

PrimeNG Migration: Version 9 → Version 10

Overview

PrimeNG 10 marks a major overhaul under the “PrimeOne” design system. Key structural changes include renaming all CSS class prefixes from ui- to p-, switching change detection to OnPush, updating PrimeIcons to v4.0, and integrating the PrimeFlex utility CSS library for layouts. This migration guide covers everything needed to upgrade a large Angular application from PrimeNG v9.1.3 to PrimeNG v10.x.

1. Class Prefix Renaming (ui- → p-)

All CSS classes used by PrimeNG components in v9 that started with ui- are renamed to p- in v10. This affects every component’s root, state, and utility classes. For example:

Examples:

- .ui-button → .p-button
- .ui-state-disabled → .p-disabled
- .ui-dialog → .p-dialog, .ui-dialog-titlebar → .p-dialog-header
- .ui-datatable → .p-datatable, .ui-state-highlight → .p-highlight

2. Theming and CSS Variables

PrimeNG v10 introduces CSS variables for theming (e.g., --p-primary-color). Legacy SASS-based themes from v9 (\$primaryColor) are still supported if you import the old SASS themes, but you are encouraged to migrate to CSS variable-based themes for runtime theming and dark mode support. Override variables in a global stylesheet:

Example:

```
:root {  
  --p-primary-color: #007ad9;  
  --p-primary-text-color: #ffffff;  
}
```

3. Dependency Updates

In package.json, bump dependencies to match PrimeNG 10 requirements:

Example:

```
"primeng": "^10.0.0",  
"primeicons": "^4.0.0",  
"primeflex": "^2.0.0"
```

4. Module Imports

Ensure your Angular modules import the correct PrimeNG modules. Most module names remain unchanged, but verify each import resolves. E.g.:

Example:

```
import { ButtonModule } from 'primeng/button';  
import { DialogModule } from 'primeng/dialog';
```

5. Component Structure Changes

Some components had minor DOM structure changes, particularly around content projection slots. The `<p-header>` and `<p-footer>` tags were deprecated in favor of `<ng-template pTemplate>`:

Before:

```
<p-dialog>
<p-header>Title</p-header>
</p-dialog>
```

After:

```
<p-dialog>
<ng-template pTemplate="header">Title</ng-template>
</p-dialog>
```

6. Removal of Legacy Grid Classes

PrimeNG v9 included basic grid CSS (`.ui-g`, `.ui-g-12`). In v10, these were removed in favor of PrimeFlex (`.p-grid`, `.p-col-12`). Install and include PrimeFlex CSS:

Command:

```
npm install primeflex@^2.0.0
```

angular.json:

```
"styles": [
  "node_modules/primeflex/primeflex.css",
  "src/styles.scss"
]
```

7. Migration Pitfalls & QA

- Update test selectors (e.g., E2E tests using `.ui-` classes).
- Use visual regression testing to catch styling issues.
- Cross-browser testing: ensure CSS variables support.
- Accessibility: verify ARIA attributes in Dialog/Overlay.
- Premium templates: update their CSS overrides similarly.

8. Step-by-Step Transition Plan

1. Create migration branch.
2. Update dependencies and run `npm install`.
3. Update module imports.
4. Bulk rename CSS classes (`ui-` → `p-`).
5. Update SCSS overrides.
6. Migrate content projection.
7. Include PrimeFlex, adjust grid classes.
8. Run app, fix template errors.
9. Execute tests, update selectors.
10. QA and deploy.