

Chevron DataScience Challenges Website

Technical Documentation

Written by Team C Minus 1 during CSM Summer Field Session 2023

Please be aware that this project was developed in the span of ~4 weeks. We apologize for any major issues that you run into :)

Front-End Information

Built on:

- Angular CLI: 16.0.1
- NodeJS: 18.16.0
- Npm: 9.5.1
- TypeScript: 5.0.4

Required packages (run “npm install package_name”):

Note that some may be installed by default

- Router
- Forms
- Common
- Rxjs
- Core
- Http
- Ionic
- BCryptJS

Default User:

- Email: zeroadmin@chevron.com
- Password: mine\$4lYfe

By default, the default the Front End Runs on internal port 3000

Connection to backend:

Assuming this stack is being run inside of a docker container, the two value services, “ValueService” and “Upload” rely on hard-coded URLs for interface with the back end. You can find these URLs in:

- ‘client/src/app/value-service.service.ts’

and

- 'client/src/app/submissions/upload.service.ts'

By default, the backend interface page operates on port 5000 and the value of configUrl is '<http://localhost:5000>'. A proxy manager or other routing mechanism may be needed in order to deploy properly. If the backend server resides in separate containers, then you will need to change localhost to the host-side, static IP address of that container.

Back-End Information

Built on:

- NodeJS: 18.16.0
- Npm: 9.5.1

Required packages (run "npm install package_name"):

Note that some may be installed by default

- express
- body-parser
- cors
- pg
- multer
- morgan
- Serve-index
- Path
- Debug
- Myapp:server
- Tfjs
- BCrypt

By default, the backend web server "interface" runs on internal port 5000

The backend relies on two file locations /uploads and /challenges. These folders MUST exist and should probably be mapped to a volume from the container to the host. Additionally, sometimes (depending on which type of OS has modified the files most recently) bcrypt needs to be rebuilt. This can be done with "npm rebuild bcrypt --build-from-source".

Database Information

Connection to Database:

Built on:

- Postgres

By default, the postgres server listens on port 5432

The 'pg' module in the backend acts as a client that can send and receive query results. The entire schema including all table columns MUST exist in the database in order for the front and back end to behave properly. This should be done automatically during the creation of the docker container (inside of the DockerFile), but if for any reason these tables need to be reset, the 'setup-script.sql' file contains lines of code in order to create the necessary tables and set up user permissions.

Docker Information

Built on:

- Docker Engine 23.0.5
- Postgres 15.3 (Debian 15.3-1)
- Lite-Server 2.6.1 (npm install)
- NodeJS 18.16.0

Please find the DockerFile in the project directory. Do NOT move any files when building a container, as they are referenced relative to the location of the DockerFile. All errors (frontend, backend, postgres) will be viewable in the container's log

Environmental Variables (not required - must alter code!):

- PGUSER postgres
- PGHOST localhost
- PGDATABASE postgres
- PGPASSWORD password
- PGPORT 5432

If you wish to alter these parameters, then you must change lines 17-21 of /server/index.js to reference ENV variables found in /server/keys.js

Volumes:

- /path/to/folder:/server/challenges

- Holds sample dataset and answer keys per challenge
 - Naming scheme: (challengeID)_sample.csv, (challengeID)_key.csv
- /path/to/folder:/server/uploads
 - Hold user submissions
 - Naming scheme: (submissionID)_upload.csv
- /path/to/folder:/var/lib/postgresql/data
 - In order to save database information outside of the container
 - If you do not do this one, you will lose all user, submission, and challenge data!!

Ports (container):

- 3000
 - Frontend
 - “EXPOSE” by default in DockerFile
- 5000
 - Backend information pages. See /server/index.js
 - “EXPOSE” by default in DockerFile
- FOR DEBUGGING ONLY
 - WARNING: these services are NOT designed to be published and may introduce security vulnerabilities if published accidentally
 - 3001
 - Frontend manager
 - 5432
 - pSQL database access

Other Notes

Due to the complexity and limited timeline of the project, as of June 12, 2023, we have not deemed this project suitable for use by end-users. Most major functionality has been implemented, but there are many limitations:

- Testing
 - In order to reach our deadline with a minimum viable product, major end-user testing was unfortunately not something that we could reasonably complete
 - Most testing has been done while angular is running in development mode, which differs slightly from the production build of the website (ng build)
- Challenge Infrastructure
 - All required functions for a challenge are present, BUT:
 - No front-end feature to edit an existing challenge
 - Minimal user pop-ups regarding state of a challenge
- Error handling

- While there are many error checking and handling methods currently in place, there is definitely still room for errors
- The largest issue present regards the back-end. Currently, if the backend cannot communicate with the postgres instance, it will crash and the container must manually be restarted