Setup

1. Install Java Open-JDK Version 1.8.0\_181
2. Download the repository (from link or by git cloning)
   1. <https://github.com/cs414glaad/cs414-f18-001-the-other-alex>
   2. git clone [git@github.com:cs414glaad/cs414-f18-001-the-other-alex.git](mailto:git@github.com:cs414glaad/cs414-f18-001-the-other-alex.git)
3. Download and install Postman: <https://www.getpostman.com/apps>

Directories

Project directories

* client – The client-side UI code.
* server - Our server-side Java code.
* docs - Our documentation and design software artifacts.

Config files

* .gitignore - A .gitignore file for git.
* .travis.yml - The travis-ci integration file which tells travis-ci how to run the build and test scripts. (Note: Travis-CI knows how to run a gradle test, but it doesn’t work with JUnit, so we had to add the gradle wrapper.)
* README.md - The readme file explaining basic instructions on how to run the project.

Gradle related files

* gradle/wrapper - The wrapper so that the client or developer don’t need to install gradle themselves. Also, this must be here for travis-ci to work.
* build.gradle - The build file. Nothing to build in the root folder, so this is emtpy.
* gradlew – The gradle wrapper script for unix like operating systems. It can be run to do any gradle build commands.
* gradlew.bat - The gradle wrapper for windows (same as gradlew)
* settings.gradle - The settings for the project.

Building and Testing the Server

To build the server, run the following:

./gradlew build

To test the server, run the following:

./gradlew test

To test only server-side, without testing client-side code (because client-side code takes longer to package), run the following:

./gradlew :server:test

Running the Server

1. Once downloaded and extracted, run the following within the root of the project directory:

./gradlew run

Read the output of the terminal and wait for the console to log that the server has started on a port:

Server started on port: 3001

(You’ll see a progress bar for gradle, since the server is running, just ignore it as it is misleading.)

1. To stop the server, press ctrl+c, and it will run the server’s shutdown routine (the project handles the SIG\_INT event call on the process).

Connecting to the Server

1. Once the server is running, you can connect to it through a web browser at the port configured (3001 by default). For example:

{server-ip}:3001 from a remote machine

localhost:3001 from the host computer