The Journey of ng-ui-pack

From Idea to Published Angular UI Library

By Saurabh Sharma

Table of Contents

1. Day 1: Project Setup
2. Day 2: Button Component
3. Day 3: Modal Component
4. Day 4: Navbar Component
5. Day 5: Card Component
6. Day 6: Tabs Component
7. Day 7: Toast Component
8. Day 8: Showcase Page
9. Day 9: Library Setup
10. Day 10: Library Publish
11. Tools & Technologies Used
12. Outcome & Reflections

Day 1: Project Setup

- We began by creating an Angular 17 project using standalone components.
- The project was named ng-ui-pack.
- We discussed the folder structure and the importance of SCSS variables for consistent design.

Day 2: Button Component

- Created a reusable Button component with multiple styles: primary, secondary, danger, and disabled.
- Introduced design tokens (SCSS variables) for colors, spacing, shadows, and typography.
- This laid the foundation of our design system.

Day 3: Modal Component

- Implemented a Modal component with open/close behavior.
- Used *ngIf and CommonModule for conditional rendering.
- Handled Angular hydration/SSR errors by disabling client hydration temporarily.

Day 4: Navbar Component

- Built a responsive Navbar with brand, links, dropdown menu, and mobile hamburger toggle.
- Styled using SCSS variables and media queries for responsiveness.
- Integrated it into the demo app for preview.

Day 5: Card Component

- Created a Card component with image, title, description, and slot for actions.
- Added hover effect with shadow for better UI experience.
- This made the library more suitable for dashboards and listings.

Day 6: Tabs Component

- Designed a Tabs container and Tab child component.
- Leveraged ContentChildren and ngTemplateOutlet for flexible dynamic tab content.
- Enabled developers to define tab labels and inject custom templates.

Day 7: Toast Component

- Implemented a toast/alert system for success, error, and info messages.
- Added auto-dismiss after 3 seconds and manual close button.
- This provided essential UX feedback for apps using the library.

Day 8: Showcase Page

- Created a showcase page (app.html) demonstrating all components together.
- Added a hero header, section blocks, and styled layout.
- Deployed the demo on GitHub Pages using gh-pages branch.
- This gave the library a live face for developers to explore.

Day 9: Library Setup

- Generated Angular library project: ng-ui-pack-lib.
- Moved all components from app into the library folder structure.
- Fixed SCSS imports by moving _variables.scss inside lib/.
- Updated public-api.ts with correct exports for all components.

Day 10: Library Publish

- Built the library using ng build ng-ui-pack-lib.
- Successfully published ng-ui-pack-lib v0.0.1 on npm.
- Tested locally by installing it in demo app via npm install ./dist/ng-ui-pack-lib.
- Cleaned .gitignore, removed unnecessary files, and confirmed everything worked.

Tools & Technologies Used

- Angular 17 (Standalone Components, SSR/Hydration handling).
- SCSS with @use for variables and theming.
- GitHub Pages with angular-cli-ghpages for demo deployment.
- npm for package publishing.
- Conventional commits for clean Git history.

Outcome & Reflections

- The ng-ui-pack-lib was successfully published on npm.
- A live showcase was hosted on GitHub Pages.
- A maintainer guide and full documentation were created for long-term use.
- This journey transformed an idea into a polished, distributable Angular UI library.