

Angular Material komponente

Dr. Mlađan Jovanović
mjovanovic@singidunum.ac.rs

Sadržaj

- ◆ Input
- ◆ Select
- ◆ Checkbox
- ◆ Forma (form)
 - Pristup podacima, korišćenje podataka, validacija
- ◆ Kalendar (DatePicker)
- ◆ Sidebar
- ◆ Toolbar
- ◆ Lista (list)
- ◆ Tab
- ◆ Kartica (card)
- ◆ Tabela
 - Prikaz, straničenje, sortiranje, filtriranje
- ◆ Dijalog (dialog)
 - Prosleđivanje, prihvata, prikaz podataka
- ◆ Napomena: Kreiranje i korišćenje modula za registraciju AM komponenti (material.module.ts) je opisano u prethodnoj lekciji

Input komponenta 1/3

Korišćenje:

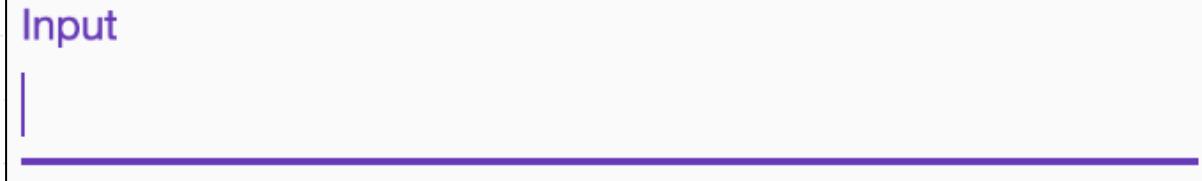
```
import {MatInputModule} from '@angular/material';
imports i exports atributi @NgModul dekoratora
(material.module.ts)
```

AM varijanta standardnih HTML tagova formi za unos podataka korisnika (<input> i <text area>)

Koriste se isključivo kao deo elementa forme !

input – jednolinijsko polje za unos teksta

```
<mat-form-field>
|   <input matInput placeholder="Input">
</mat-form-field>
```



Input

Input komponenta 2/3

textarea – višelinijsko polje za unos teksta

```
<mat-form-field>
|   <textarea matInput placeholder="Textarea"></textarea>
</mat-form-field>
```



Jednolinijsko polje može biti specifičnog formata sadržaja:

date
email
password
text
number

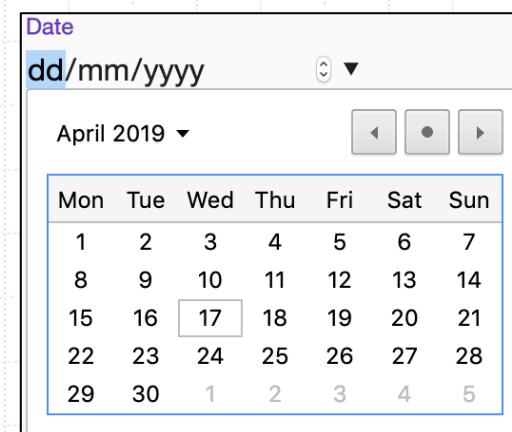
Input komponenta 3/3

```
<mat-form-field>
|   <input matInput type="email" placeholder="Email">
</mat-form-field>
```

Email

mladja@acm.org|

```
<mat-form-field>
|   <input matInput type="date" placeholder="Date">
</mat-form-field>
```



Input komponenta

```
<mat-form-field>
| <input matInput type="password" placeholder="Password">
</mat-form-field>
```

Password

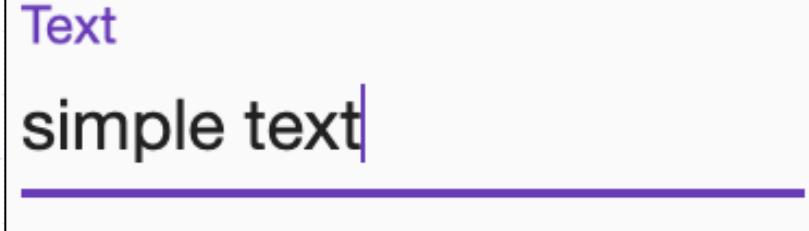
••••••



```
<mat-form-field>
| <input matInput type="text" placeholder="Text">
</mat-form-field>
```

Text

simple text



Select

Korišćenje:

```
import {MatSelectModule} from '@angular/material';
imports i exports atributi @NgModule dekoratora
(material.module.ts)
```

Element koji omogućava izbor vrednosti iz skupa opcija

Koriste se isključivo kao deo forme !

AM implementacija HTML select elementa

```
<mat-select placeHolder="placeholder_text">
<mat-option value="value1" [selected]>Value1</mat-option >
<mat-option value="value1" [disabled]>Value1</mat-option >
```

...

```
<mat-select>
```

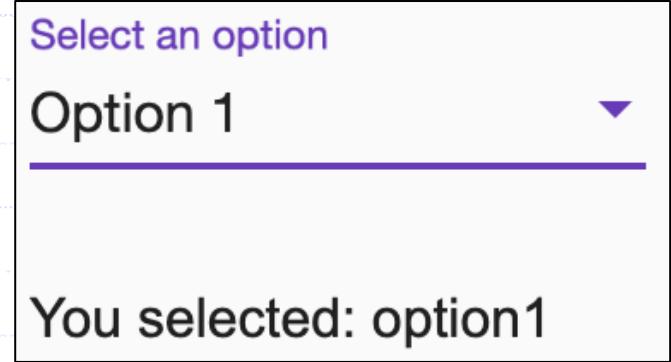
atribut value predstavlja vrednost izabrane opcije (može se koristiti is kontrolera, tj. pozadinskog TS koda komponente)

Opcije se mogu unapred izabrati (select) ili onemogućiti (disabled)

Select

Obostrani (two-way) data binding
Templejt (html) korisničke komponente:

```
<mat-form-field>
  <mat-select [(value)]="selected" placeholder="Select an option">
    <mat-option>None</mat-option>
    <mat-option value="option1">Option 1</mat-option>
    <mat-option value="option2">Option 2</mat-option>
  </mat-select>
</mat-form-field>
<p>You selected: {{selected}}</p>
```



Kontroler korisničke komponente (TS):

```
export class SelectValueBindingExample {
  selected : string = 'option2';
}
```

Select

Dinamičko kreiranje opcija iz kontrolera
Templejt (html) korisničke komponente:

```
<mat-form-field>
  <mat-select [(ngModel)]="selectedCar" placeholder="Select car:>
    <mat-option *ngFor="let car of cars" [value]="car.value">
      {{car.viewValue}}
    </mat-option>
  </mat-select>
</mat-form-field>
<p> Selected car: {{selectedCar}}</p>
```

Select car:

Saab

Selected car: saab

Kontroler korisničke komponente (pozadinski kod):

```
export interface Car {
  value: string;
  viewValue: string;
}
```

```
export class SelectFormExample {
  selectedCar: string;
  cars: Car[] = [
    {value: 'volvo', viewValue: 'Volvo'},
    {value: 'saab', viewValue: 'Saab'},
    {value: 'mercedes', viewValue: 'Mercedes'}];
}
```

CheckBox

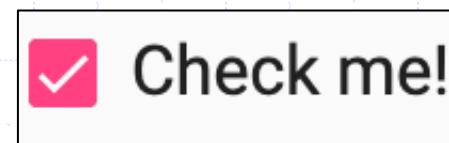
Korišćenje:

import {MatCheckboxModule} from '@angular/material';

imports i exports atributi @NgModul dekoratora
(material.module.ts)

AM alternativa HTML elementu (<input type="checkbox">)
sa stilizovanim izgledom i animacijama

```
<mat-checkbox>Check me!</mat-checkbox>
```



Stanja: **checked, indeterminate, disabled**

CheckBox – prve tri kontrole daju primer za obostrani data binding, dok poslednja kontrola predstavlja primer property binding-a



Templejt (html) korisničke komponente:

```
<mat-checkbox [(ngModel)]="checked">Checked</mat-checkbox>
<mat-checkbox [(ngModel)]="indeterminate">Indeterminate</mat-checkbox>
<mat-checkbox [(ngModel)]="disabled">Disabled</mat-checkbox>
<br>
<mat-checkbox
  [checked]="checked"
  [indeterminate]="indeterminate"
  [disabled]="disabled">
  I'm a checkbox
</mat-checkbox>
```

Kontroler korisničke komponente (pozadinski kod):

```
export class CheckboxConfigurableExample {
  checked = false;
  indeterminate = false;
  disabled = false;
}
```

CheckBox – rezultat primera

Checked Indeterminate Disabled

I'm a checkbox

Checked Indeterminate Disabled

I'm a checkbox

Checked Indeterminate Disabled

I'm a checkbox

Forma (form) i polje forme (mat-form-field)

Korišćenje:

```
import {FormsModule} from '@angular/forms';
import {MatFormFieldModule} from '@angular/material';
imports i exports atributa @NgModule dekoratora
(material.module.ts)
```

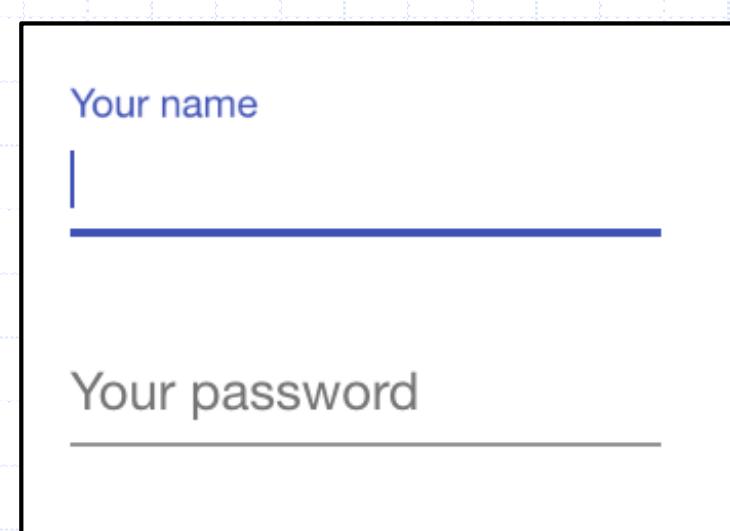
Složena kontrola korisničkog interfejsa za prihvat i obradu podataka korisnika, tzv. *data-entry* zadaci:

1. Prihvati i validacija ulaza korisnika u view delu (html templejt)
2. Kreiranje modela podataka forme (form model) u pozadini (kontroler)
3. Manipulacija modelom i praćenje promena podataka (kontroler)

Forma (form) i polje forme (mat-form-field) - struktura

```
<form fxLayout="column" fxLayoutAlign="center center">
<mat-form-field>
  <input type="email" matInput placeholder="Your name">
</mat-form-field>
<mat-form-field>
  <input type="password" matInput placeholder="Your password">
</mat-form-field>
</form>
```

form je kontejnerski tag u okviru kog se input elementi smještaju u mat-form-field tag omotač (wrapper)



Your name

Your password

Forma (form) i polje forme (mat-form-field) – pristup podacima forme

```
<form fxLayout="column" fxLayoutGap="10px" fxLayoutAlign="center center"  
#f="ngForm" (ngSubmit)="onSubmit(f)">  
  <mat-form-field>  
    <input type="email" matInput placeholder="Your name"  
          ngModel name="email">  
  </mat-form-field>  
  <mat-form-field>  
    <input type="password" matInput placeholder="Your password"  
          ngModel name="password">  
  </mat-form-field>  
  <button type="submit" mat-raised-button color="primary">Submit</button>  
</form>
```

Forma se kao promenjiva tipa objekta forme (`#f = ngForm`) prosleđuje metodi osluškivača na događaj slanja podataka forme (`ngSubmit`)
Pojedinačnim poljima forme se može pristupiti tako što se ona najpre proglase delom objekta forme koji se kreira u pozadini, tzv. model forme (`ngModel`) i zatim dodeli naziv polju (`name` atribut)

Forma (form) i polje forme (mat-form-field) – korišćenje podataka forme

```
onSubmit(form: NgForm) {  
    var email = form.value.email;  
    var password = form.value.password;  
    console.log(form);  
}
```

Poljima modela se pristupa na osnovu naziva, tj. vrednosti name atributa templejta

```
name: "email"  
path: (...)  
pending: (...)  
pristine: (...)  
status: (...)  
statusChanges: (...)  
touched: (...)  
untouched: (...)  
update: EventEmitter {_isScalar: false}  
valid: (...)  
validator: (...)  
value: "mladjan.jovanovic@gmail.com"
```

Forme - validacija

- ◆ Validacija AM formi se vrši u templejtu komponente, tj. pre nego se podaci proslede kontroleru komponente
- ◆ Definiše se lokalna promenjiva templejta (#emailInput) u kojoj se čuva sadržaj polja za unos email adrese
- ◆ Poruke o grešci se prikazuju u <mat-error> tagu pomoći ngIf

```
<mat-form-field>
  <input type="email" matInput placeholder="Your name"
    ngModel name="email" email required #emailInput="ngModel">
    <mat-error *ngIf="emailInput.hasError('required')">
      | Field should not be left empty.
    </mat-error>
    <mat-error *ngIf="!emailInput.hasError('required')">
      | Please provide correct email format.
    </mat-error>
</mat-form-field>
```

Forme - validacija

```
<mat-form-field>
  <input type="email" matInput placeholder="Your name"
    ngModel name="email" email required #emailInput="ngModel">
  <mat-error *ngIf="emailInput.hasError('required')">
    | Field should not be left empty.
  </mat-error>
  <mat-error *ngIf="!emailInput.hasError('required')">
    | Please provide correct email format.
  </mat-error>
</mat-form-field>
```

Your name *

mladjan

Please use the correct email format.

Your password *

.....

Please enter minimum 6 characters.

11 / 6

Submit

Kalendar (DatePicker)

Korišćenje:

```
import {MatDatepickerModule} from '@angular/material';  
imports i exports atributa @NgModule dekoratora  
(material.module.ts)
```

Često se koristi kao deo forme (na primer, registracija korisnika)

Vrednost se može uneti ručno ili izabrati iz kalendara

Kontrola se sastoji iz tri dela:

1. tekstualnog polja za unos datuma ([input](#))
2. kalendara ([mat-datepicker](#)) i
3. ikone koja prikazuje kalendar ([mat-datepicker-toggle](#))

Kalendar (DatePicker)

```
<mat-form-field>
  <input matInput placeholder="Your date of birth."
    [matDatepicker]="picker" ngModel name="birthDate" required>
  <mat-datepicker-toggle matSuffix [for]="picker">
  </mat-datepicker-toggle>
  <mat-datepicker #picker></mat-datepicker>
</mat-form-field>
```

Tekstualno polje i kalendar se povezuju pomoći **matDatePicker** property-ja.

Definiše se lokalna promenjiva (**picker**) koja predstavlja kalendar i prosleđuje kao parametar polju za prikaz datuma i ikoni.

Korišćenjem **matSuffix** direktive ikona se prikazuje posle tekstualnog datuma (kao sufiks).

Kalendar (DatePicker)

```
<mat-form-field>
  <input matInput placeholder="Your date of birth."
         [matDatepicker]="picker" ngModel name="birthDate" required>
  <mat-datepicker-toggle matSuffix [for]="picker">
  </mat-datepicker-toggle>
  <mat-datepicker #picker></mat-datepicker>
</mat-form-field>
```

Your name *

mladjan.jovanovic@gmail.com

Your password *

.....

Please enter minimum 6 characters.

8 / 6

Your date of birth. *

10/11/2018



Submit

Sidebar (vertikalna linija alata)

Korišćenje:

import {MatSidenavModule} from '@angular/material';

imports i exports atributa @NgModule dekoratora
(material.module.ts)

Vertikalni element kontejner za zaglavla, nazine ili akcije

Akcije mogu biti predstavljene tekstualno (dugmad ili linkovi) ili kao simboli (ikone)

Može se koristiti i kao *collapsible* kontrola koja se nalazi uz levu ivicu prikaza i otvara/zatvara po potrebi (mobilni interfejsi)

U praksi se kombinuje sa horizontalnom linijom (toolbar) tako da se izgled dinamički prilagođava dimenzijama:

veće dimenziye (tj. širina) => prikazuje se toolbar

manje dimenziye (tj. širina) => prikazuje se sidenav (hamburger meni kod mobilnih telefona)

Opisani mehanizam je poznat pod nazivom *responsive layout*

Sidebar (vertikalna linija alata)

Kontrola se sastoji iz tri dela:

1. koren i tj. kontejnerski element (`mat-sidenav-container`)
2. sidenav deo koji kontroliše prikaz, tj. vidljivost sadržaja (`mat-sidenav`)
3. sadržaj koji se prikazuje (`mat-sidenav-content`)

Često se koristi za definisanje globalne navigacije aplikacije

```
<mat-sidenav-container>
  <mat-sidenav opened="true">
    <p>Sidenav</p>
  </mat-sidenav>
  <mat-sidenav-content>
    <router-outlet></router-outlet>
  </mat-sidenav-content>
</mat-sidenav-container>
```

Toolbar (horizontalna linija alata)

Korišćenje:

```
import {MatToolbarModule} from '@angular/material';
```

imports i exports atributa @NgModule dekoratora
(material.module.ts)

Horizontalni element kontejner za zaglavla, nazine ili akcije
Akcije mogu biti predstavljene tekstualno (dugmad ili linkovi) ili kao simboli (ikone)

Može se koristiti samostalno kao horizontalna linija alatki ili linkova koja se prikazuje u zaglavlu stranice

U praksi se često kombinuje sa horizontalnom linijom za alate (*sidebar*)

Toolbar (horizontalna linija alata)

Koristi se kao kontejner (`mat-toolbar`) u okviru kojeg se dodaju AM ili HTML elementi

Kombinacija za vertikalnom linijom alata je prikazana ispod
Toolbar kao element se smešta u sadržaj vertikalne linije alata
(`mat-sidenav-content`)

```
<mat-sidenav-content>
  <mat-toolbar color="primary">
    <div><button (click)="sidenav.toggle()">Toggle sidenav</button></div>
    <div>Logo</div>
    <div>
      <a routerLink="/signup">Signup</a>
      <a routerLink="/login">Login</a>
      <a routerLink="/training">Training</a>
    </div>
  </mat-toolbar>
  <router-outlet></router-outlet>
</mat-sidenav-content>
```

```
<mat-sidenav-container>
  <mat-sidenav #sidenav opened="true">
    <p>Sidenav</p>
  </mat-sidenav>
  <mat-sidenav-content>
    <mat-toolbar color="primary">
      <div>
        <button (click)="sidenav.toggle()">
          <mat-icon>menu</mat-icon>
        </button>
      </div>
      <div>
        <a routerLink="/">Logo</a>
      </div>
      <div>
        <a routerLink="/signup">Signup</a>
        <a routerLink="/login">Login</a>
        <a routerLink="/training">Training</a>
      </div>
    </mat-toolbar>
  </mat-sidenav-content>
</mat-sidenav-container>
```

List (lista)

Korišćenje:

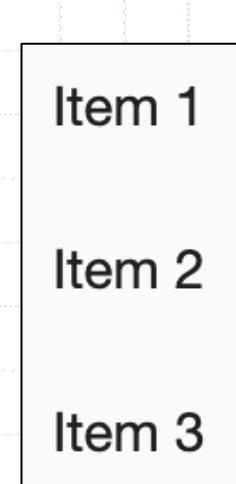
```
import {MatListModule} from '@angular/material';
imports i exports atributa @NgModule dekoratora
(material.module.ts)
```

AM element nabranjanja (analogno HTML list elementu)

Varijante:

1. Osnovna lista

```
<mat-list>
  <mat-list-item>Item 1</mat-list-item>
  <mat-list-item>Item 2</mat-list-item>
  <mat-list-item>Item 3</mat-list-item>
</mat-list>
```



List (lista)

2. Navigaciona lista (*navigation list*)

```
<mat-sidenav #sidenav opened="true">  
  <mat-nav-list>  
    <a mat-list-item routerLink="/signup">Signup</a>  
    <a mat-list-item routerLink="/login">Login</a>  
    <a mat-list-item routerLink="/training">Training</a>  
    <a mat-list-item routerLink="/">Logout</a>  
  </mat-nav-list>  
</mat-sidenav>
```

3. Akciona lista (*action list*)

```
<mat-action-list>  
  <button mat-list-item (click)="save()"> Save </button>  
  <button mat-list-item (click)="undo()"> Undo </button>  
</mat-action-list>
```

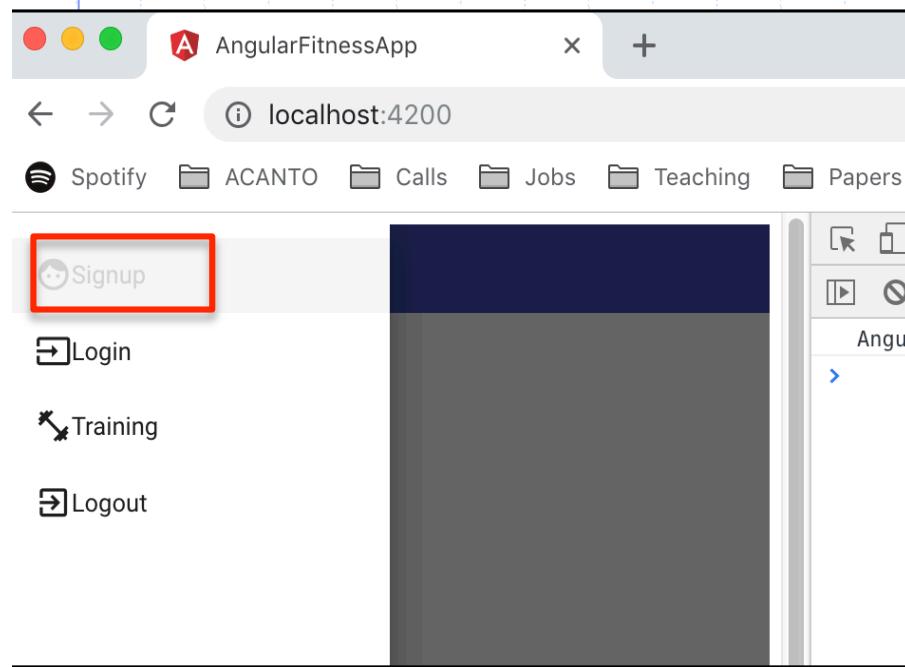
Navigaciona lista kao ikone

```
<mat-nav-list>
  <a mat-list-item routerLink="/signup">
    <mat-icon>face</mat-icon>
    Signup
  </a>
  <a mat-list-item routerLink="/login">
    <mat-icon>input</mat-icon>
    Login
  </a>
  <a mat-list-item routerLink="/training">
    <mat-icon>fitness_center</mat-icon>
    Training
  </a>
  <a mat-list-item routerLink="/">
    <mat-icon>exit_to_app</mat-icon>
    Logout
  </a>
</mat-nav-list>
```

Navigaciona lista kao ikone

```
<mat-nav-list>
    <a mat-list-item routerLink="/signup" (click)=" sidenav.close() " >
        <mat-icon>face</mat-icon>
        Signup
    </a>
    <a mat-list-item routerLink="/login" (click)=" sidenav.close() " >
        <mat-icon>input</mat-icon>
        Login
    </a>
    <a mat-list-item routerLink="/training" (click)=" sidenav.close() " >
        <mat-icon>fitness_center</mat-icon>
        Training
    </a>
    <a mat-list-item routerLink="/" (click)=" sidenav.close() " >
        <mat-icon>exit_to_app</mat-icon>
        Logout
    </a>
</mat-nav-list>
```

Navigaciona lista



AngularFitnessApp

localhost:4200/signup

Spotify ACANTO Calls Jobs Teaching Papers

Logo

Your name *

Your password *

Please enter minimum 6 characters. 0 / 6

Your date of birth. *

Submit

I accept the terms of conditions of use.

Tab (tabovi)

Korišćenje:

```
import {MatTabsModule} from '@angular/material';
```

imports i exports atributa @NgModule dekoratora
(material.module.ts)

Organizacija sadržaja kao odvojenih pogleda, gde je u datom trenutku vidljiv samo jedan pogled

Aktivan tab se može postaviti korišćenjem atributa `selectedIndex`

Sadržaj mat-tab taga može biti bilo koji element ili komponenta

```
<mat-tab-group>
  <mat-tab label="First"> Content 1 </mat-tab>
  <mat-tab label="Second"> Content 2 </mat-tab>
  <mat-tab label="Third"> Content 3 </mat-tab>
</mat-tab-group>
```

Tab – navigacija i sadržaj komponenti aplikacije

```
< <mat-tab-group>
    <mat-tab label="Current training">
        <app-current-training></app-current-training>
    </mat-tab>
    <mat-tab label="New training">
        <app-new-training></app-new-training>
    </mat-tab>
    <mat-tab label="Past training">
        <app-previous-training></app-previous-training>
    </mat-tab>
</mat-tab-group>
```

Card (kartica)

Korišćenje:

```
import {MatCardModule} from '@angular/material';
```

```
imports i exports atributa @NgModul dekoratora  
(material.module.ts)
```

Organizacija specifičnog tipa sadržaja po sekcijama kao zaglavlje, sadržaj, akcije i podnožje (footer)

<mat-card> je kontejnerski tag koji sadrži elemente:

<mat-card-title> Naslov kartice

<mat-card-content> Primarni sadržaj kartice (tekst, simboli)

<mat-card-actions> Akcije kao dugmad ispod sadržaja

<mat-card-footer> Podnožje kartice (footer) za eventualne dodatne informacije

Card (kartica)

```
<mat-card fxFlex="400px">
  <mat-card-title fxLayoutAlign="center">
    Start training
  </mat-card-title>
  <mat-card-content fxLayoutAlign="center">
    Choose exercise:
  </mat-card-content>
  <mat-card-actions fxLayoutAlign="center">
    <button type="submit" mat-button>Start</button>
  </mat-card-actions>
</mat-card>
```

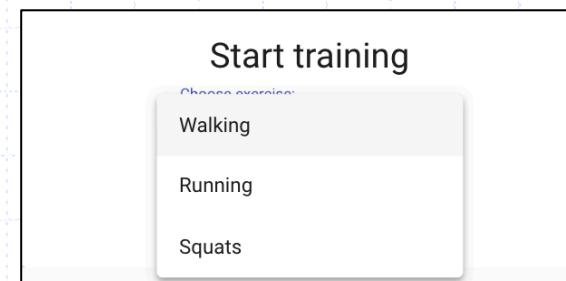
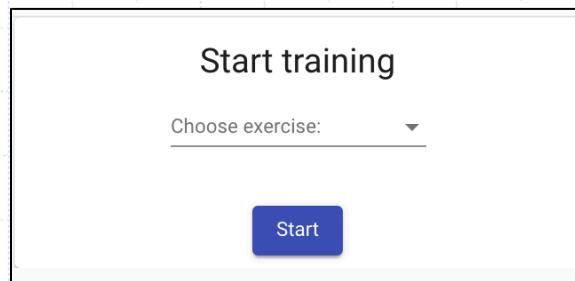
Naslov kartice

Sadržaj kartice

Akcije kartice

Card (kartica) – sadržaj može biti složeniji (select)

```
<mat-card-content fxLayoutAlign="center" fxLayoutGap="20x">
  <mat-form-field>
    <mat-select placeholder="Choose exercise:">
      <mat-option value="walking">
        Walking
      </mat-option>
      <mat-option value="running">
        Running
      </mat-option>
      <mat-option value="squats">
        Squats
      </mat-option>
    </mat-select>
  </mat-form-field>
</mat-card-content>
```



Tabele

Korišćenje:

```
import {MatTableModule} from '@angular/material';
imports i exports atributa @NgModule dekoratora
(material.module.ts)
```

AM funkcionalnost tabele sa dodatnim funkcijama sortiranja, filtriranja i straničenja kao zasebnim modulima

Obrazac za definisanje tabele je sledeći (po redosledu navođenja):

- 1. Glavni tag `<mat-table>`** čiji je glavni atribut *dataSource* (izvor podataka koji se prikazuju)
- 2. Kolone** su definisane u okviru `<ng-container>` taga koji sadrži tagove zaglavlja ćelija kolone (`<mat-header-cell>`) i vrednosti ćelija kolone (`<mat-cell>`)
- 3. Redovi:** `<mat-header-row>` koji definiše prvi red sa zaglavljima kolona i `<mat-row>` koji ispisuje vrednosti ćelija kolona.

Tabele – osnovni slučaj (templejt komponente)

```
<mat-table [dataSource]="dataSource">  
  <ng-container matColumnDef="position">  
    <mat-header-cell *matHeaderCellDef>No.</mat-header-cell>  
    <mat-cell *matCellDef="let element"> {{element.position}} </mat-cell>  
  </ng-container>  
  <ng-container matColumnDef="name">  
    <mat-header-cell *matHeaderCellDef>Name</mat-header-cell>  
    <mat-cell *matCellDef="let element"> {{element.name}} </mat-cell>  
  </ng-container>  
  <ng-container matColumnDef="weight">  
    <mat-header-cell *matHeaderCellDef>Weight</mat-header-cell>  
    <mat-cell *matCellDef="let element"> {{element.weight}} </mat-cell>  
  </ng-container>  
  <ng-container matColumnDef="symbol">  
    <mat-header-cell *matHeaderCellDef> Symbol </mat-header-cell>  
    <mat-cell *matCellDef="let element"> {{element.symbol}}</mat-cell>  
  </ng-container>  
  <mat-header-row *matHeaderRowDef="displayedColumns"></mat-header-row>  
  <mat-row *matRowDef="let row; columns: displayedColumns;"></mat-row>  
</mat-table>
```

Tabele – osnovni slučaj (kontroler komponente)

```
export interface PeriodicElement {  
    name: string;  
    position: number;  
    weight: number;  
    symbol: string;  
}  
  
const ELEMENT_DATA: PeriodicElement[] = [  
    {position: 1, name: 'Hydrogen', weight: 1.0079, symbol: 'H'},  
    {position: 2, name: 'Helium', weight: 4.0026, symbol: 'He'},  
    {position: 3, name: 'Lithium', weight: 6.941, symbol: 'Li'},  
    {position: 4, name: 'Beryllium', weight: 9.0122, symbol: 'Be'},  
    {position: 5, name: 'Boron', weight: 10.811, symbol: 'B'},  
    {position: 6, name: 'Carbon', weight: 12.0107, symbol: 'C'},
```

```
export class TableBasicExample {  
    displayedColumns: string[] = ['position', 'name', 'weight', 'symbol'];  
    dataSource = ELEMENT_DATA;  
}
```

Tabele – osnovni slučaj (izgled tabele)

No.	Name	Weight	Symbol
1	Hydrogen	1.0079	H
2	Helium	4.0026	He
3	Lithium	6.941	Li
4	Beryllium	9.0122	Be
5	Boron	10.811	B
6	Carbon	12.0107	C
7	Nitrogen	14.0067	N

Tabele – složeniji slučaj (templejt komponente)

```
<mat-table [dataSource]="exerciseSource">
  <ng-container matColumnDef="date">
    <mat-header-cell *matHeaderCellDef>Date</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.date | date }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="name">
    <mat-header-cell *matHeaderCellDef>Name</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.name }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="duration">
    <mat-header-cell *matHeaderCellDef>Duration</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.duration | number }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="calories">
    <mat-header-cell *matHeaderCellDef>Calories</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.calories | number }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="status">
    <mat-header-cell *matHeaderCellDef>Status</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.status }}</mat-cell>
  </ng-container>
  <mat-header-row *matHeaderRowDef="displayedColumns"></mat-header-row>
  <mat-row *matRowDef="let rows; columns: displayedColumns;">[]</mat-row>
</mat-table>
```

Tabele – složeniji slučaj (kontroler komponente)

```
import { Component, OnInit } from '@angular/core';
import { MatTableDataSource } from '@angular/material';
import { Exercise } from '../exercise.model'
import { TrainingService } from '../training.service';

@Component({
  selector: 'app-previous-training',
  templateUrl: './previous-training.component.html',
  styleUrls: ['./previous-training.component.css']
})
export class PreviousTrainingComponent implements OnInit {

  displayedColumns = ["date", "name", "duration", "calories", "status"];
  exerciseSource = new MatTableDataSource<Exercise>();

  constructor(private trainingService : TrainingService) { }

  ngOnInit() {
    this.exerciseSource.data = this.trainingService.getExercises();
  }
}
```

Tabele – složeniji slučaj (servis i struktura podataka)

```
export interface Exercise {  
    id : string;  
    name : string;  
    duration : number;  
    calories : number;  
    date? : Date;  
    status? : 'done' | 'cancelled' | null;  
}
```

```
import { Exercise } from "./exercise.model";  
  
export class TrainingService {  
    private exercises : Exercise [] = [  
        {id: 'running', name: 'Morning running', duration: 15, calories: 100},  
        {id: 'squats', name: 'Midday squats', duration: 5, calories: 70, date: new Date('2023-09-15T12:00:00Z')},  
        {id: 'walking', name: 'Afternoon walking', duration: 20, calories: 120},  
        {id: 'running', name: 'Evening running', duration: 15, calories: 100}  
    ]  
    getExercises() {  
        return this.exercises;  
    }  
}
```

Tabele – složeniji slučaj (izgled tabele)

New training	Past training	
Date	Name	Duration
Dec 2, 2018	Morning running	15
Dec 2, 2018	Midday squats	5
Dec 2, 2018	Afternoon walking	20
Dec 2, 2018	Evening running	15

Sortiranje tabele

Korišćenje:

```
import {MatSortModule} from '@angular/material';
```

```
imports i exports atributa @NgModule dekoratora (material.module.ts)
```

Dodavanje sortiranja kao funkcije se svodi na ubacivanje specifičnih direktiva u postojeću tabelu:

matSort direktiva u koreni element tabele (**mat-table**) – govori da se tabela kao celina može sortirati

mat-sort-header direktiva u zaglavljue kolone tabele (**mat-header-cell**)
– govori da se specifična kolona može sortirati

```
<ng-container matColumnDef="position">
  <th mat-header-cell *matHeaderCellDef mat-sort-header> Name </th>
  <td mat-cell *matCellDef="let element"> {{element.position}} </td>
</ng-container>
```

Sortiranje tabele – templejt komponente

```
<mat-table [dataSource]="exerciseSource" matSort>
  <ng-container matColumnDef="date">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Date</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.date | date }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="name">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Name</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.name }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="duration">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Duration</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.duration | number }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="calories">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Calories</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.calories | number }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="status">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Status</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.status }}</mat-cell>
  </ng-container>
  <mat-header-row *matHeaderRowDef="displayedColumns"></mat-header-row>
  <mat-row *matRowDef="let rows; columns: displayedColumns;"></mat-row>
</mat-table>
```

Sortiranje tabele – pozadinski kod

- ◆ Sortiranje predstavlja vrstu funkcije
- ◆ Da bi se pravilno implementirala potrebno je:
 1. Definisati objekat u pozadinskom kodu koji referencira elemente templejta, tj. podatke tabele
`sort: MatSort`
 2. Definisati samu funkciju sortiranja nad referenciranim podacima
`matTableDataSource.sort = sort`
 3. Potrebno je još obezbediti da se tabela može sortirati tek kada se podaci zaista učitaju i prikažu
implementacija interfejsa `AfterViewInit`

Sortiranje tabele – pozadinski kod

```
import { Component, OnInit, ViewChild, AfterViewInit } from '@angular/core';
import { MatTableDataSource, MatSort } from '@angular/material';
import { Exercise } from '../exercise.model'
import { TrainingService } from '../training.service';

@Component({
  selector: 'app-previous-training',
  templateUrl: './previous-training.component.html',
  styleUrls: ['./previous-training.component.css']
})
export class PreviousTrainingComponent implements OnInit, AfterViewInit {

  displayedColumns = ["date", "name", "duration", "calories", "status"];
  exerciseSource = new MatTableDataSource<Exercise>();

  @ViewChild(MatSort) sort : MatSort;

  constructor(private trainingService : TrainingService) { }

  ngOnInit() {
    this.exerciseSource.data = this.trainingService.getExercises();
  }

  ngAfterViewInit() {
    this.exerciseSource.sort = this.sort;
  }
}
```

Sortiranje tabele - izgled

Past training

Name	Duration	Calories ↓
Morning running	15	100
Evening running	15	100
Afternoon walking	20	80
Midday squats	5	70

Straničenje tabele

Korišćenje:

```
import {MatPaginatorModule} from '@angular/material';  
imports i exports atributa @NgModule dekoratora  
(material.module.ts)
```

Dodavanje straničenja kao funkcije se svodi na ubacivanje specifičnih direktiva u postojeću tabelu:

<mat-paginator> tag se dodaje u nastavku tabele sa atributima [pageSize] za veličinu stranice kao broj redova i [pageSizeOptions] kao opcije za promenu veličine stranice

Dodavanje elemenata u pozadinsku logiku (slično sortiranju)

MatPaginator

Straničenje tabele – templejt komponente

```
<mat-table [dataSource]="exerciseSource" matSort>
  <ng-container matColumnDef="date">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Date</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.date | date }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="name">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Name</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.name }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="duration">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Duration</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.duration | number }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="calories">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Calories</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.calories | number }}</mat-cell>
  </ng-container>
  <ng-container matColumnDef="status">
    <mat-header-cell *matHeaderCellDef mat-sort-header>Status</mat-header-cell>
    <mat-cell *matCellDef="let element">{{ element.status }}</mat-cell>
  </ng-container>
  <mat-header-row *matHeaderRowDef="displayedColumns"></mat-header-row>
  <mat-row *matRowDef="let rows; columns: displayedColumns;"></mat-row>
</mat-table>
<mat-paginator [pageSize]="1" [pageSizeOptions]="[2,4,6,8,10]">/mat-paginator>
```

Straničenje tabele – pozadinski kod

- ◆ Straničenje predstavlja vrstu funkcije
- ◆ Da bi se pravilno implementiralo potrebno je:
 1. Definisati objekat u pozadinskom kodu koji referencira elemente templejta, tj. podatke tabele paginator: `MatPaginator`
 2. Definisati samu funkciju straničenja nad referenciranim podacima
`matTableDataSource.paginator = paginator`
 3. Potrebno je još obezbediti da se tabela može straničiti tek kada se podaci zaista učitaju i prikažu implementacija interfejsa `AfterViewInit`

Straničenje tabele – pozadinski kod

```
import { Component, OnInit, ViewChild, AfterViewInit } from '@angular/core';
import { MatTableDataSource, MatSort, MatPaginator } from '@angular/material';
import { Exercise } from '../exercise.model'
import { TrainingService } from '../training.service';

@Component({
  selector: 'app-previous-training',
  templateUrl: './previous-training.component.html',
  styleUrls: ['./previous-training.component.css']
})
export class PreviousTrainingComponent implements OnInit, AfterViewInit {

  displayedColumns = ["date", "name", "duration", "calories", "status"];
  exerciseSource = new MatTableDataSource<Exercise>();

  @ViewChild(MatSort) sort : MatSort;
  @ViewChild(MatPaginator) paginator : MatPaginator;

  constructor(private trainingService : TrainingService) { }

  ngOnInit() {
    this.exerciseSource.data = this.trainingService.getExercises();
  }

  ngAfterViewInit() {
    this.exerciseSource.sort = this.sort;
    this.exerciseSource.paginator = this.paginator;
  }
}
```

Straničenje tabele - izgled

Duration

15

Calories

100

Status

done

5

70

done

Items per page:

2 ▾

1 - 2 of 4



Filtriranje tabele

- ◆ Filter ne postoji kao zasebna AM komponenta već se dodaje kao *input* polje forme (*mat-form-field*)
- ◆ Obično se dodaje pre, tj. iznad tabele templejt komponente:

```
<mat-form-field>
  <input matInput type="text" (keyup)="doFilter($event.target.value)" placeholder="Filter">
</mat-form-field>
```

pozadinski kod:

```
doFilter(filterValue : string) {
  this.exerciseSource.filter = filterValue.trim().toLowerCase();
}
```

AM datasource filter podrazumevano vrši konkatenaciju (spajanje) svih vrednosti reda tabele kao stringova, i potom traži vrednost filtera u rezultujućem stringu.

Filtriranje tablele

The screenshot shows a mobile application interface for managing training sessions. At the top, there is a blue header bar with the word "Logo" on the left and "Signup Login Training" on the right. Below the header, there are two tabs: "New training" and "Past training", with "Past training" being the active tab, indicated by a blue underline. A "Filter" input field is positioned below the tabs. The main content area displays a table with columns: Date, Name, Duration, Calories, and Status. One row is visible, showing a session from Dec 9, 2018, named "Morning running 15" with a duration of 100 and a status of "done". At the bottom, there is a pagination control with "Items per page: 1" and arrows for navigating through the results, which are "1 - 1 of 4".

Date	Name	Duration	Calories	Status
Dec 9, 2018	Morning running 15	100		done

Items per page: 1 ▾ 1 - 1 of 4 < >

Dijalog (dialog)

Korišćenje:

```
import {MatDialogModule} from '@angular/material';  
imports i exports atributa @NgModule dekoratora  
(material.module.ts)
```

Kreiranje AM stilizovanih modalnih dijaloga

Nekoliko direktiva stoji na raspolaganju za organizaciju sadržaja dijaloga:

mat-dialog-title atribut za naslov dijaloga, u heading elementu (`<h1>`, `<h2>`)

<mat-dialog-content> Glavni sadržaj dijaloga

<mat-dialog-actions> Kontejner za akcije, tj. dugmad

mat-dialog-close atribut koji se dodaje `<button>` elementu, čini da dugme zatvara dijalog sa opcionom vrednošću koja se može koristiti

Dijalog (dialog) – templejt

```
<section>
  <h1 mat-dialog-title>Stop training</h1>
  <mat-dialog-content>Are you sure?</mat-dialog-content>
  <mat-dialog-actions>
    <button mat-button [mat-dialog-close]="true">Yes</button>
    <button mat-button [mat-dialog-close]="false">No</button>
  </mat-dialog-actions>
</section>
```

Dijalog- pozadinski kod

- ◆ Dijalog se otvara i zatvara iz kontrolera
- ◆ Da bi se pravilno implementiralo potrebno je:
 1. Definisati servis u pozadinskom kodu koji upravlja dijalogom (otvaranje/zatvaranje)
`dialog: MatDialog`
 2. Kreirati servis kao parametar konstruktora
 3. Otvaranje i zatvaranje dijaloga pomoću servisa
`dialogRef = this.dialog.open()`
`dialogRef.afterClosed()`

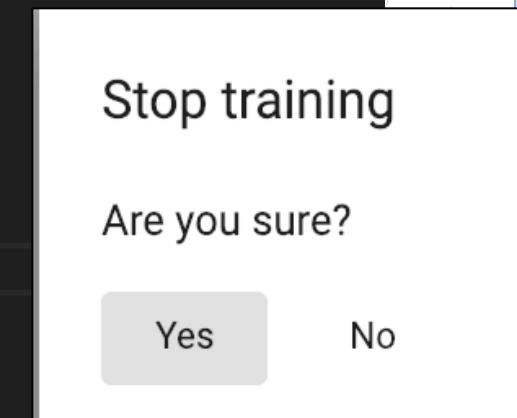
Registracija MatDialog servisa

- ◆ U cilju optimizacije performansi aplikacije, Angular izvršno okruženje zahteva od programera da komponente koje će se generisati dinamički (MatDialog) eksplicitno navede u vreme pisanja koda programa.
- ◆ Konkretno, komponente koje će se koristiti na ovaj način se navode kao element *entryComponents* niza *@Component* dekoratora korenog modula aplikacije (app.module.ts) kao ispod:

```
imports: [
  BrowserModule,
  MaterialModule,
  RoutingModule,
  BrowserAnimationsModule,
  FlexLayoutModule,
  FormsModule],
providers: [TrainingService],
bootstrap: [AppComponent],
entryComponents: [StopTrainingComponent])
```

Otvaranje dijaloga iz pozadinskog koda pozivajuće komponente

```
export class CurrentTrainingComponent implements OnInit {  
    progress = 0;  
    timer : any;  
    constructor(private dialog : MatDialog) { }  
    ngOnInit() {  
        this.timer = setInterval(() => {  
            this.progress += 5;  
            if (this.progress >= 100) {  
                clearInterval(this.timer);  
            }  
        }, 1000);  
    }  
    onStop() {  
        clearInterval(this.timer);  
        this.dialog.open(StopTrainingComponent);  
    }  
    onResume(){  
        this.timer = setInterval(() => {  
            this.progress += 5;  
            if (this.progress >= 100) {  
                clearInterval(this.timer);  
            }  
        }, 1000);  
    }  
}
```



Prosleđivanje podataka ka dijalogu

- ◆ Prosleđivanje podataka dijalogu se vrši prilikom otvaranja, tj. u pozivu metode *open*
- ◆ *Open* metoda može imati i drugi parametar tipa *any*
- ◆ Podaci koje prosleđujemo se kreiraju kao objekti programa predstavljeni u formatu *ime:vrednost*

```
onStop() {  
    clearInterval(this.timer);  
    this.dialog.open(StopTrainingComponent, {  
        data: {  
            progress: this.progress  
        }  
    });  
}
```

Prihvatanje prosleđenih podataka u dijalogu

- ◆ `MAT_DIALOG_DATA` je predefinisana simbolička konstanta koja nam omogućuje pristup podacima koje AM interno (u formi objekata u pozadini) čuva kada pozovemo *open* metodu (otvaranje dijaloga)
- ◆ `@Inject` direktiva se koristi za "ubacivanje" podataka referenciranih sa `MAT_DIALOG_DATA` u promenjivu koju mi definišemo (`receivedData`). Promenjivu definišemo da bismo ispisali preuzete podatke u templejtu

Prihvati prosleđenih podataka

```
import { Component, OnInit, Inject } from '@angular/core';
import { MAT_DIALOG_DATA} from "@angular/material";

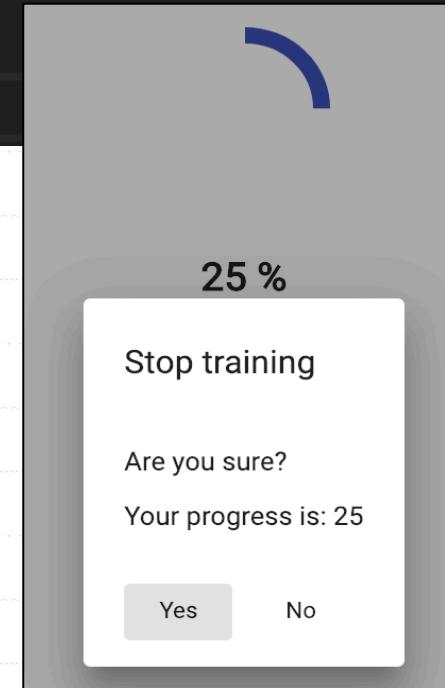
@Component({
  selector: 'app-stop-training',
  templateUrl: './stop-training.component.html',
  styleUrls: ['./stop-training.component.css']
})
export class StopTrainingComponent implements OnInit {

  constructor(@Inject(MAT_DIALOG_DATA) public receivedData: any) { }

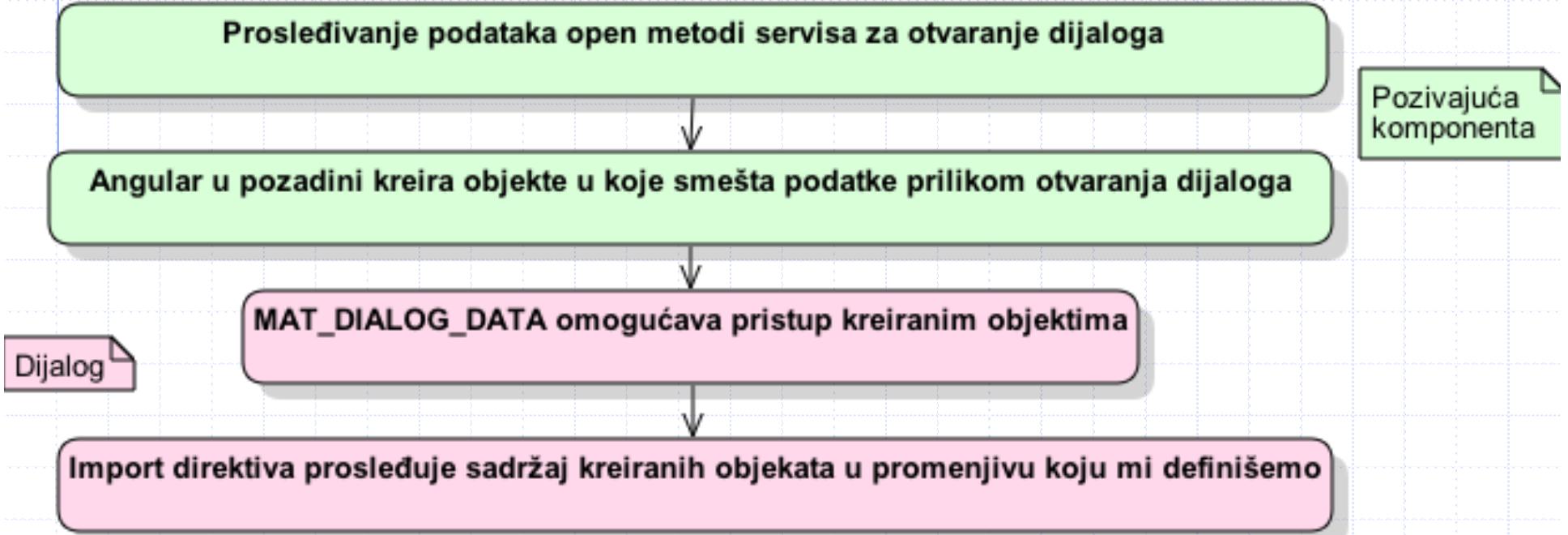
  ngOnInit() {
  }
}
```

Prikaz prosleđenih podataka

```
<section>
  <h1 mat-dialog-title>Stop training</h1>
  <mat-dialog-content>
    <p>Are you sure?</p>
    <p>Your progress is: {{ receivedData.progress }}</p>
  </mat-dialog-content>
  <mat-dialog-actions>
    <button mat-button [mat-dialog-close]="true">Yes</button>
    <button mat-button [mat-dialog-close]="false">No</button>
  </mat-dialog-actions>
</section>
```



Šema prosleđivanja podataka ka dijalogu



Sažetak

- ◆ Input
- ◆ Select
- ◆ Checkbox
- ◆ Forma (form)
 - Pristup podacima, korišćenje podataka, validacija
- ◆ Kalendar (DatePicker)
- ◆ Sidebar
- ◆ Toolbar
- ◆ Lista (list)
- ◆ Tab
- ◆ Kartica (card)
- ◆ Tabela
 - Prikaz, straničenje, sortiranje, filtriranje
- ◆ Dijalog (dialog)
 - Prosleđivanje, prihvatanje, prikaz podataka

Materijali

◆ Angular Material primer.pdf

- Korak-po-korak kreiranje horizontalnog prototipa aplikacije
- Horizontalni prototip = frontend

◆ angular-material-primer.zip

- Implementirani projekat

Linkovi ka dokumentaciji i primerima

- ◆ Input komponenta primeri:

<https://material.angular.io/components/input/examples>

- ◆ Select – obostrani data binding:

<https://stackblitz.com/angular/aeeyokplndq?file=app%2Fselect-value-binding-example.ts>

- ◆ Select – dinamičko kreiranje opcija:

<https://stackblitz.com/angular/mnamdxoxvay?file=app%2Fselect-overview-example.ts>

- ◆ Checkbox binding primeri:

<https://stackblitz.com/angular/kganmelael?file=app%2Fcheckbox-configurable-example.html>

- ◆ Tabela:

<https://material.angular.io/components/table/overview>

Angular Material komponente

Dr. Mlađan Jovanović
mjovanovic@singidunum.ac.rs