

Standalone - Angular

Od Angular 14 (stabilna wersja od Angular 15) mamy możliwość tworzenia aplikacji w sposób **standalone**. Ale nie musimy stosować tego dla całej aplikacji – możemy także tworzyć pojedyncze komponenty, dyrektywy lub pipe'y w sposób **standalone** i używać ich w aplikacji bazującej na NgModule's.

Standalone odnosi się do tego, że dany element aplikacji (np. komponent) jest utworzony bez wykorzystania modułu (nie musimy go zadeklarować / eksportować w żadnym module). To samo dotyczy się dyrektyw i pipe. Taki **standalone** element posiada wszystkie zależności w swojej klasie w tablicy **imports** lub **providers** (np. dla serwisów).

Aby oznaczyć element jako **standalone**, musimy użyć metadana **standalone: true** w dekoratorze używanym na klasie.

```
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [CommonModule, RouterOutlet, FooterComponent],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
```

Migracja projektu na standalone

Od wersji Angular 15.2.0 (oraz wyższe) możemy migrować swoje projekty bazujące na modułach w projekty standalone w sposób automatyczny. Angular CLI udostępnia komendę:

ng generate @angular/core:standalone

Przed próbą migracji upewnij się, że Twój projekt ma wersję Angular 15.2.0 lub większą i że buduje się bez żadnych błędów. Do tego najlepiej zrób kopie projektu / utwórz osobną gałąź GIT. Kroki do wykonania w celu migracji:

1. Uruchom **ng g @angular/core:standalone** i zaznacz „Convert all components, directives and pipes to standalone”.
2. Uruchom **ng g @angular/core:standalone** i zaznacz „Remove unnecessary NgModule classes”.
3. Uruchom **ng g @angular/core:standalone** i zaznacz „Bootstrap the project using standalone APIs”.

Po wykonaniu tych kroków aplikacja powinna zostać przekonwertowana w aplikację standalone – jednak należy dokładnie sprawdzić, czy nie spowodowało to jakichś błędów i czy nie są wymagane jakieś ręczne zmiany w plikach.