.

Important:

- The solution must work within the Labinf VMs, with the software already installed in those machines.
- The solution must be uploaded to a git repository for which you will get the credentials.

Useful tips

- PublicList.js is the file including the functions that generate the content of the GUI of the Public Tasks page.
- In API.js, getPublicTasks() is the function that creates the HTTP Request to send to the ToDoManager service to retrieve the list of public tasks.
- In App.js, getPublicTasks() and refreshPublic() are the functions that are executed when the content of the Public Tasks page must be visualized or updated.
- If you are using the solution proposed for Lab5, you can easily change the number of tasks shown in each page by modifying the number in the constants.js file.