

Design Documentation

High level

This project follows the standard structure of most NextJS projects. There are essentially two different node projects that exist independant of eachother which represent the seperation of the front end and the backend.

All user interation is handled by the **ui** portion of the project. Throught the **ui** the user is able to make calls which reflect changes in the **api** portion of the project.

For example the user is able to create an account for the site by interacting with the **createAccount** page which makes the required calls to the **api** portion of the project. The **api** portion of the project will handle whatever is required to create the user account.

The **ui** directory represents the front end. The **api** directory represents the backend.

UI

The UI consists of **pages** which represent a web page that the user is able to interact with and the **components** which represent reusable parts that can be used within any of the **pages**.

All of the **ui** pages serve strictly as an interface to handle user interaction. There isn't anything that a user can do to modify items in the account store or **api** folder that would circumvent the **api** project.

API

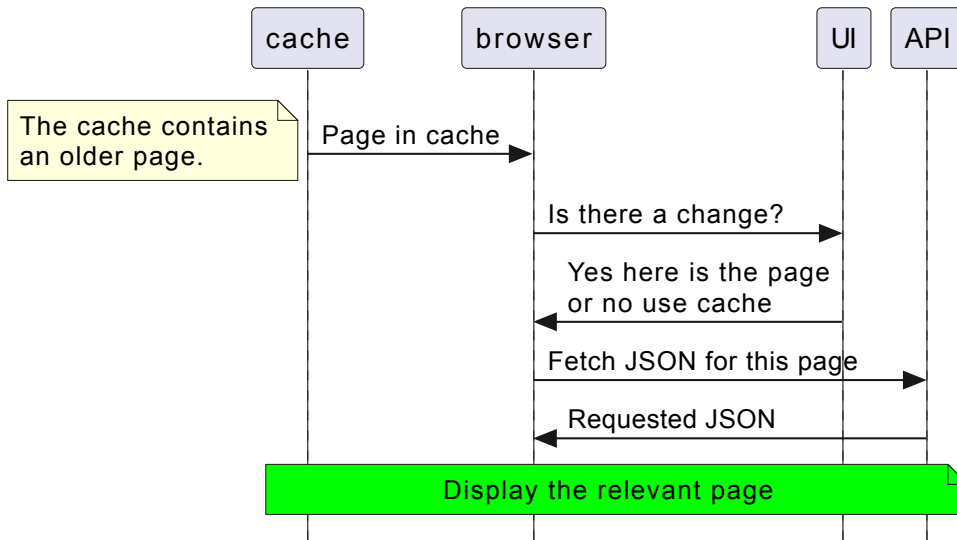
The **api** project serves strictly as a worker for managing things such as user login state, links that the user wants to share, or the user bio. The **api** project can best be seen as a warehouse worker who will fetch or store the requests that are made by the **ui** project.

The **api** project is strictly restricted to actions that interact with the data that we store for user information. **api** will never be used for actually displaying the information or handling it in any way that would modify it. Only for serving and storing.

Examples

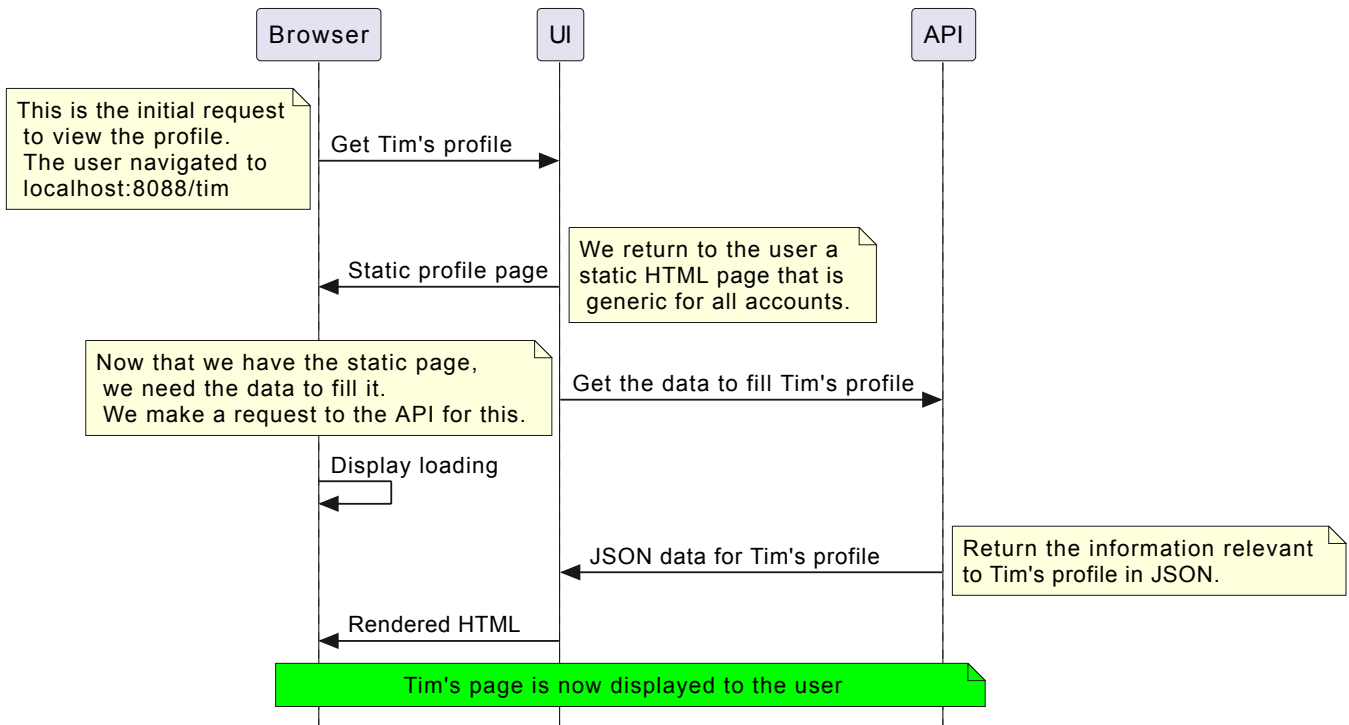
Page Caching

Flow of operations for page caching



Loading a profile

Flow of operations for fetching the user Tim's profile



Directory Structure

```

.
├── api
│   ├── index.js # The point in which we initialize the server with any
│   │           # required settings and define the api routes.
│   ├── routes
│   │   ├── accounts.js # Contains required logic for users to create and
│   │   │               # account and login.
│   │   ├── profile.js # Contains logic for loading a user profile.
│   │   │               # Slightly unique because it cannot require a login.
│   │   ├── session.js # Manages the user login session. This ensures that
│   │   │               # as users navigate between pages that we keep them logged in between all of
│   │   │               # them
│   │   └── upload.js # Provides the api for uploading an image from the
│   │               # front end. This will in turn save the uploaded image on the UI side of the
│   │               # project.
│   ├── store
│   │   └── accounts.js # Serves as the database for user account
│   │               # information.
│   └── util
│       └── rand.js # Simple helper class for creating a random string
├── ui
│   ├── components # The components directory contains a reusable
│   │               # components that are intended to be used by files in the pages directory
│   │   ├── UserBackground.tsx # A component that overrides the default
│   │   │               # background for use profile pages.
│   │   ├── footer.js # The footer used by all pages other than user
│   │   │               # profiles
│   │   ├── layout.js # The standard layout of all pages. It wraps a pages
│   │   │               # body content with the header and footer components.
│   │   ├── linkEditor.js # Component used by the manage profile window
│   │   │               # which provides users an interactive way of editing their links
│   │   └── navbar.js # The navbar or header used by all pages except
│   │               # profile
│   ├── css
│   │   └── styles.css # Style sheet that applied to all pages
│   ├── hooks
│   │   └── session.js # UI end of managing a users login session
│   ├── pages
│   │   └── [profile].js # This page displays a user's profile. No session
│   │               # is required.
│   │   ├── _app.js # global page the defines how the others will look
│   │   ├── account
│   │   │   ├── background.js # Allows the user to modify their background
│   │   │   │               # image used on their profile.
│   │   │   ├── createAccount.js # Allows a user to create a new account.
│   │   │   ├── links.js # Allows a user to edit the links on their
│   │   │   │               # profile.
│   │   │   ├── login.js # Allows a user to login to their account
│   │   │   └── logout.js # Doesn't represent a page, but allows the user
│   │   │               # to trigger a logout request.

```

```
├── manage.js # Allows a user to manage their account.
├── index.js # The homepage of the web app.
├── styles.module.css # contains the site background color
└── public
    ├── backgrounds # A directory that contains all of the user's
uploaded background images.
    ├── USERPLAINWHITE.jpg
    ├── rwGgk8wiF0Q.jpg
    └── example.png
```