

Assignment 2: Service Oriented Architectures

Web services and REST

Recall Assignment 1 and the architecture diagram you designed. We now want to start implementing the system and write Python modules for some of the services. For each service, you have to build only a part of the functionality, which is specified in the service description below. The main focus will be on **defining and implementing the interfaces** that each service exposes:

1. *Authentication service*: this service will be responsible for authenticating and logging users into the system. The following user groups (roles) are defined:
 - Administrator: can create and delete users and manage all types of data.
 - Secretary: can manage some data.
 - Manager: can submit optimization requests (batch jobs) and collect results.

The service will expose an authentication API with the following functions:

- A function that accepts username/password and, if correct, emits a simple token (containing user role and a random base-64 string).
- A function for verifying that the token for a given user is still valid.

This service will be implemented with an in-memory cache (without persistent storage).

2. *Master data service*: this service will be responsible for the metadata for the simulations and manage at least two data tables in a persistent data store:
 - jobs: a table containing metadata information for each submitted simulation job (user who sent the job, timestamp submitted, status (submitted, processing, done), date range, assets (a collection of integers from 1 until 100))
 - results: a table containing metadata information for the result of each job (job ID, timestamp, assets/weights (a collection of pairs asset number/a real number between 0.0 and 1.0))

The service will expose an API for submitting jobs and fetching and updating data on running or done jobs. Note that only users of the user group *managers* and *administrators* are allowed to use this service. All other users and unauthenticated users will get an authorization error.

Deadline: 2023/04/18, 23:59 CET.

Notes

- All files shall be submitted in a single zip file.
- A README.MD file will also be included in the submission with a short description of the submitted files.
- You may build a simple web UI for testing - since UI development is out of the scope of the course, it will not be graded.
- Every request performed by a client and all server responses must be logged with the following information: source, destination, headers (if applicable), metadata (if applicable), message body.
- You have free choice as to what web service/communication framework to use for each service: please explain your choice in the README.MD file.
- Don't spend much time on the business logic: please focus on setting up the services, APIs between services and logging.

Assessment

Total: 15 points.

- All requirements are satisfied: 10 points.
- The documentation is concise and the choice on which web service/framework was used is well explained and technically correct: 5 points.