

## Migration Guide: NgModules to Standalone Components (Angular 15–19)

This guide outlines the recommended approach to migrate Angular applications from NgModule-based architecture to modern Standalone Components introduced in Angular 14 and stabilized in Angular 15+.

### 1. Audit Your Current App Structure

- List all components, pipes, and directives.
- Identify NgModules that declare them.
- Classify whether each item is a candidate for standalone conversion or tightly coupled.

### 2. Migrate Leaf Components First

- Start with components that are not reused and don't rely heavily on module-level DI.
- Update their definition using `standalone: true` in the `@Component` decorator.

Example (Before):

```
@NgModule({ declarations: [UserCardComponent] })  
  
export class UserCardModule {}
```

Example (After):

```
@Component({  standalone:  true,    selector:    'app-user-card',    templateUrl:  
'./user-card.component.html', imports: [CommonModule] })  
  
export class UserCardComponent {}
```

### 3. Adjust Import Paths

- Remove declarations from NgModules.
- Import the standalone component directly in consuming components via 'imports' property.

### 4. Provide Required Services (DI)

- Use `provideHttpClient` with interceptors in `bootstrapApplication`.
- Alternatively, import modules like `HttpClientModule` in the standalone component directly if not bootstrapping.

## 5. Update Routing to loadComponent Syntax

Before (NgModule-based):

```
{      path:      'dashboard',      loadChildren:      ()      =>
import('./dashboard/dashboard.module').then(m => m.DashboardModule) }
```

After (Standalone):

```
{      path:      'dashboard',      loadComponent:      ()      =>
import('./dashboard/dashboard.component').then(m => m.DashboardComponent) }
```

## 6. Clean Up Obsolete NgModules

- Delete modules once all internal components are migrated.
- Move shared providers if needed to feature provider modules.

## 7. Test, Refactor & Optimize

- Use ng test, ng lint, and ng build --configuration=production.
- Apply lazy loading and shared providers where appropriate.

## 8. Compatibility Notes

### 9. CLI Migration Support (Optional)

- Generate standalone component manually:

```
ng generate component my-comp --standalone
```

- Auto-migrate (Angular 16+):

```
ng generate @angular/core:standalone-migration
```