C# client-side interface for TuSoW

Giacomo Cavalieri Nicolò Di Domenico

January 26, 2021

The project aims to create a C# client-side library that makes it possible to interface with the TuSoW server, leveraging the asynchronous programming paradigm that comes with the chosen language.

1 Goals and requirements

This project has two main goals:

1. since TuSoW has no public documentation available, the first step will be to figure out the valid HTTP routes, their semantics and their mandatory/optional arguments
2. the design and implementation of the proper client-side library, leveraging C#'s native asynchronous programming features in order to comply with the LINDA semantics without stalling the thread that makes the HTTP request

One extra goal, which is not strictly mandatory but would improve the quality of life for the development team, is implementing an efficient CI system by using GitLab's own tooling, so that each merge request is safely tested before being merged into master.

2 Division of tasks

Initially the whole team will be focusing on analyzing TuSoW's source code in order to document the available REST endpoints. Next, the team's efforts will be completely focused on the design of the library.

Once the design is done and deemed satisfactory, each member of the team will begin working on writing the actual code. The chosen Git workflow (which to the team's knowledge has no "standard" name) follows these rules:

- the `master` branch is always guaranteed to work
- each feature gets its own branch
- the only way to push to `master` is by creating a merge request, which must be reviewed and approved by the other team member

The quality of the code will be guaranteed thanks to GitLab's own CI pipeline, which also has the task of running tests and denying the approval of a merge request until all of them pass.

Di Domenico will be in charge of setting up the CI system. The design of the library and the writing of the final report will be done by both members. Each feature will be developed by mostly one member (there may be occasional fixes and suggestions from the other), and the expected workload of the whole project should be roughly evenly split between the two team members.