Collaborative Code Editor

Yazan Abdullah

Contents

Chapter 1

Introduction

In this document, I will present my solution to the final project in Atypon's Training.

Part I

Backend

Chapter 2

Code Execution: Docker Remote API

I used docker containers to safely fun users' code. I allowed for maximal flexibility, by cloning users' entire projects to the container, allowing them to structure their projects and run them like they would in a local environment.

To interact with docker, I used Docker remote rest API over the web. This is to allow me to move the docker server to a dedicated machine for performance and security purposes.

I designed a DockerClient class, that uses a WebClient object to issue commands and receive results from the docker server.

Each method in my docker client is responsible for an operation; it receives a configuration object holding the different parameters needed for the method.

Configuration objects can be huge in terms of the number of parameters, this is why I used the builder design pattern with them. I provided a directive that will further ease preparing the configuration objects.

Chapter 3

Real-Time Collaboration: Websockets & Conflict Resolution

hello

Chapter 4

Database: MongoDB

Chapter 5

Version Control

1. Layer on top of the database 2. stable/active project 3. merge conflicts: needleman-wucnsch

Chapter 6

Spring Security

Part II

Frontend

Chapter 7

React with Typescript

Chapter 8

Code Editor: Monaco Editor

Chapter 9

Styling the Frontend: MUI