

Angular Material

[Version 12.0.4]

- Material provides UI and UX components form Angular.
- It is native to Angular.
- It is built-by Google Angular Team
- Provide Unified UX across devices.

Install Angular Material

- You have to add “Angular Material” to existing project.
- Run the following command in Angular Workspace
 - > **ng add @angular/material**
 - *Would you like to add Typography? Y
 - *Select a Pre-built Theme: Choose any theme
 - *Add Browser Animations Module? Y
- This will update following files
 - package.json
 - index.html

Roboto font

- angular.json

Configure Styles

- app.module.ts

- Setup theme file in “Styles.css”

@import '@angular/material/prebuilt-themes/indigo-pink.css';

Angular Components

How to Explore Components?

- Visit

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

- Select any component

- Go to “API” tab to know about the properties, methods and modules required for component

- Know your component module name

- Know the dependencies for required module

- Know the selector for component

- Know the properties and methods for component

Component: Form Field [Directive as Element]

- Module Required for Form Field
 - **MatFormFieldModule**
- Dependencies
 - **MatError**
 - **MatFormField**
 - **MatHint**
 - **MatPlaceholder**
 - **MatPrefix**
 - **MatSuffix**
 - **MatLabel**
- Get Selector Name from API
 - **mat-form-field**
- Get Properties from API
 - **@Input()** appearance:
MatFormFieldAppearance
[appearance]

- @Input() color: ThemePalette **[color]**
- Go to “Interfaces” category in API to get information about property values.
 - **MatFormFieldAppearance** = 'legacy' | 'standard' | 'fill' | 'outline'

Syntax:

```
<mat-form-field appearance="legacy |
standard | fill | outline">
  </mat-form-field>
```

Component: Input [Directive as Attribute]

- Module
 - **MatInputModule**
- Attribute Name
 - **matInput**

Syntax:

```
<input type="text" matInput>
```

Ex: Implementing Material Components

- Go to “app.module.ts”

```
import { MatFormFieldModule } from
 '@angular/material/form-field';
import { MatInputModule } from
 '@angular/material/input';
```

```
@NgModule({
  imports: [
    BrowserModule,
    FormsModule,
    BrowserAnimationsModule,
    MatFormFieldModule,
    MatInputModule
  ],
})
```

- Go to your “component.html”

```
<div class="container">
  <h2>Material Demo</h2>
  <mat-form-field appearance="legacy"
class="block-style" >
```

```
<mat-label>User Name</mat-label>
  <input [(ngModel)]="UserName"
type="text" matInput placeholder="Enter
User Name">
  <mat-hint>Name in Block Letters</mat-
hint>
  <mat-error>Name Required</mat-
error>
</mat-form-field>
<h3>Hello ! {{UserName}} </h3>
</div>
```

- **Component.css**

```
.block-style {
  width: 100%;
}
```

- **Component.ts**

```
UserName = ' ';
```

Material Datepicker

- **Modules**
 - MatDatepickerModule
 - MatNativeDateModule
- **Selectors**
 - mat-datepicker-toggle
 - mat-datepicker

Ex:

- Import the following modules in
“app.module.ts”

```
import { MatNativeDateModule } from  
'@angular/material/core';
```

```
import { MatFormFieldModule } from  
'@angular/material/form-field';
```

```
import { MatInputModule } from  
'@angular/material/input';
```

```
import { MatDatepickerModule } from  
'@angular/material/datepicker';
```

```
imports: [  
    BrowserModule,  
    FormsModule,  
    BrowserAnimationsModule,  
    MatFormFieldModule,  
    MatInputModule,  
    MatDatepickerModule,  
    MatNativeDateModule  
],
```

Component.html

```
<mat-form-field class="block-style mt-3">  
    <mat-label>Departure Date</mat-label>  
  
    <input type="text" matInput  
[matDatepicker]="picker" >  
    <mat-datepicker-toggle matSuffix  
[for]="picker" ></mat-datepicker-toggle>
```



```
<mat-datepicker #picker></mat-  
datepicker>  
</mat-form-field>
```

Material Icons

Ex:

App.module.ts

```
import { MatIconModule } from  
'@angular/material/icon';
```

```
imports: [
```

```
  BrowserModule,
```

```
  FormsModule,
```

```
  BrowserAnimationsModule,
```

```
  MatFormFieldModule,
```

```
  MatInputModule,
```

```
  MatDatepickerModule,
```

```
  MatNativeDateModule,
```

```
  MatIconModule
```

```
],  
  <button>  
    <mat-icon>home</mat-icon>  
    Home  
  </button>
```

Angular CDK (Component Dev Kit)

- It defines set of behaviour primitives for build UI component.
- It provides UX [User Experience]

Implementing Lazy Loading with Virtual Scroll CDK

- Lazy loading allows to load only the content required for situation.
- **<cdk-virtual-scroll-viewport>**
- ***cdkVirtualFor**

Ex:

App.module.ts

```
import { ScrollingModule } from
 '@angular/cdk/scrolling';

[
  ScrollingModule
]
```

Component.ts

```
products = [
  {Name: 'JBL Speaker', Photo:
    'assets/speaker.jpg'},
  {Name: 'EarPods', Photo:
    'assets/earpods.jpg'},
  .... Add records
]
```

Component.html

```
<div class="container">
  <h2>Lazy Loading - Virtual Scrolling</h2>
  <cdk-virtual-scroll-viewport itemSize="100"
class="view-port" >
    <div class="card" *cdkVirtualFor="let
product of products">
      <img [src]="product.Photo" height="50"
class="card-img-top">
      <div class="card-body">
        <h2>{{product.Name}}</h2>
      </div>
    </div>
  </cdk-virtual-scroll-viewport>
</div>
```


