Admin Site

The companion web app to the **IBM FISE Concierge** allowing friends of (elderly)
users to set up accounts and manage the details of all members in their circle.
Admins have the ability to add and remove
services from member accounts and view service usage history for each member.

Prerequisites

To get started with development, you will need a browser that runs `JavaScript` and some extra prerequisites:

- Node.js - Package manager
- npm - This should come with your Node installation.
- A (relatively recent) browser that allows JavaScript - more information on supported browsers can be found in
  the `package.json` file.

Installation

To install the necessary packages, run the following command:

```
npm install
```

Deployment

Before running the application, ensure the
backend is running on `localhost`
on port `8100`.

- Should you want to run the server on a different port, you must change the `PORT` value in
  the `src/backend/api.js` script to match before running.

The backend can also be started by running:

```
npm run-script backend
```

To view the web app in a development server run the following command:

```
npm run-script start:dev
```

To view the web app in production mode (with all necessary optimisations performed) run the following command:

```
npm start
```

For convenience, the backend can be started along with the (development) web app in a single command.

The development version:

```
npm run-script start:dev:backend
```

The production version:

```
npm run-script start:backend
```

Note that this starts the backend without any logging information and also starts and shutdowns the server on every run.

- It is recommended you start the backend manually and leave it running in the backend instead of using these two
  commands.

Testing

The tests can be run using the following command:

```
npm test
```

The coverage report can be viewed in your default browser using:

```
npm run-script coverage
```

The report is the `index.html` file located in `coverage/lcov-report` if a browser window is not opened after all test
suites have run.

The `tests/unit/components` directory contains integration tests for the `.vue` single file
components with the working store and backend interaction.

Note that the testing suite will spawn a new instance of the backend server to use for testing and will shut it down
once complete. Ensure no other service is using port `8100` prior to running these tests.

However, this starts and shutdowns the backend every time the test suite is run. If you would like to run the tests
without this, opting to run the server manually in the background:

```
npm run-script test:noserver
```

Similarly, for the coverage report:

```
npm run-script coverage:noserver
```

Publish

To publish an update to the website, run the following command:

```
npm run-script deploy
```

A live version of the web app can be found here should you not want to serve it locally.

- Note that the backend server **must** still be manually started.

**Note that the web app does not work on Safari if not served locally (i.e., on `localhost` instead of
at `fise-concierge.web.app`) due to cross-origin restrictions and the lack of a HTTPS certificate for the backend.**

- Safari blocks API calls to `localhost` as the backend uses `HTTP` (instead of `HTTPS`) - it also does not allow
  self-signed SSL certificates to be used to get around this.
- Using self-signed certificates would prevent the web app from working entirely on Safari.