Distributed Systems Programming

A.Y. 2023/24

Exam Assignment for Exam Call on 28/02/2024

Deadline for submission: 25/02/2024 EOD

Modify the solution of Laboratory 2, by introducing 1) client identification and 2) per-client registration and negotiation of some parameters of the gRPC Converter service, according to the following specifications:

1. Each client of the Converter service is uniquely identified in the service by means of an id, which is a string.
2. Each client of the Converter service must be registered in the service before being able to use the service to perform file conversions. A client sends its id to the service in each file conversion request and the service accepts the request only if the id is registered. The service keeps the registration information in memory without persistency. In the registration phase, a client chooses its id and negotiates some parameters of the service.
3. If the chosen id is already taken by another registered client, the service refuses registration.
4. The parameters of the service that can be negotiated by each client are:
    a. The image types that can be used for the *input* files and, for each type, the maximum allowed input file size (size 0 means no maximum).
    b. The image types that can be used for the *output* files and, for each type, the maximum allowed output file size (size 0 means no maximum).
5. Negotiation works in this way: the client requests the *desired* parameter values, and the server decides the *actual* parameter values, which must be a subset of the desired ones: if an image type is desired by the client for input or for output, it can be granted or not by the server. If it is granted, the server may establish for it a maximum allowed file size which is less than or equal to the desired one.
6. When a client requests a file conversion, the server checks that the request respects the negotiated parameters and, if not, it refuses the request.

Design the interface of the updated service (proto file) and modify the implementation of the Java service and of the Node client accordingly.

When designing the updated interface (proto file), pay attention to error management, and to the guidelines taught in the course (note that in the proposed solution error management was not fully developed).
The modified Java service must read the information about the acceptable parameters from a file with lines having the following format:
<input/output> <type> <maxsize>
where <input/output> is either “input” or “output”, <type> is an image type string and <maxsize> is the maximum admissible size for that case in kbytes. For example:
input PNG 2000
input JPG 10000
output PNG 5000
output JPG 10000
The result of the negotiation will be the minimum of the desired maximum value and the maximum admissible value for each allowed type.
The Node client reads its id and the desired parameters from a file with similar structure as the one read by the Java server, but with an initial line expressing the id.
In the REST API implementation, add a new API that can be used to read the service parameters negotiated by the Node client with the server. Moreover, update the REST API implementation so that it accepts media types and file sizes compliant with the result of the negotiation.

Submit the updated solution, including all the following items:

- The proto file for the conversion service
- The Java implementation of the conversion service
- All the json schemas
- The full Open API documentation of the REST APIs, including examples of JSON documents to be used when invoking the operations, and examples of invocations of the API operations, possibly as a Postman collection
- The REST APIs implementation code
- The database file, including some sample data
- README.md files that specify the contents of folders, the instructions on how to run the code from scratch and the main design choices made.

Important:

- Organize the project files as in the solution provided for Laboratory 2.
- The solution must work within the Labinf Ubuntu machines, with the software already installed in those machines.
- The solution will be tested by running it in the Labinf machines.
- The solution must be uploaded to a git repository for which you will soon get the credentials.