Which file determines how to build a Docker image?

Respective dockerfiles.

What is the difference between a Docker image and a container?

A docker image determines what the service is, what languages and services it is running and so on. The container is the actual environment the service or application will be running in, virtualizing the environment on the local mahine. Why do you need to set the port when running the containers? So that docker knows through which ports to connect to the service from outside the container itself (as our web browser for example does not run in the docker container and thus needs to have a metaphorical "door" it can enter through). The same principle applies to services that are running in multiple docker containers, as is the case here. Both containers, frontend and server, are separate from each other, and the specified ports facilitate the ability to communicate with each other.

What did you need to change in order to start a second server?

We had to change the `SERVER_LIST` value defined in the frontend dockerfile to include the second server started at `127.0.0:8002:80`. After saving the changes we then had to rebuilt the image and restart the containers to display both connected servers in the frontend.

Why do changes in the code are not directly reflected in the application?

For each change in the code the docker image has to be rebuilt before restarting the container.

Which REST URIs are defined in the server?

- `/entries` (POST)
- `/entries` (GET)
- `/entries/<entry_id>` (PUT)
- `/entries/<entry_id>` (DELETE)
- `/message` (POST)