

Test Report

Angular Calendar Application

Mansour Jouya

Overview

This report outlines the testing process and results for the Angular calendar application, developed according to specified requirements. The application leverages Angular Material and Angular CDK, adhering to Angular best practices such as dependency injection, lazy loading, Angular forms, RxJS, and routing. The tested features include creating, deleting, and moving events via drag-and-drop, calendar rendering using ngFor, and new capabilities such as event display by hour, event editing and deletion, and month navigation.

Features

1. Display of Days and Months:

- The calendar displays days and months, with the number of events shown on each corresponding day.

2. Event Hours Display:

- Clicking on a day shows the events' hours (from 0 to 23).

3. Event Creation, Editing, and Deletion:

- Multiple events can be created for each hour of a day.
- Events are editable, deletable, and movable via drag-and-drop.

4. Event Sorting:

- Events within each hour are sorted by time, ensuring correct order when multiple events exist in the same hour.

5. Month Navigation:

- Users can navigate between months, with the date automatically updating.

Test Scope

The scope of the testing includes the following:

1. Event Creation:

- Testing the ability to add events to the calendar.
- Ensuring the event form uses Angular reactive forms with proper validation and form control handling.

2. Event Deletion:

- Verifying the functionality to delete events from the calendar.

3. Event Editing and Drag-and-Drop:

- Testing the ability to edit and move events using drag-and-drop functionality.

4. Event Hour Display:

- Ensuring the correct display of event hours (0-23) after selecting a specific day.

5. Event Sorting:

- Verifying that events are sorted correctly by time, with events displayed in chronological order for each hour.

6. Month Navigation and Date Update:

- Testing the ability to navigate between months and confirming the date updates correctly.

7. Lazy Loading and Routing:

- Testing lazy loading for modules and routes to enhance performance.
- Verifying proper routing functionality using router-outlet.

8. RxJS and Dependency Injection:

- Ensuring correct use of RxJS for asynchronous operations and dependency injection for service interactions.

Testing Phases

1. Environment Setup:

- Ensuring the project is built correctly and can run without errors.
- Verifying that the application passes linting rules (eslint) without issues.

2. Functional Testing:

- **Event Creation:** Testing the event form's functionality, ensuring data binding, form validation, and interaction with the event service.
 - **Event Deletion:** Verifying the deletion of events by interacting with the delete button or other UI controls.
 - **Event Editing and Drag-and-Drop:** Testing the edit and drag-and-drop functionality for events.
 - **Event Hour Display:** Verifying the correct display of event hours for a selected day.
 - **Event Sorting:** Ensuring events are sorted correctly by time, especially when multiple events exist in the same hour.
 - **Month Navigation:** Testing the ability to navigate between months and confirming the date updates.
3. **Code Quality:**
- Reviewing the code to ensure best practices are followed, including the proper use of Angular features like lazy loading, dependency injection, and RxJS.
4. **Cross-Browser Compatibility:**
- Verifying that the application functions correctly across major browsers (Chrome, Firefox, Safari, Edge).

Test Results

1. **Event Creation:**
 - The form works as expected, allowing users to add events with proper validation. Data is correctly bound to the form, and changes trigger the required operations.
2. **Event Deletion:**
 - Events are correctly deleted upon interaction with the delete button in the event UI.
3. **Event Editing and Drag-and-Drop:**
 - The edit and drag-and-drop functionalities work as intended, enabling users to modify and move events.
4. **Event Hour Display:**
 - The hours for each day (0-23) are correctly displayed after selecting a day.
5. **Event Sorting:**
 - Events are sorted accurately by time, with multiple events within the same hour being displayed in the correct order.
6. **Month Navigation and Date Update:**

- Month navigation works correctly, with the date updating automatically.

7. Lazy Loading and Routing:

- Lazy loading for modules and routes functions correctly, improving performance.
- Routing works as expected with router-outlet, enabling smooth navigation between views.

8. RxJS and Dependency Injection:

- RxJS is effectively used for managing asynchronous data flows, and services are correctly integrated using dependency injection.
















Conclusion

The Angular calendar application meets all specified requirements. It passes linting rules, builds without errors, and the core features—event creation, deletion, editing, drag-and-drop, calendar rendering, event hour display, and month navigation—function as expected. Additionally, the application follows Angular best practices, including lazy loading, dependency injection, and RxJS.

The application has been thoroughly tested for functionality, code quality, and performance, and is ready for deployment.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Thursday, December 12, 2024

+	0:00	
+	1:00	
+	2:00	<div><div>New Project Review</div><div>A meeting to review the progress of the project and delegate tasks to team members.</div><div></div></div> <div><div>Client Contract Meeting</div><div>Discussion and negotiation for signing a new contract with the client for future projects.</div><div></div></div>
+	3:00	
+	4:00	<div><div>Colleague Feedback Review</div><div>A session to review feedback received and plan improvements in processes.</div><div></div></div>
+	5:00	<div><div>Marketing Team Meeting</div><div>Review of marketing strategies for the new season and analysis of last month's performance.</div><div></div></div> <div><div>Support Team Review</div><div>Evaluation of the support team's performance and planning improvements in customer response.</div><div></div></div>
+	6:00	
+	7:00	