PrimeNG Migration: Version 15 → Version 20

Section 1: Migration v15 → v16

PrimeNG v16 focuses on Angular v16 compatibility and enhancements: - **Dependency Updates:** Update to `primeng@^16.0.0`, `primeicons@^5.1.0`, `primeflex@^3.3.0`. - **Theming:** Introduced dynamic theme loading API for CSS variables; use `PrimeNGConfig.set('theme', 'lara')` dynamically. - **Accessibility Improvements:** Enhanced ARIA roles on key components like Button, Table, and Dialog. - **Removal of Deprecated Classes:** Any legacy CSS variables or classes from v10–v14 removed; ensure SCSS overrides use new variable names. - **Migration Steps:** 1. Update `package.json` and run `npm install`. 2. Adjust theming initialization in `main.ts` or `app.module.ts`. 3. Rebuild and run tests. 4. Verify ARIA attributes and perform accessibility audit.

Section 2: Migration v16 → v17

PrimeNG v17 brings performance optimizations and new component variants: - **Performance:** VirtualScroller and Table optimizations reduce bundle size by code-splitting component modules. - **New Components:** Introduced `MegaMenu` with improved keyboard navigation. - **Breakpoints Update:** PrimeFlex updated grid breakpoints from `p-col-` classes to support CSS container queries optionally. - **Migration Steps:** 1. Update to `primeng@^17.0.0` in `package.json`. 2. Review import paths for Lazy-loaded component modules. 3. Update grid CSS classes if using container query mode. 4. Run build and tests; validate menu keyboard flows.

Section 3: Migration v17 → v18

PrimeNG v18 aligns with Angular v18 features: - **Signals Integration:** Components emit signals APIs for reactive rendering; opt into signals with `PrimeNGConfig.enableSignals()`. - **Zoneless Mode Support:** Dialog and Overlay components support zone-less operation. - **Deprecations:** Removed support for legacy dialog templates (`` / ``) entirely. - **Migration Steps:** 1. Bump to `primeng@^18.0.0`. 2. Enable signals in `app.module.ts`. 3. Replace any deprecated template tags with ``. 4. Execute application and validate change detection boundaries.

Section 4: Migration v18 → v19

PrimeNG v19 introduces SSR hydration improvements and standalone component support: - **SSR Hydration:** Templates support hydration-friendly markup for faster server-side rendering. - **Standalone Components:** UI components available as standalone exports for direct use without NgModule. - **Theme Builder:** CLI tool to generate custom themes via JSON configuration. - **Migration Steps:** 1. Install `primeng@^19.0.0` and `primeicons@^5.2.0`. 2. Update overlays to container-friendly hydration wrappers. 3. Migrate modules to standalone imports (`import {Button} from 'primeng/button'`). 4. Generate and include custom theme via `primeng theme` CLI.

Section 5: Migration v19 → v20

PrimeNG v20 stabilizes the builder and introduces Vue and React wrappers: - **Builder Enhancements:** PrimeNG CLI Builder supports incremental rebuilds and filesystem caching. - **Vue/React Support:** Experimental wrappers for Vue 3 and React 18 released; see `@primeng/react` and `@primeng/vue` packages. - **Deprecations:** Removed legacy PrimeFlex utility classes in favor of CSS utility modules (`@primeng/utils-css`). - **Migration Steps:** 1. Upgrade to `primeng@^20.0.0`, `primeicons@^5.3.0`. 2. Replace PrimeFlex references with `@primeng/utils-css` imports. 3. Evaluate Vue/React wrapper usage if relevant to polyglot architectures. 4. Run full regression and cross-framework integration tests.