ActiveTO Web Project Overview

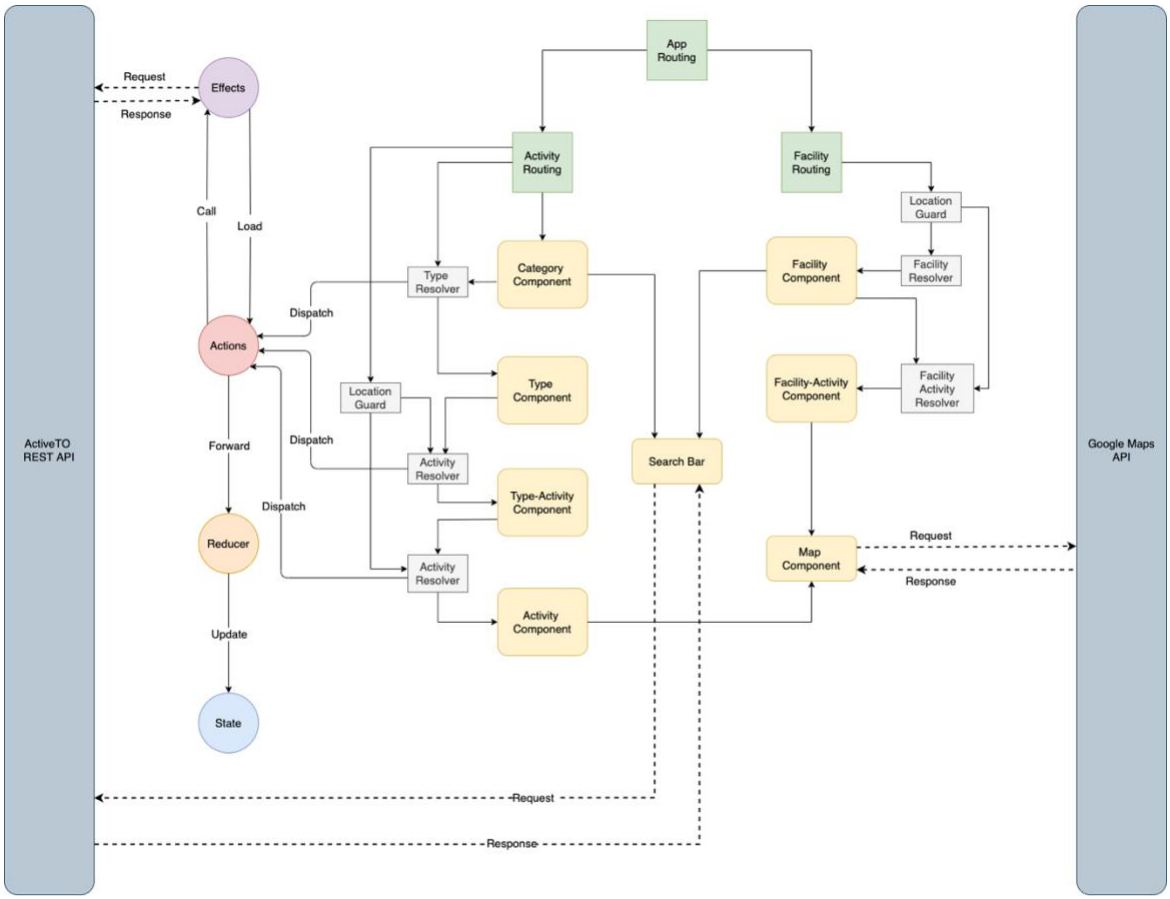
1. Project Summary

Project Name: ActiveTO Web

Technologies Used: Angular, NgRx, Material UI, Node.js, Google Maps API

Description
ActiveTO is a publicly accessible website that designed to provide an easy access for users to browse drop-in activities at any City of Toronto's recreation facilities as well as facilities' information. The application allows users to browse, search and sort activities by category, type and time and visualize locations on a map.

2. Architecture Overview



Core Components

- Component-Based Structure: The application is built using Angular, with a focus on reusable components. Each component represents a distinct part of the UI.
- Resolvers and Guards: Resolve necessary data for rendering components by dispatching to the NgRx store and prevent components from being rendered before data is fetched.
- State Management with NgRx: State management is handled using NgRx, implementing a Redux-inspired architecture. This allows for a predictable state flow and makes it easier to manage complex interactions within the application.

- Routing: Angular's RouterModule is used for handling navigation within the application. It enables deep linking and lazy loading of modules, improving performance and user experience.
- API Integration
    - ActiveTO REST API: Services are used to manage data fetching, caching, and error handling, ensuring a seamless user experience.
    - Google Maps API: Integrated to provide facility location visualization and directions.
- Third-Party Integrations
    - Material UI: Used for responsive design and ensuring consistency across different devices and screen sizes.

3. Deployment and CI/CD Pipeline

Tools & Technologies

- CI/CD Platform: GitHub Actions
- Version Control: Git/GitHub
- Build Tool: Webpack
- Proxy server: Nginx
- Hosting platform: DigitalOcean

Pipeline Workflow

- Build: Code is pulled from the GitHub repository and built with Angular command line tool.
- Deploy: Built static files are copied to the DigitalOcean instance via SSH.

4. Security Considerations

- Environment Variables: Sensitive information such as API key is stored in environment variables on the GitHub repository and never exposed in the client-side code.
- CORS Handling: Proper CORS policies are implemented on the backend to ensure secure API communications.

5. Live Demo and Source Code

- Live Demo: https://www.mollyzhang.dev/apps/activeto
- Source Code: https://github.com/dongyue-zhang/ActiveTO_web