**Blog Application**

|  |  |
| --- | --- |
| Blog Application | Version 1.0 |
| Software Architecture & Design Document | Date: 07-09-2020 |
| Author | Gayaprasad Tiwari |

Table of Contents

[**1.** **Introduction** 3](#_Toc50478798)

[2. **Architecture** 3](#_Toc50478799)

[**3.** **Approach** 3](#_Toc50478800)

[**4.** **Accessibility** 4](#_Toc50478801)

[**5.** **Device support** 4](#_Toc50478802)

[**6.** **Unit test** 4](#_Toc50478803)

[**7.** **Services** 4](#_Toc50478804)

[**8.** **Component** 4](#_Toc50478805)

[**9.** **Lit Element** 5](#_Toc50478806)

[**10.** **Ngrx** 5](#_Toc50478807)

[**11.** **Tools and libraries used** 5](#_Toc50478808)

[**12.** **Open Item** 5](#_Toc50478809)

[**13.** **List of improvement** 6](#_Toc50478810)

[**14.** **Setup and run** 6](#_Toc50478811)

# **Introduction**

This is a blog application where user can see the list of blogs in home page. A user can register and log into the application, after login user can add, edit and delete the blogs.

# **Architecture**

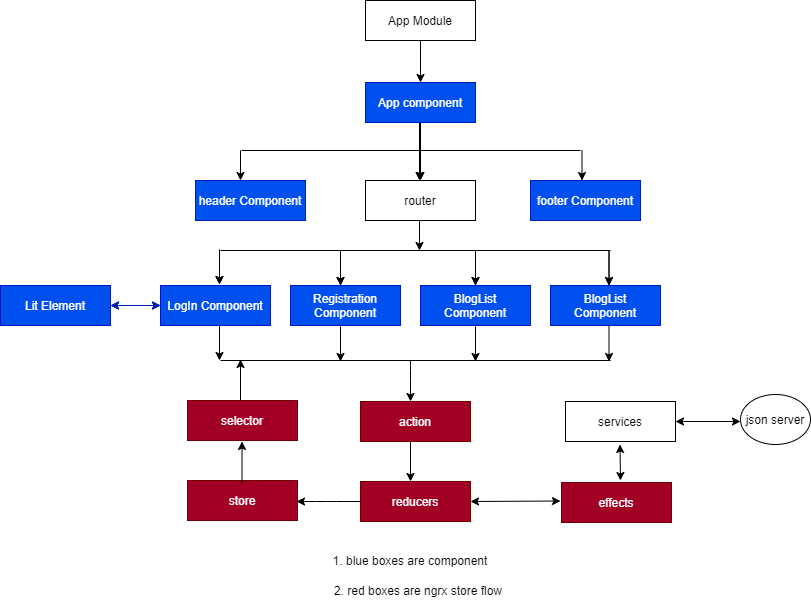


Figure 1: Architecture for Blog Application

# **Approach**

Application consist of routes, NGRX and one Lit Elements and has the theming implemented for light and dark themes. For API calls JSON-server is used.

Below is routes view

* List View – it is for displaying blog list and user can edit and delete blog when it is logged in.
* Add-edit - user can add new blog in application.
* Log in – so user can log in.
  + Added lit login- element inside the log-in component
* Registration – so user can do registration.

In Application component is using to ngrx store for state management

# **Accessibility**

* Added tab index
* Added aria-label, area-placeholder

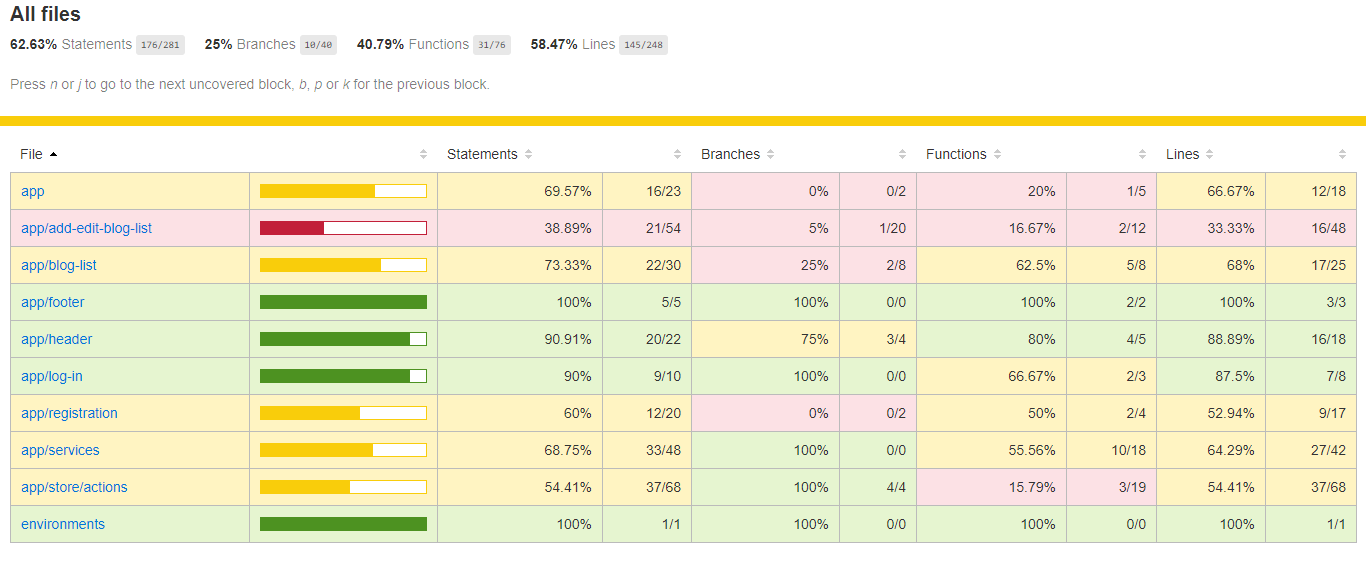
# **Device support**

* Application is responsive supported for mobile , table and desktop

# **Unit test**

Test Suites: 10 passed, 10 total

Tests: 16 passed, 16 total



# **Services**

Below services used in Blog app

* auth.service - this related to login and registration
* blog-service.service – this is related to blog list, add and edit
* theme.service - this is for theming

# **Component**

* HeaderComponent– contains headings and link
* AddEditBlogListComponent- adding and editing blog
* BlogListComponent – displaing blog list with add and edit button
* FooterComponent – footer for application
* LogInComponent- for log in
* RegistrationComponent- for user registration

# **Lit Element**

* LoginElement- web component for log in

# **Ngrx**

Below NGRX classes used in Blog app

* Actions
  + auth.actions
  + blog.action
* Reducers
  + auth.reducers
  + blog.reducer
* Effects
  + blog.effects
  + auth.effects

# **Tools and libraries used**

* "@angular/animations": "~10.1.0",
* "@angular/common": "~10.1.0",
* "@angular/compiler": "~10.1.0",
* "@angular/core": "~10.1.0",
* "@angular/forms": "~10.1.0",
* "@angular/platform-browser": "~10.1.0",
* "@angular/platform-browser-dynamic": "~10.1.0",
* "@angular/router": "~10.1.0",
* "@ngrx/effects": "^10.0.0",
* "@ngrx/store": "^10.0.0",
* "lit-element": "^2.4.0",
* "pwa-helpers": "^0.9.1",
* "reselect": "^4.0.0",
* "rxjs": "~6.6.2",
* "tslib": "^2.0.0",
* "zone.js": "~0.10.2"

# **Open Item**

* storybook

# **List of improvement**

* Users profile page can be added
* Blog details page can be added
* can add some other details for blog

# **Setup and run**

* Install Node with npm
* Install angula cli
* git clone from master branch  “https://github.com/gayaprasad-Tiwari/blogs.git”
* cd blogs
* npm install
* npm start
* for running json server - “json-server db.json -m ./node\_modules/json-server-auth “
* fot test – “npm run test-watch