Services

Step 1: Create **Project**
ng new product-store --no-standalone
==========
Step 2: Create **Services**
ng generate service product
OR
ng g service product
==========
Step 3: Create **Components**
ng generate component product-list
OR
ng g component product-list
==========
Service
Step 4: Go to src/app/**product.service.ts**

```
// To Store the Products
import { Injectable } from '@angular/core';

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

// Create an array
  private products = [
    { id: 1, name: 'Laptop', price: 1000 },
    { id: 2, name: 'Smartphone', price: 700 },
    { id: 3, name: 'Tablet', price: 300 },
  ];

  constructor() {}

  getProducts() {
    return this.products; // Create a method to return the array
  }
}
```

TS product.service.ts ×

product-store > src > app > TS product.service.ts > ...

```ts
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  private products = [
    { id: 1, name: 'Laptop', price: 1000 },
    { id: 2, name: 'Smartphone', price: 700 },
    { id: 3, name: 'Tablet', price: 300 },
  ];

  constructor() {}

  getProducts() {
    return this.products;
  }
}

```

==========

Component

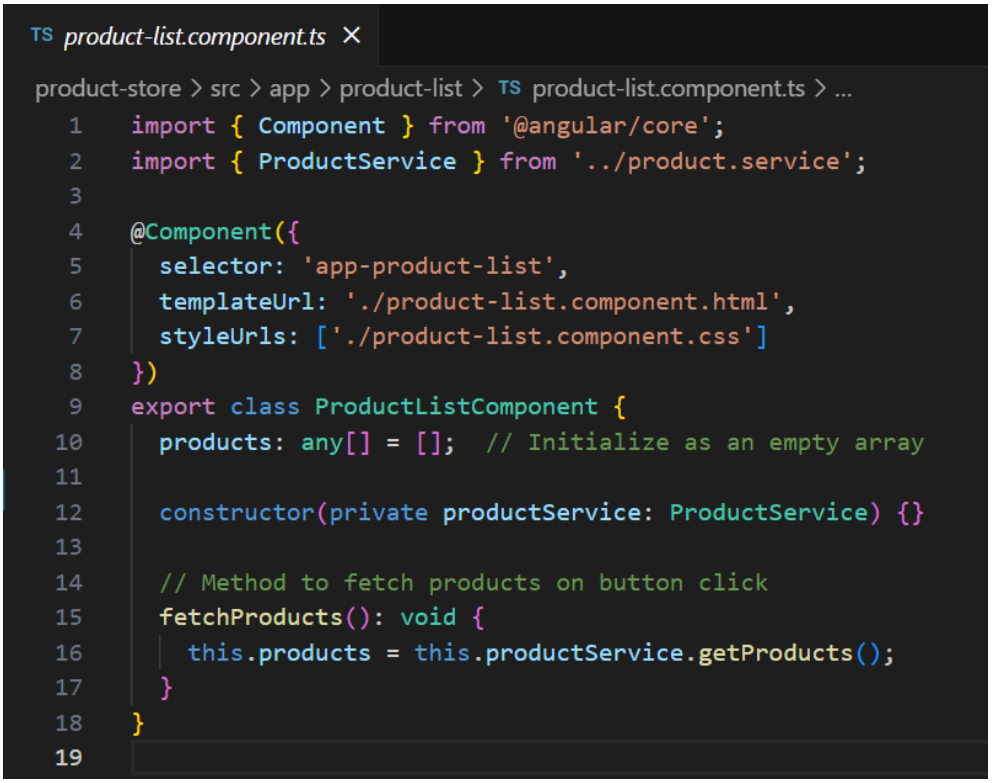
Step 5: Go to src/app/product-list/**product-list.component.ts**

```
// Fetch the products from the above file and store it in an empty array
import { Component } from '@angular/core';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent {
  products: any[] = [];  // Initialize as an empty array

  constructor(private productService: ProductService) {}

  // Method to fetch products on button click
  fetchProducts(): void {
    this.products = this.productService.getProducts();
  }
}
```



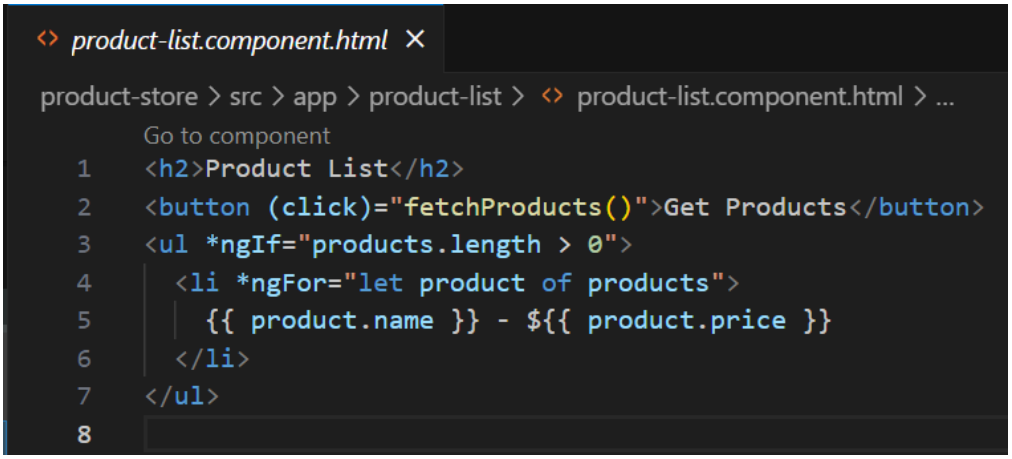
==========

Component

Step 6: Go to src/app/product-list/**product-list.component.html**

// Display title, button, and the response

```
<h2>Product List</h2>
<button (click)="fetchProducts()">Get Products</button>
<ul *ngIf="products.length > 0">
  <li *ngFor="let product of products">
    {{ product.name }} - ${{ product.price }}
  </li>
</ul>
```



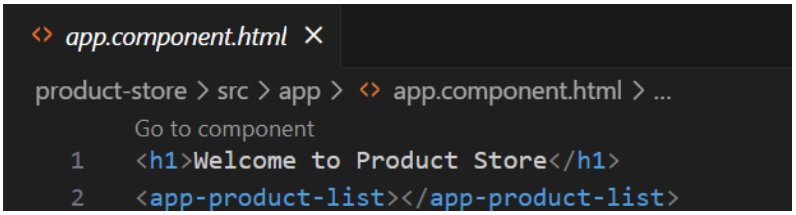
==========

Main Component

Step 7: Go to src/app/**app.component.html**

// Include it to display

<h1>Welcome to Product Store</h1>
<app-product-list></app-product-list>



==========

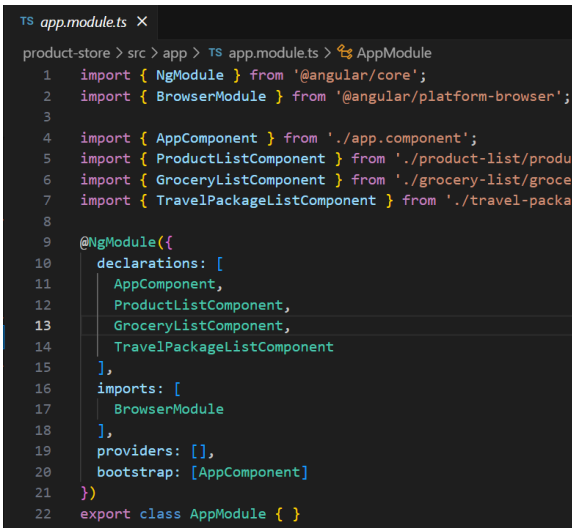
App Module

Step 8: Go to src/app/**app.module.ts**

```
// Settings
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { ProductListComponent } from './product-list/product-list.component';

@NgModule({
  declarations: [
    AppComponent,
    ProductListComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```



==========

Last Step:

ng serve

==========

FileSystem
Part a: Writing Content to a File Using Node.js

// Step 1: Create the File indus.js with the following code:

```
// indus.js
var fs = require('fs');

fs.writeFile('file1.txt', 'Indus University, formerly Indus Institute of Technology and Engineering is a private university established in 2006. In 2012, Indus received university status and was recognized by the University Grants Commission. Indus University is an All India Council for Technical Education approved university.', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

//Step 2: Run the Code:

```
node indus.js
```

==========

Part b: Reading the Content of the File Using Node.js

// Step 1: Create a file readFile.js

```
// readFile.js
const fs = require("fs");
const buf = new Buffer.alloc(1024);

console.log("Opening an existing file");

fs.open("H:/file1.txt", "r+", function (err, fd) {
  if (err) {
    return console.error(err);
  }
  console.log("File opened successfully!");
  console.log("Reading the file");

  fs.read(fd, buf, 0, buf.length, 0, function (err, bytes) {
    if (err) {
      console.log(err);
    }
    console.log(bytes + " bytes read");
    if (bytes > 0) {
      console.log(buf.slice(0, bytes).toString());
    }
  });
});
```

//Step 2: Run the Code:

node readFile.js

==========

writeFile.js

```
// indus.js
var fs = require('fs');

fs.writeFile('file1.txt', 'Indus University, formerly Indus Institute of Technology and Engineering is a private university established in 2006. In 2012, Indus received university status and was recognized by the University Grants Commission. Indus University is an All India Council for Technical Education approved university.', function (err) {
    if (err) throw err;
    console.log('Saved!');
});
```

readFile.js

```
var fs = require("fs");                 // Import the 'fs' (file system) module in Node.js
var buf = Buffer.alloc(1024);           // Create a buffer with a size of 1024 bytes

fs.open("file1.txt", "r+", function (err, fd) {     // Open the file 'file1.txt' in read and write mode
    if (err) throw err;
    console.log("Reading the file");

    fs.read(fd, buf, 0, buf.length, 0, function (err, bytes) {    // Read data from the file into the buffer
        if (err) throw err;
        console.log(bytes + " bytes read");                       // Print the number of bytes read
        if (bytes > 0) {                                          // Check if any bytes were read
            console.log(buf.slice(0, bytes).toString());          // Convert read bytes to a string and print
        }
    });
});

```

ng-Module (Contact-app in zip)

Step 1:

ng new contact-app –no-standalone

==========

Step 2:

ng g component contact

==========

Step 3: Contact.component.css

.contact-wrapper {
background-color: #62d89f;
padding: 40px 20px;
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
min-height: 100vh;
}
.contact-header {
text-align: center;

```
color: white;
margin-bottom: 30px;
}
.contact-header h2 {
font-size: 32px;
font-weight: bold;
margin: 0;
}
.contact-header p {
font-size: 16px;
margin: 10px 0 0;
color: rgba(255, 255, 255, 0.8);
}
.contact-form-card {
background-color: white;
padding: 30px;
border-radius: 10px;
box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.1);
max-width: 500px;
width: 100%;
}
.form-row {
display: flex;
justify-content: space-between;
margin-bottom: 15px;
}
.form-group {
flex: 1;
margin-right: 15px;
}
.form-row .form-group:last-child {
margin-right: 0;
}
label {
font-size: 14px;
font-weight: bold;
display: block;
margin-bottom: 5px;
}
input, textarea {
width: 100%;
padding: 12px;
border: 1px solid #ddd;
border-radius: 5px;
font-size: 16px;
box-sizing: border-box;
}
textarea {
```

```
height: 100px;
resize: none;
}
button {
width: 30%;
padding: 12px;
background-color: #62d89f;
color: white;
border: none;
border-radius: 30px;
font-size: 16px;
cursor: pointer;
transition: background-color 0.3s ease;
margin-left: 32%;
}
button:hover {
background-color: #249e5b;
}
button:disabled {
background-color: #ccc;
cursor: not-allowed;
}
```

==========

Step 4: Contact.component.html

```
<div class="contact-wrapper">
<div class="contact-header">
<h2>Get in touch</h2>
</div>
<div class="contact-form-card">
<form (ngSubmit)="onSubmit()" #contactForm="ngForm">
<div class="form-row">
<div class="form-group">
<label for="firstName">First Name</label>
<input type="text" id="firstName" placeholder="First Name" name="firstName" ngModel
required />
</div>
<div class="form-group">
<label for="lastName">Last Name</label>
<input type="text" id="lastName" placeholder="Last Name" name="lastName" ngModel
required />
</div>
</div>
<div class="form-row">
<div class="form-group">
<label for="email">Email</label>
<input type="email" id="email" placeholder="Email" name="email" ngModel required />
</div>
<div class="form-group">
```

```
<label for="phone">Phone No.</label>
<input type="tel" id="phone" placeholder="Phone No." name="phone" ngModel required
/>
</div>
</div>
<div class="form-group">
<label for="message">Message</label>
<textarea id="message" placeholder="Write your message..." name="message" ngModel
required></textarea>
</div>
<button type="submit" [disabled]="!contactForm.form.valid">Send Message</button>
</form>
</div>
</div>
```

==========

Step 6: contact.component.ts

```
import { Component } from '@angular/core';
@Component({
selector: 'app-contact',
templateUrl: './contact.component.html',
styleUrls: ['./contact.component.css']
})
export class ContactComponent {
onSubmit() {
alert("Message sent successfully!");
}
}
```

==========

Step 7: App.component.html

```
<app-contact></app-contact>
```

==========

Step 8: app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { ContactComponent } from './contact/contact.component';
import { FormsModule } from '@angular/forms';
@NgModule({
declarations: [
AppComponent,
ContactComponent
],
imports: [
BrowserModule,
FormsModule
],
providers: [],
```

```
bootstrap: [AppComponent]
})
export class AppModule { }
```