



If you do not pay any relevant course fee to maintain Earth University BIT platform,
Be ethical enough to contribute while you are using our Academic Contents

www.earth.lk/community
2004-2024 (20th Anniversary)

Sprint-2 Plan

Employee Detail Management

1. Establishing Project Folder Architecture
2. Server App Initialization
3. **Sprint-2 Execution** [2(a), **2(b)**, 2(c), 2(d), 2(e), 2(f)]
4. Completing Sprint-1 Objectives

2(a) Employee Module - Analysis, Design & DB-Preparation

Data Design, UI Design, Sample Data, Database Preparation

2(b) Employee View

1. [Controller Design](#)
2. [Coding-Server App](#)
 - [Version-1 \(Client Pagination\)](#)
[Entity Coding, DAO Coding, Controller Coding, Testing](#)
 - Version-2 (Server Pagination)
Controller Modification, Testing
3. [Coding-Client App](#)
 - [Version-1 \(Client Pagination\)](#)
[UI Preparation, Entity Coding, Service Coding, Controller Coding, Testing](#)
 - Version-2 (Server Pagination)
Service Modification, Controller Modification, Testing

2(c) Employee Search

2(d) Employee Insert

2(e) Employee Update

2(f) Employee Delete

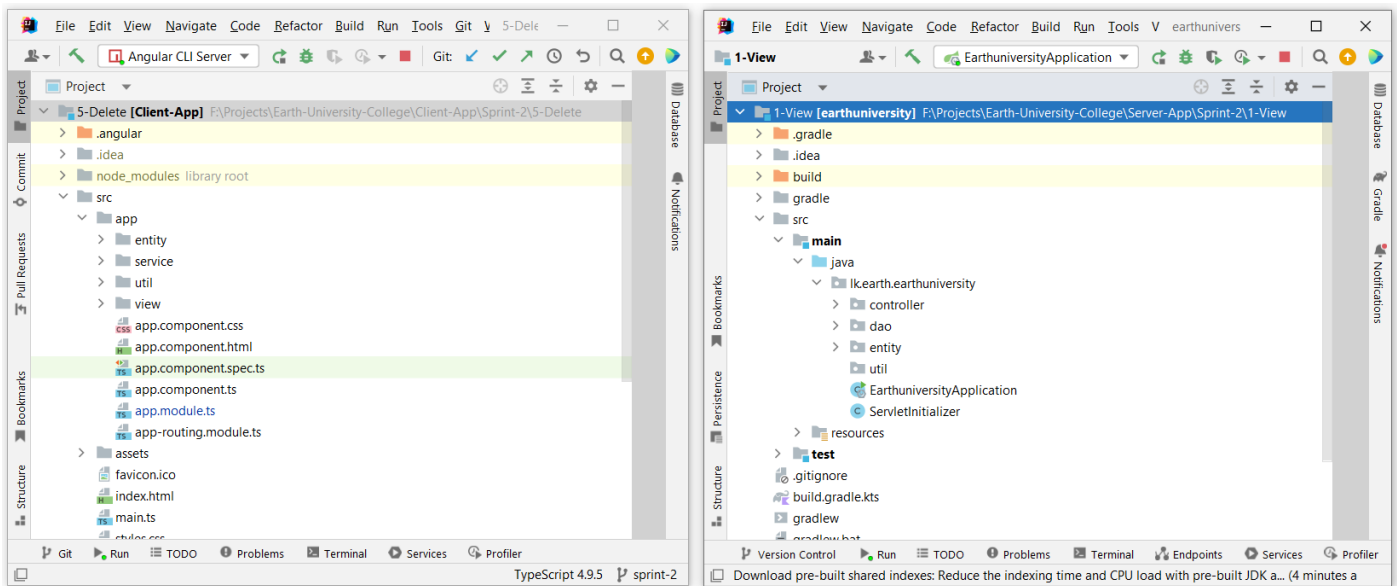
2(b) Employee View

1. [Controller Design](#)
2. [Coding Server-App](#)
3. [Coding Client-App](#)
 - Version-1 (Client Pagination)**
 1. Client App Preparation for Sprint-2
 2. UI Preparation with Static Data & Testing
 3. Coding Entity Classes
 4. Data Loading with Dynamic Data & Testing
 5. Coding Service Classes for Dynamic Data & Testing (With & Without DI)
 6. Error Correction – Server App – Employee Data Service
 7. **Data loading from Server & Testing**, Promise-Pending/Resolve/Reject Status
 8. Client Pagination
 9. Client Search
 - Version-2 (Server Pagination)**

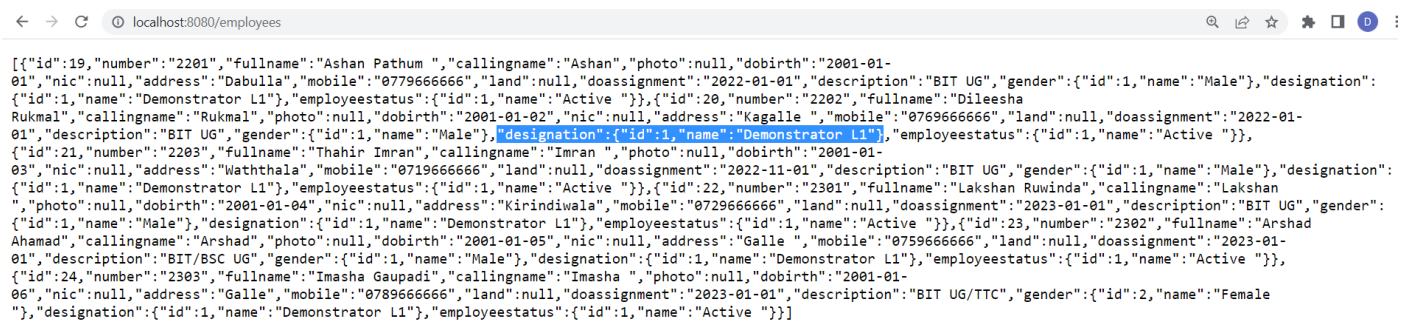
Data Loading from the Server and Testing

1. Open and Run both the Server-App & Client-App in separate IntelliJ-IDEA windows.

Server App – 8080, Client App – 4200



2. Test the Employee Service “localhost:8080/employees”



Important Notice on Entity Classes

During this step, we do not use internal coding of the Entity Classes.

We use entity classes only for semantic readability and extendibility in the future needs.

We use strong naming conventions and relationship mapping at the server side.

Hence the output JSON of the services are well formatted.

Client side JavaScript functionalities of JSON.parse() and JSON.stringify() with Angular HttpClient functions will generate well formatted set of JS Objects from JSON string.

3. Import “HTTP-Client” Module in “app.modules.ts”.
(This module contains the injectable Class “HttpClient”).
This class will handle the HTTP request and return HTTP responses converting JSON Strings into JS Object Arrays)

```

19 import {MatSidenavModule} from "@angular/material/sidenav";
20 import {MatListModule} from "@angular/material/list";
21 import {MatExpansionModule} from "@angular/material/expansion";
22 import {MatIconModule} from "@angular/material/icon";
23 import {MessageComponent} from "../util/dialog/message/message.component";
24 import {MatDialogModule} from "@angular/material/dialog";
25 import {CourseComponent} from "../view/modules/course/course.component";
26 import {MatTableModule} from "@angular/material/table";
27 import {HttpClientModule} from '@angular/common/http';
28 import {EmployeeService} from "../service/employeeservice";
29
30
31 @NgModule({
32   declarations: [

```

```

39 CourseComponent,
40 MessageComponent
41 ],
42 imports: [
43   BrowserModule,
44   AppRoutingModule,
45   BrowserModule,
46   MatGridListModule,
47   MatCardModule,
48   ReactiveFormsModule,
49   MatFormFieldModule,
50   MatButtonModule,
51   MatInputModule,
52   MatToolbarModule,
53   MatSidenavModule,
54   MatListModule,
55   MatExpansionModule,
56   MatIconModule,
57   MatDialogModule,
58   MatTableModule,
59   HttpClientModule
60 ],
61 providers: [EmployeeService],
62 bootstrap: [AppComponent]
63 })
64 export class AppModule {
65 }

```

4. Modify the Employee Service Class as follows

Be vigilant on usages of “async”, Promis<Array<Employee>>, “await” (*Concepts will be explained*)
Dependency Injection (HTTP-Client) in the Constructor

```

1 import {Employee} from "../entity/employee";
2 import {Injectable} from "@angular/core";
3 import {HttpClient} from "@angular/common/http";
4
5 @Injectable({
6   providedIn: 'root'
7 })
8
9 export class EmployeeService {
10
11   constructor(private http: HttpClient) { }
12
13   async getAll(): Promise<Array<Employee>> {
14
15     const employees = await this.http.get<Array<Employee>>({ url: 'http://localhost:8080/employees' }).toPromise();
16     if(employees == undefined){
17       return [];
18     }
19     return employees;
20   }
21 }

```

5. Modify the EmployeeUI Controller as follows

```

1 import {Component} from '@angular/core';
2 import {Employee} from "../../entity/employee";
3 import {EmployeeService} from "../../service/employeeservice";
4
5 @Component({
6   selector: 'app-employee',
7   templateUrl: './employee.component.html',
8   styleUrls: ['./employee.component.css']
9 })
10
11 export class EmployeeComponent {
12
13   columns: string[] = ['number', 'callingname', 'gender', 'designation', 'name'];
14   employees: Array<Employee> = [];
15
16   constructor(private es:EmployeeService) {
17
18     this.es.getAll().then( (emps: Employee[]) => { this.employees=emps; } );
19   }
20 }

```

6. Observer the Employee UI

Number	Calling Name	Gender	Designation	Full Name
2201	Ashan	Male	Demonstrator L1	Ashan Pathum
2202	Rukmal	Male	Demonstrator L1	Dileesha Rukmal
2203	Imran	Male	Demonstrator L1	Thahir Imran
2301	Lakshan	Male	Demonstrator L1	Lakshan Ruwinda
2302	Arshad	Male	Demonstrator L1	Arshad Ahamad
2303	Imasha	Female	Demonstrator L1	Imasha Gaupadi

```

[{"id":19,"number":"2201","fullname":"Ashan Pathum ","callingname":"Ashan","photo":null,"dobirth":"2001-01-01","nic":null,"address":"Dabulla","mobile":"0779666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}},{"id":20,"number":"2202","fullname":"Dileesha Rukmal","callingname":"Rukmal","photo":null,"dobirth":"2001-01-02","nic":null,"address":"Kagalle ","mobile":"0769666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}},{"id":21,"number":"2203","fullname":"Thahir Imran","callingname":"Imran ","photo":null,"dobirth":"2001-01-03","nic":null,"address":"Waththala","mobile":"0719666666","land":null,"doassignment":"2022-11-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}},{"id":22,"number":"2301","fullname":"Lakshan Ruwinda","callingname":"Lakshan ","photo":null,"dobirth":"2001-01-04","nic":null,"address":"Kirindiwala","mobile":"0729666666","land":null,"doassignment":"2023-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}},{"id":23,"number":"2302","fullname":"Arshad Ahamad","callingname":"Arshad","photo":null,"dobirth":"2001-01-05","nic":null,"address":"Galle ","mobile":"0759666666","land":null,"doassignment":"2023-01-01","description":"BIT/BSC UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}},{"id":24,"number":"2303","fullname":"Imasha Gaupadi","callingname":"Imasha ","photo":null,"dobirth":"2001-01-06","nic":null,"address":"Galle","mobile":"0789666666","land":null,"doassignment":"2023-01-01","description":"BIT UG/TTC","gender":{"id":2,"name":"Female"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]

```