

Kampuotas 19

App
Components
Pipes
Services
Routing

Kas yra Angular?

<https://angular.dev/overview>

- Modernus, TypeScript pagrįstas karkasas (framework) internetinių aplikacijų kūrimui.
- Sukurta ir prižiūrima Google.
- Pateikia integruotus sprendimus maršrutizavimui (routing), būsenos valdymui (state management), formoms ir priklausomybių įvedimui (dependency injection).

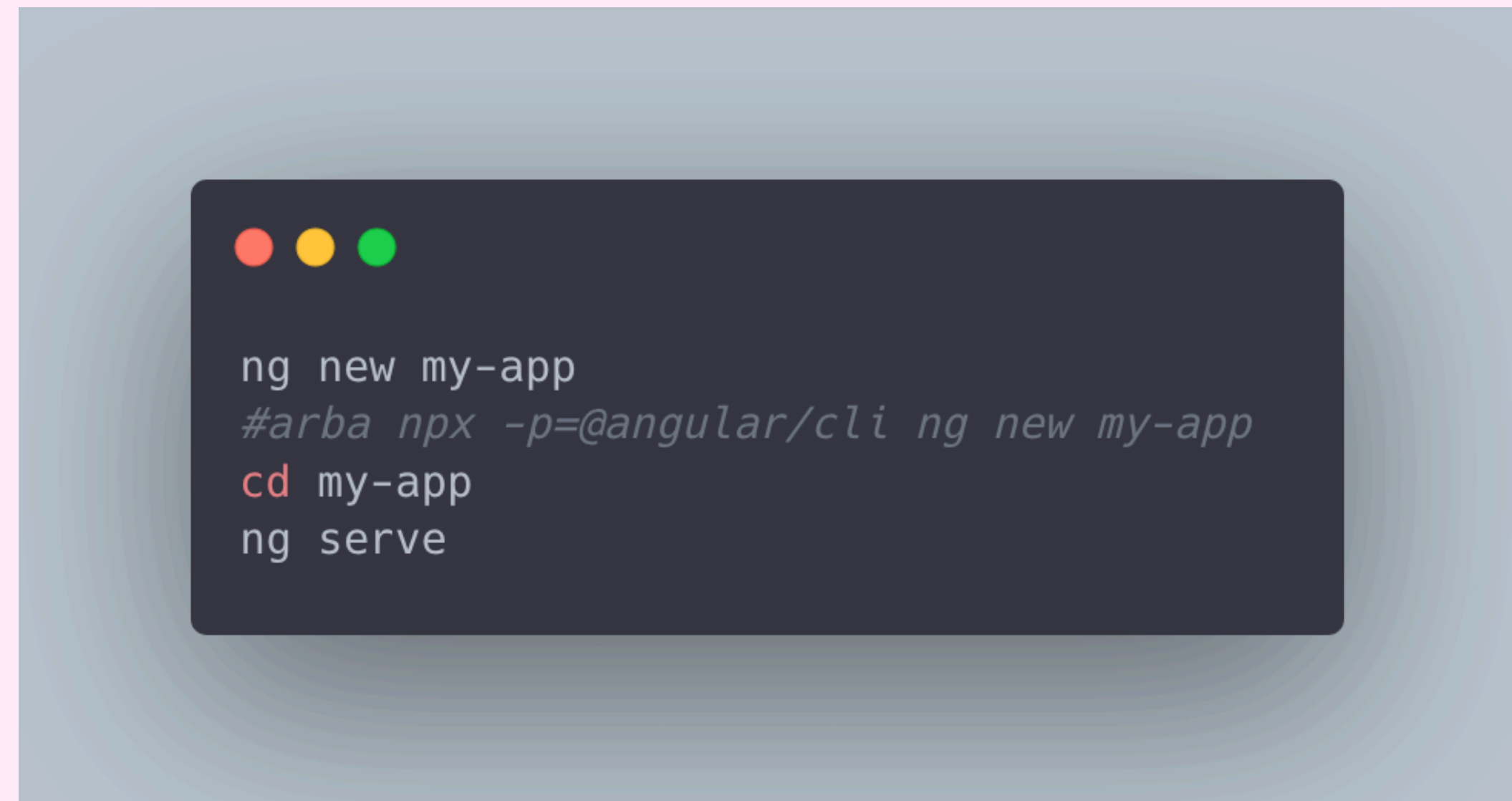
Angular CLI ir diegimas

<https://angular.dev/tools/cli>

- Įrašomas Angular CLI (`npm install -g @angular/cli`) pagalba.



```
npm install -g @angular/cli  
#arba ne
```



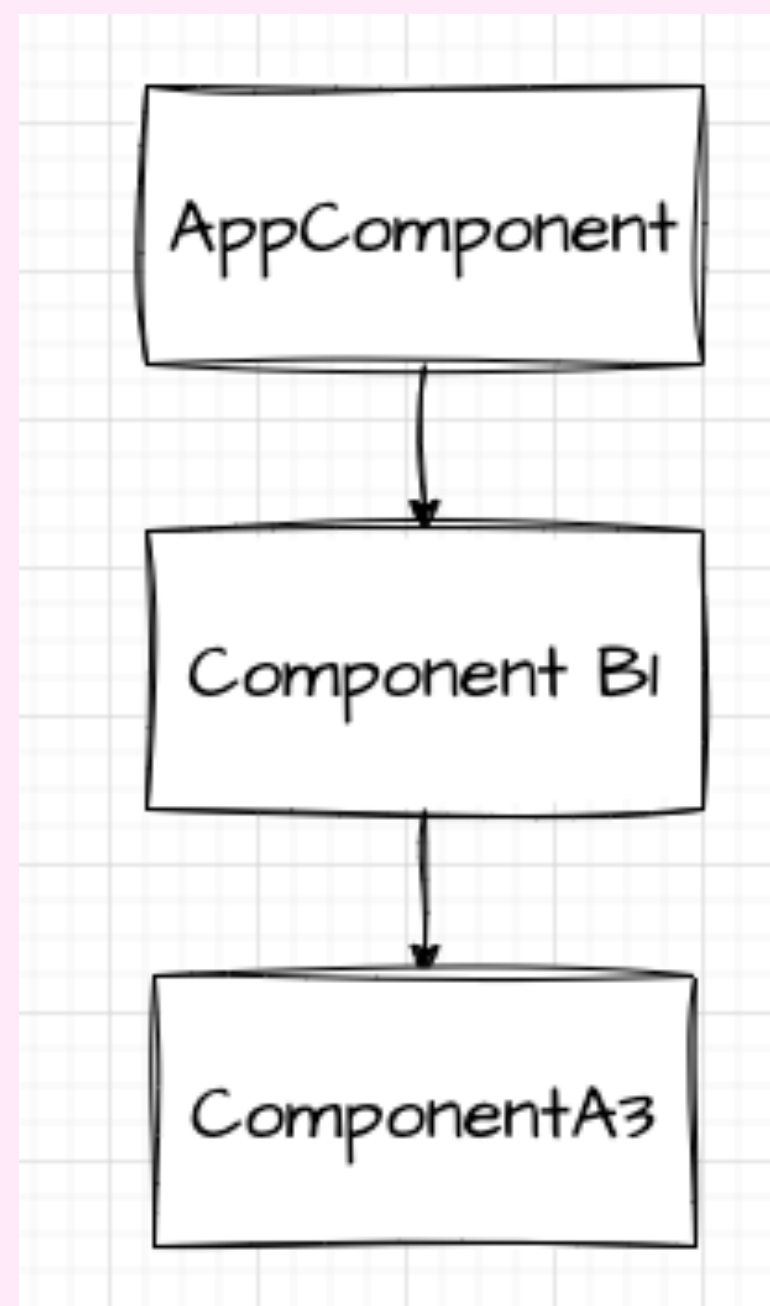
```
ng new my-app  
#arba npx -p=@angular/cli ng new my-app  
cd my-app  
ng serve
```

- Projekto struktūros apžvalga:
 - src/app/ – Pagrindinė aplikacijos logika
 - main.ts – Aplikacijos paleidimo įėjimo taškas

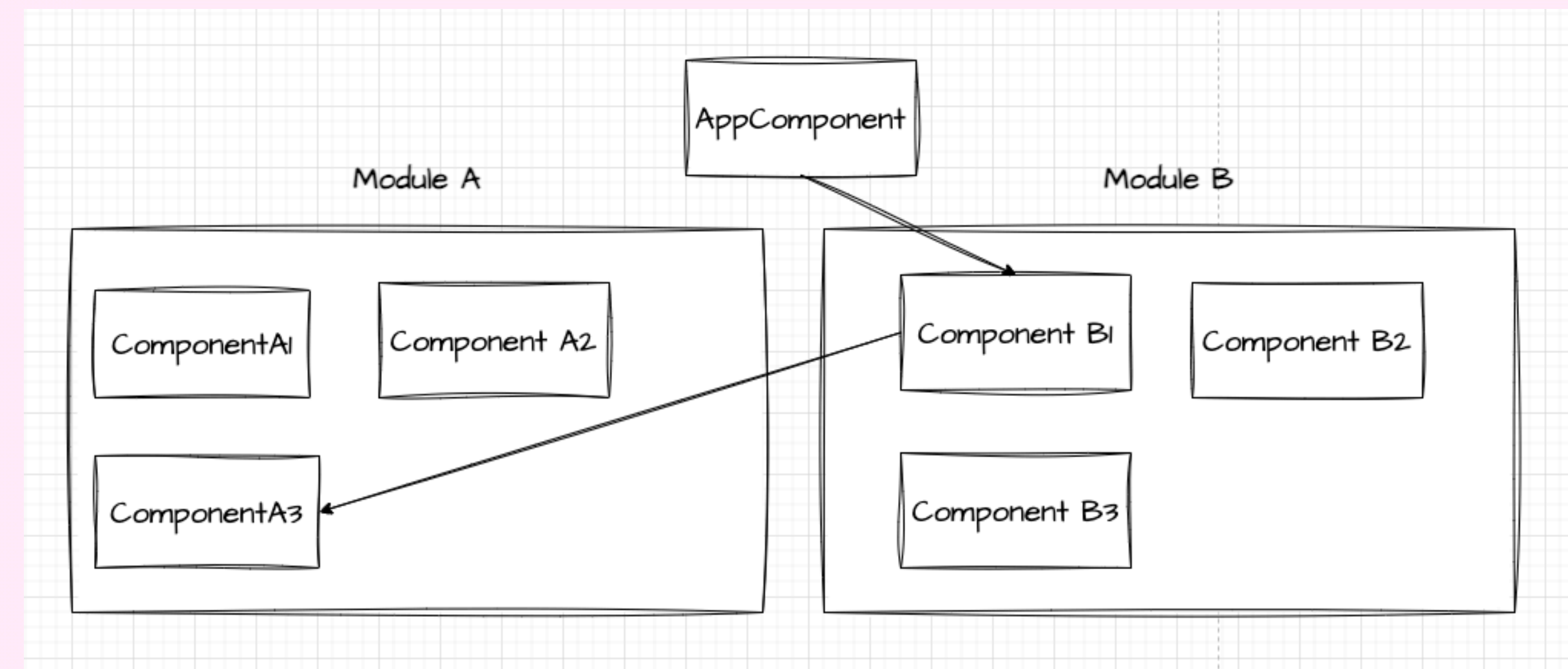
Standalone API vs. Moduliai

<https://angular.dev/guide/ngmodules/overview>

Standalone API (Nauja nuo Angular 14+)



Tradicinis NgModule metodas



Standalone API vs. Moduliai

<https://angular.dev/guide/ngmodules/overview>

Standalone API (Nauja nuo Angular 14+)

```
import { bootstrapApplication } from '@angular/platform-browser';
import { AppComponent } from './app/app.component';

bootstrapApplication(AppComponent);
```

Tradicinis NgModule metodas

```
import { NgModule } from '@angular/core';
@NgModule({
  declarations: [AppComponent],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Angular komponentai

<https://angular.dev/guide/components>

- Kas yra komponentas?
 - Pagrindinis Angular aplikacijos blokas
 - Susideda iš TypeScript (logika), HTML (vaizdas) ir CSS (stiliai)

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains TypeScript code for an Angular component.

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-example',
  template: '<h1>Sveiki atvykę į Angular!</h1>'
})
export class ExampleComponent {}
```

Komponentų komunikacija

<https://angular.dev/guide/components/inputs>

<https://angular.dev/guide/components/outputs>

- @Input() - Duomenų perdavimas iš tėvinio į vaikų komponentą

```
@Component({ selector: 'app-vaikas', template: `<p>{{ message }}</p>` })  
export class ChildComponent {  
  @Input() message!: string;  
}
```

- @Output() - Įvykių perdavimas iš vaikų į tėvinį komponentą

```
@Component({ selector: 'app-tevas', template: `<button (click)="sendData()">Siųsti</button>` })  
export class ChildComponent {  
  @Output() event = new EventEmitter<string>();  
  sendData() { this.event.emit('Labas vakaras!'); }  
}
```


Angular Control Flow (valdymo srautas)

<https://angular.dev/guide/directives/structural-directives>

<https://angular.dev/guide/templates/control-flow>



```
@if (items.length > 0) {  
  @for (item of items) {  
    <li>{{ item }}</li>  
  }  
} @else {  
  <p>Nėra elementų.</p>  
}
```


Task #1

Jūs esate **jaunesnysis programuotojas** startuolyje, kuriantis **asmeninį užduočių sekiklį**. Jūsų tikslas – sukurti Angular programėlę, kurioje vartotojai galėtų:

- Pridėti naują užduotį į sąrašą.
- Peržiūrėti užduočių sąrašą, atvaizduojamą dinamiškai.
- **Rodyti pranešimą**, kai nėra jokių užduočių.