Angular Note

Install Aangular CLI

```
npm install -g @angular/cli
```

Create new project

```
ng new my-site
```

Start the website

```
ng serve
```

Important Config Files

package.json
tsconfig.json
angular.json

Important Files

index.html
main.ts

Creating New Component

```
ng generate component components/button
```

Add Fields in the component

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

@Component({
  selector: 'app-button',
  templateUrl: './button.component.html',
  styleUrls: ['./button.component.css']
})
export class ButtonComponent implements OnInit {
  @Input() text:string;
  @Input() color:string;

  constructor() { }

  ngOnInit(): void {
  }
}
```

In button component html, put this:

```
<input type="button" [ngStyle]="{'background-color':color}" value="{{text}}" (click)="onClick()" />
```

Add The Event Handler in button component.ts

```
onClick(){
        console.log("button clicked");
 }
```

Use the button component in root component html:

```
<app-button text="Click-Me" color="red"></app-button>
```

Now Click the button to see output in console log

Now Let’s add CSS Class as input:

```
<app-button text="Click-Me" cssClass="blueBorder" color="red"></app-button>
```

Put blueBorder in main styles.css

```
.blueBorder{
        border: 1px blue dashed;
}
```

In button component’s html, put:

```
<input type="button" [ngClass]="cssClass" [ngStyle]="{'background-color':color}" value="{{text}}"
(click)="onClick()" />
```

Add another input in button component’s class

```
 @Input() cssClass:string;
```

Now We Convert This to EventEmitter

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';

@Component({
  selector: 'app-button',
  templateUrl: './button.component.html',
  styleUrls: ['./button.component.css']
})
export class ButtonComponent implements OnInit {
  @Input() text:string;
  @Input() color:string;
  @Input() cssClass:string;
  @Output() btnClick = new EventEmitter();

  constructor() { }

  ngOnInit(): void {
  }

  onClick(){
        this.btnClick.emit();
  }

}
```

In App component we can now write:

```
<header>
        <h1>{{title}}</h1>
        <app-button cssClass="blueBorder" color="blue" text="Add"
(btnClick)="printMessage()"></app-button>
```

```
</header>
```

Add the method in App component’s class

```
printMessager() {
          console.log("clicked");
 }
```

Now Click the button to see output in console log

Now Let’s create a new component “Course”

```
ng generate component components/course
```

Add that component below our button

```
<header>
          <h1>{{title}}</h1>
          <app-button cssClass=”blueBorder” text="Add" (btnClick)="printMessager()"></app-button>
          <app-course></app-course>
</header>
```

Now Let us create a service

```
ng generate service services/course
```

Create ICourse interface and Course class in data folder:

```
export interface ICourse
{
      name : string;
      fees : number;
}
```

```
import { ICourse } from "./ICourse";

export class Course implements ICourse{
      public name : string;
      public fees : number;

      // constructor syntax to easily initialize current object
      public constructor(init?:Partial<Course>) {
         Object.assign(this, init);
      }
}
```

Let’s modify the service class to contain the data fetching logic

```
import { Injectable } from '@angular/core';
import { ICourse } from '../data/ICourse'
import { Course } from '../data/Course';

@Injectable({
  providedIn: 'root'
})
export class CourseService {

  constructor() { }

  getCourses() : ICourse[]{
      return [
      new Course({ name : "C#", fees : 8000 }),
      new Course({ name : "Asp.net", fees : 30000 })
      ];
  }
}
```

And modify the app component to use the service

```
import { Component } from '@angular/core';
import { ICourse } from './data/ICourse';
import { CourseService } from './services/course.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title : string = 'first-demo-app';
  courses: ICourse[];

  constructor(private courseService: CourseService){ }

  update(){
      this.courses = this.courseService.getCourses();
  }
}
```

Update App component html to add the course data in course component:

<h1>{{title}}</h1>
<app-button text="Click-Me" cssClass="blueBorder" color="red" (btnClick)="update()"></app-button>
<app-course [courses]="courses"></app-course>

Add following code in course component's html file:

<div *ngFor="let c of courses">
    {{c.name}} {{c.fees}}
</div>

Now click the button to see the output

Now let's add rxjs to make the method observable

Import rxjs in service

```
import { Observable, of } from 'rxjs';
```

Modify getCourses method

```
getCourses() : Observable<ICourse[]>{
      return of([
      new Course({ name : "C#", fees : 8000 }),
      new Course({ name : "Asp.net", fees : 30000 })
      ]);
  }
```

Change update method

```
update(){
      this.courseService.getCourses().subscribe((data) =>
      (this.courses = data)
      );
  }
```

Now click the button to see the course list through observable.

Connect With backend Server

Finally our service should look like:

```
import { Injectable } from '@angular/core';
import { ICourse } from '../data/ICourse';
import { Course } from '../data/Course';
import { Observable, of } from 'rxjs';
import { HttpClient } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})
export class CourseService {

  private tokenUrl : string = "/api/v3/token";
  private dataUrl: string = "/api/v3/course/getdata"
  private username: string = "api.user@gmail.com";
  private password: string = "Hello$DevSkill22";

  constructor(private http: HttpClient) { }

  getToken() : Observable<string>{
      let token = this.http.get(this.tokenUrl +
      `?email=${this.username}&password=${this.password}`,
      {
            headers: {'Accept':'text/html, application/xhtml+xml,
*/*','Content-Type':'application/x-www-form-urlencoded'},
            responseType: 'text'
      },
      );

      return token;
  }

  getCourses(token: string) : Observable<ICourse[]>{

      return this.http.get<ICourse[]>(this.dataUrl,
      {
      headers: {'Authorization':`Bearer ${token}`}
      },
      );
  }
}
```

Update the app component’s update method:

```
update() {
      this.courseService.getToken().subscribe((token) =>
      (
      this.courseService.getCourses(token).subscribe((data) =>
            (this.courses = data)
      )
      )
      );
  }
```

Now Add the HttpClient Module in app.module file

```
import { HttpClientModule } from '@angular/common/http';
```

Also,

```
imports: [
      BrowserModule,
      AppRoutingModule,
      HttpClientModule
  ],
```

Now create a proxy file name : “proxy.conf.json” at the root of the site:

```
{
  "/api/v3/*": {
  "target": "https://localhost:7003",
  "secure": false,
  "logLevel": "debug",
  "changeOrigin": true
}
}
```

Then add the following line in angular.json

```
"serve": {
        "builder": "@angular-devkit/build-angular:dev-server",
        "options": {
        "browserTarget": "firstdemo-front:build",
        "proxyConfig": "proxy.conf.json"
        },
```

Make sure to add the url : http://localhost:4200
In the CORS rule of the API