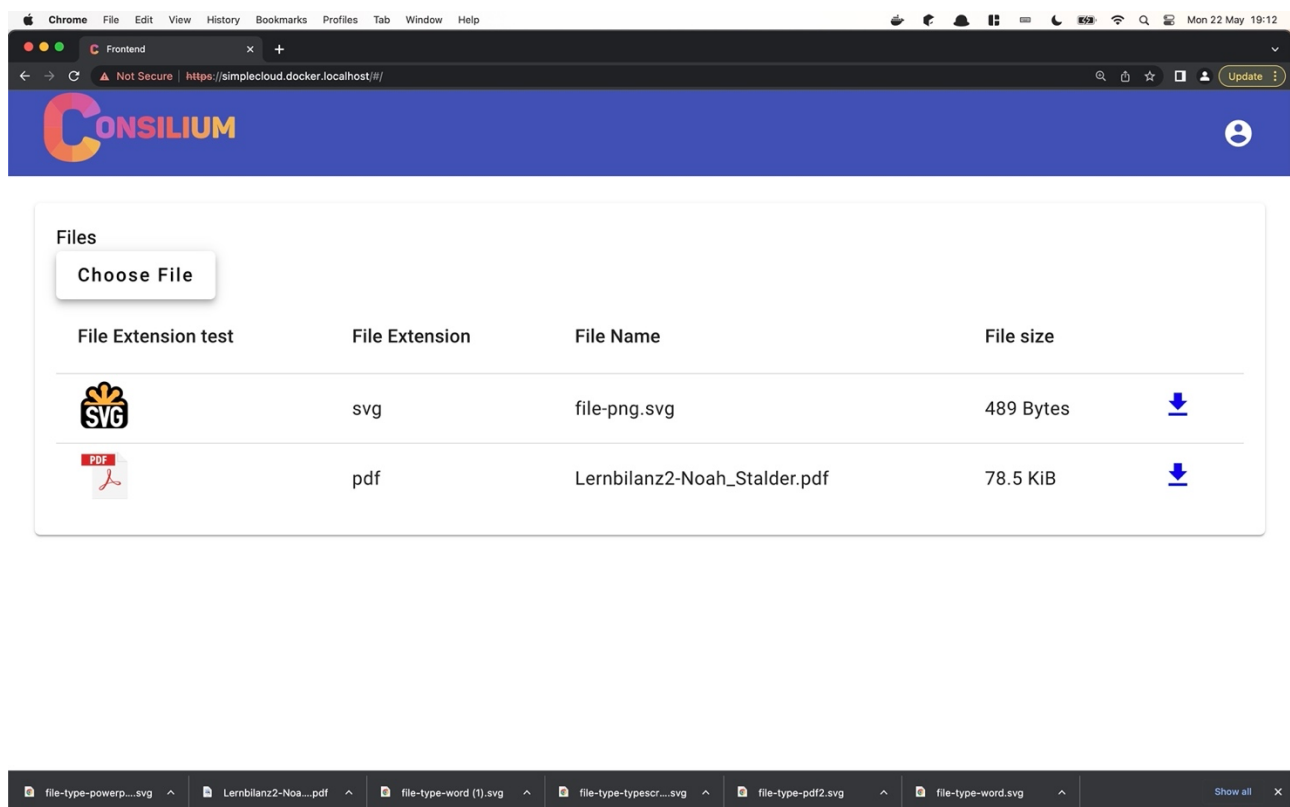


1 DSYS Challenge Task Solution

1.1 Solution Overview

For the DSYS challenge task we implemented a file share. Our solution allows for upload and download of files:



Via clicking the “Choose File” button a file can be uploaded (selected by a file browser).

All uploaded files are available for download and listed in a table. The table displays the icon of the file type, the name of the file and the file size.

1.2 How to start

Our solution can be started by running ‘mvn install’ to generate everything and then in the folder /frontend run ‘npm install’ and ‘npm run build’.

Make sure to set the correct URL in the “traefik_dynamic.yml” file. The first part of the url needs to correspond with the folder the project lays in. For example if the folder is named “dsys-challenge” the URL needs to be “dsys-challenge-frontend-1:8080”.

The whole setup can then be started using “docker compose up”.

1.3 Roles

For the development we decided to focus on the load balancing part of our solution together. Joel took the lead with the backend development and the stack and Noah did the frontend.

1.4 Architectural Considerations

1.4.1 Architectural Overview

We decided to use a stack we are familiar with consisting of an Angular frontend, a Java backend and a PostgreSQL Database.

For the communication between the frontend and the backend we use a REST API.

The Java backend uses Jooq to access and modify the data in the database.

As we used a standard Angular setup we won't go into further details architecture wise.

1.4.2 Backend Architecture

Given the size of this assignment we decided to use a simple architecture which is still cleanly separated. The controller package contains the REST Resources of the Files and the Health check. The needed DTOs lay in the package controller/dto.

The service folder contains all business logic and accesses the database. We decided against the use of a separate repository layer which we would usually use to separate data and logic, as the split remains clear without.

1.4.2.1 Health Controller

The health controller is used to signal once the startup is complete.

1.4.2.2 File Controller

The file controller offers endpoints for upload and download of files. Additionally, it offers an endpoint to fetch all Infos necessary for the frontend to display the list of downloadable files.

1.4.2.3 File Service

The file service stores, loads and aggregates the necessary data for the file controller.

1.5 Load balancing

The load balancing consists of three instances of the frontend and backend each. With an interval of 10 seconds the health check endpoint is called to register the once the services is booted up.