

1. Creamos un nuevo proyecto, ejecutando el comando:

```
ng new jsonplaceholder --routing
```

2. Ingresamos a la url <https://material.angular.io/>

Get started

Copiamos el comando y lo ejecutamos en la terminal

```
npm install --save @angular/material @angular/cdk
```

Copiamos la instrucción y la pegamos en el archivo styles.css del proyecto:

```
@import "~@angular/material/prebuilt-themes/indigo-pink.css";
```

3. Revisamos la documentación de angular material  
<https://material.angular.io/components/button/api>

```
import { MatButtonModule } from '@angular/material/button';
```

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

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

4. Copiamos las instrucciones anteriores en **app.module.ts**, adicionalmente importamos el módulo http.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { HttpClientModule } from '@angular/common/http';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
```

```
@NgModule({
  declarations: [AppComponent],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    MatButtonModule,
    MatCardModule
  ],
})
```

5. Revisamos nuevamente la documentación de angular material, puntualmente en la sección de ejemplo de los botones:

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

Copiamos el estilo deseado para los botones que vamos a agregar a nuestra UI.

6. Pegamos las etiquetas en **app.component.html**

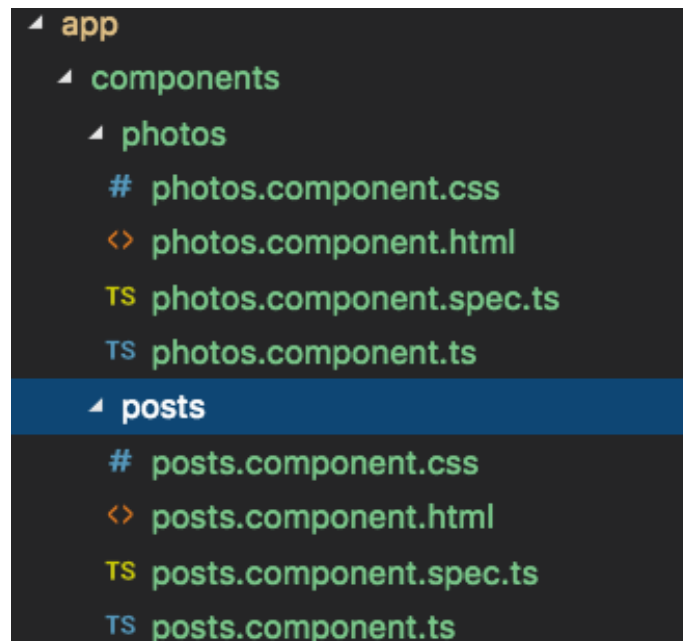
```
<button mat-stroked-button color="primary">Post</button>
<button mat-stroked-button color="accent">Photos</button>
<router-outlet></router-outlet>
```

7. Ejecutamos los siguientes comandos para crear los componentes requeridos:

```
ng generate component components/posts --module app app-routing
```

```
ng generate component components/photos --module app app-routing
```

8. Verificamos la correcta creación en la carpeta definida:



9. En **app-routing.module.ts**, importamos los nuevos componentes y definimos las rutas para los mismos.

```
import { NgModule } from "@angular/core";
import { Routes, RouterModule } from "@angular/router";
import { PostsComponent } from "../components/posts/posts.component";
import { PhotosComponent } from "../components/photos/photos.component";

const routes: Routes = [
  {
    path: "posts",
    component: PostsComponent
  },
  {
    path: "photos",
    component: PhotosComponent
  }
];
```

10. Agregamos el atributo routerLink en los botones de **app.component.html**

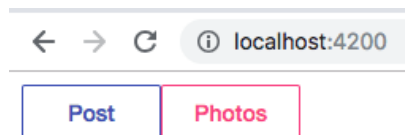
```
<button mat-stroked-button color="primary" routerLink="/posts">Post</button>
<button mat-stroked-button color="accent" routerLink="/photos">Photos</button>
<router-outlet></router-outlet>
```

11. Levantamos el servidor para verificar los cambios realizados:

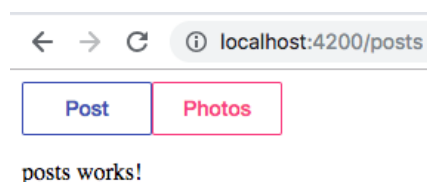
```
ng serve -o
```

12. Así debe aparecer en las rutas :

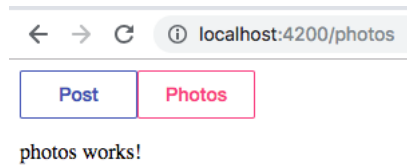
raíz "/"



"/posts"



“/photos”



13. Definimos algunos estilos para separar los botones.

```
# app.component.css x
1  button:first-child {
2    margin-right: 1em;
3  }
```



14. Definimos la lógica de **post.component.ts**

Importamos la clase HttpClient

```
import { Component, OnInit } from "@angular/core";
import { HttpClient } from "@angular/common/http";
```

Declaramos una variable para establecer la url del servicio web de posts y una variable para almacenar los registros que serán enviados a la vista.

```
API_URL = "https://jsonplaceholder.typicode.com/posts";
posts;
```

Realizamos inyección de dependencias en el método constructor para poder extender de la clase HttpClient

```
constructor(private http: HttpClient) {}
```

Creamos un método para consumir el servicio web por medio del método get de la clase HttpClient. Posteriormente lo asignamos a la variable posts.

```
getPosts() {  
  this.http.get(this.API_URL).subscribe(data => {  
    this.posts = data;  
  });  
}
```

Invocamos el método getPosts en la ejecución del componente.

```
ngOnInit() {  
  this.getPosts();  
}
```

15. En **posts.component.html** creamos la directiva para recorrer el array con los registros del servicio web y su posterior muestra de datos por medio de interpolaciones dentro de etiquetas

```
<h1>Posts</h1>  
<mat-card *ngFor="let post of posts">  
  <h3>{{post.title}}</h3>  
  <p>{{post.body}}</p>  
</mat-card>
```

Debemos obtener este resultado:

Post

Photos

## Posts

**sunt aut facere repellat provident occaecati excepturi optio reprehenderit**  
quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

**qui est esse**  
est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

**ea molestias quasi exercitationem repellat qui ipsa sit aut**  
et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

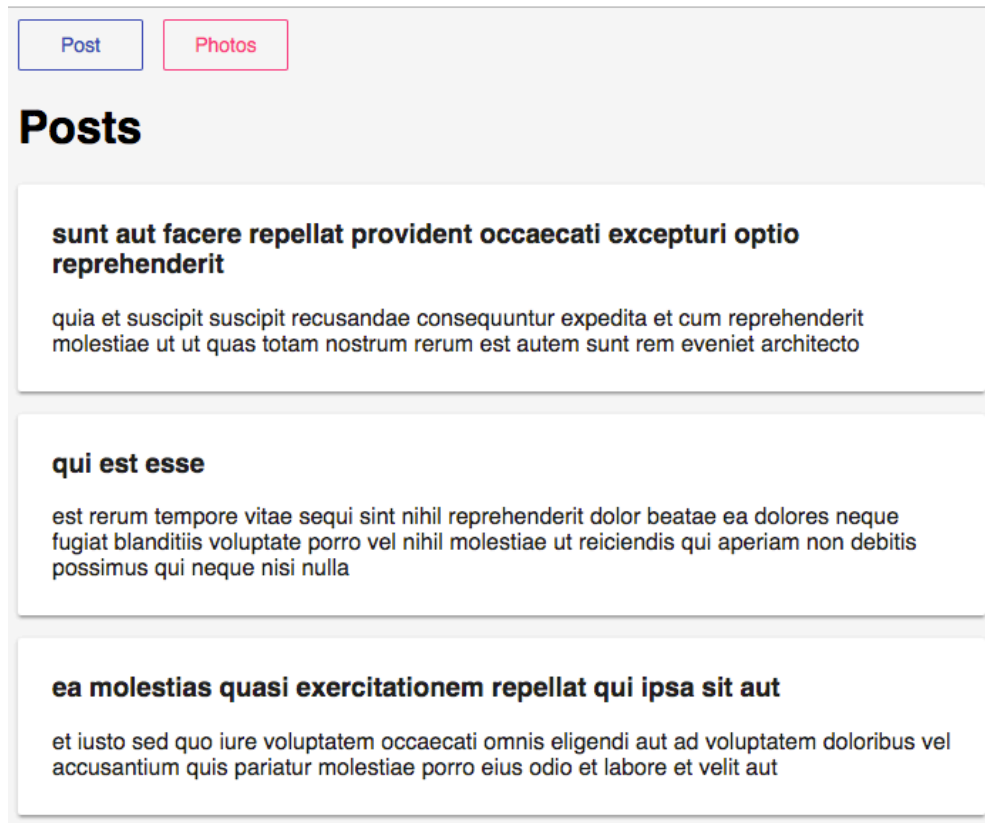
16. Mejoramos la tipografía del proyecto y definimos un color de fondo, agregando las siguientes propiedades en **styles.css**

```
body {  
  background-color: #f7f7f7;  
}  
  
* {  
  font-family: sans-serif;  
}
```

17. Separamos los componentes card en **posts.component.css**

```
posts.component.css x  
1  mat-card {  
2    margin-bottom: 1em;  
3  }
```

Debemos obtener este resultado:



## 18. Definimos la lógica de **photos.component.ts**

Importamos la clase HttpClient

```
import { Component, OnInit } from "@angular/core";  
import { HttpClient } from "@angular/common/http";
```

Declaramos una variable para establecer la url del servicio web de photos y una variable para almacenar los registros que serán enviados a la vista.

```
API_URL = "https://jsonplaceholder.typicode.com/photos";  
photos;
```

Realizamos inyección de dependencias en el método constructor para poder extender de la clase HttpClient

```
constructor(private http: HttpClient) {}
```

Creamos un método para consumir el servicio web por medio del método get de la clase HttpClient. Posteriormente lo asignamos a la variable photos.

```
getPhotos() {  
  this.http.get(this.API_URL).subscribe(data => {  
    this.photos = data;  
  });  
}
```

Invocamos el método getPhotos en la ejecución del componente.

```
ngOnInit() {  
  this.getPhotos();  
}
```

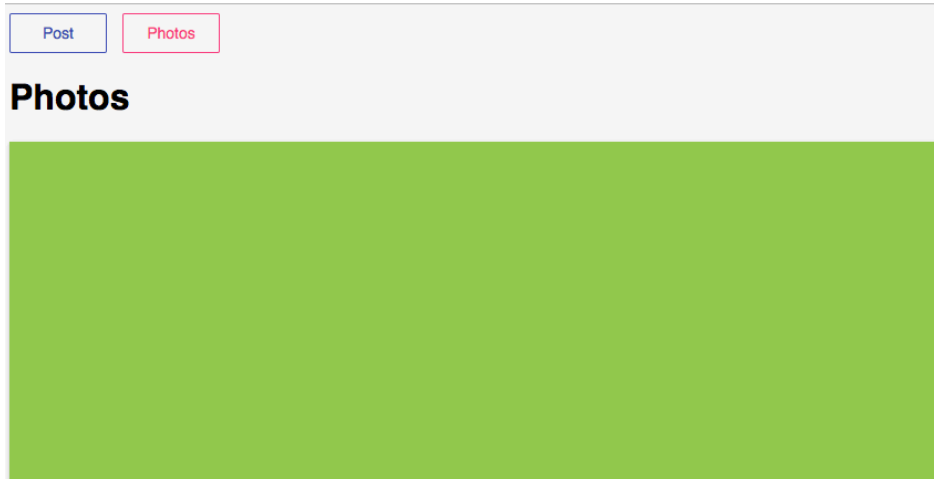
## 19. En **photos.component.html** creamos la directiva para recorrer el array con los registros del servicio web y su posterior muestra de datos por medio de interpolaciones dentro de etiquetas

```

<h1>Photos</h1>
<div class="container">
  <mat-card class="example-card" *ngFor="let photo of photos">
    
    <mat-card-content>
      <p>
        {{photo.title}}
      </p>
    </mat-card-content>
  </mat-card>
</div>

```

Debemos obtener este resultado:



20. Mejoramos la visualización de los datos por medio de css grid layout dentro de **photos.component.css**

```

.container {
  display: grid;
  grid-template: auto / repeat(4, 22vw);
  grid-gap: 1em;
  justify-content: space-around;
}

```



Debemos obtener este resultado:

Post

Photos

Photos

150 x 150

accusamus beatae ad facilis cum  
similique qui sunt

150 x 150

reprehenderit est deserunt velit  
ipsam

150 x 150

officia porro iure quia iusto qui ipsa ut  
modi

150 x 150

culpa odio esse rerum omnis  
laboriosam voluptate repudiandae

150 x 150

150 x 150

150 x 150

150 x 150