

# Guards

Tzw. **route guards** w Angular pozwalają nam między innymi zabezpieczać wybrane ścieżki w adresie aplikacji przed nieautoryzowanym dostępem, załadować potrzebne dane przed wejściem na wybraną ścieżkę itd. Typy guard'ów, które są dostępne w Angular:

`canActivate`

Używany do zabezpieczenia ścieżki w adresie aplikacji przed nieautoryzowanym dostępem

`canActivateChild`

Używany do zabezpieczenia zagnieżdżonych ścieżek (child routes) w adresie aplikacji przed nieautoryzowanym dostępem

`canDeactivate`

Używany do sprawdzania, czy użytkownik może opuścić ścieżkę, w której aktualnie się znajduje (przydatne np. przy formularzach, kiedy chcemy upewnić się, że użytkownik nie wyjdzie bez zapisania postępów)

`canMatch`

Używany gdy definiujemy w aplikacji dwie takie same ścieżki, z różnymi widokami do wyświetlania treści – np. mamy zaimplementowany nowy widok dla części użytkowników (którzy będą go testować).

`resolve`

Używany do pobrania potrzebnych dla danego komponentu danych, przed aktywowaniem danej ścieżki.

`canLoad`

Używany do sprawdzenia, czy dany moduł (lazy loaded) może zostać załadowany.

## Tworzenie Guard'ów

Aby utworzyć guard możemy skorzystać z Angular CLI:

**ng generate guard <nazwa>**

Po wykonaniu komendy dostaniemy możliwość wyboru, który guard chcemy utworzyć:

```
? Which type of guard would you like to create? (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
>(*) CanActivate
  ( ) CanActivateChild
  ( ) CanDeactivate
  ( ) CanLoad
  ( ) CanMatch
```

Do tego od Angular 15 mamy możliwość implementacji guard'ów w postaci funkcji, zamiast klas.