



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](#)

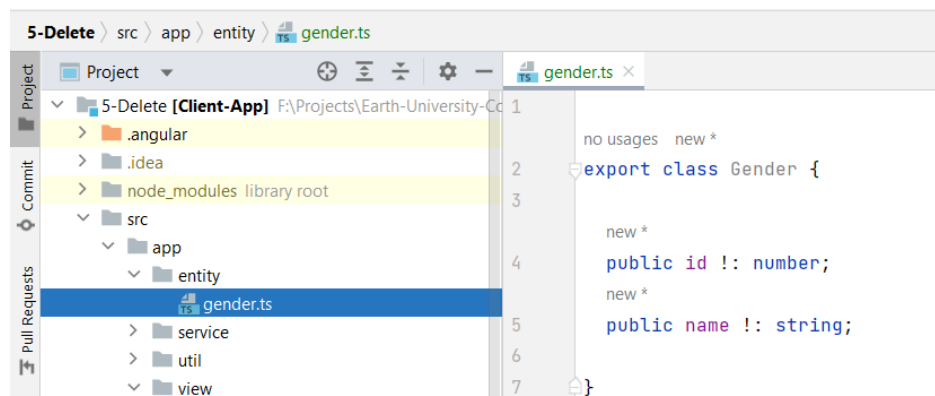
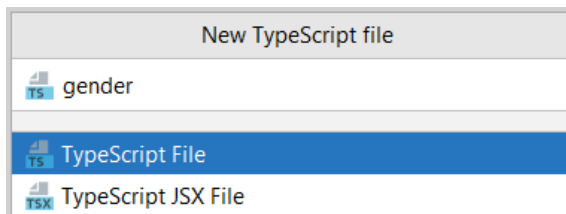
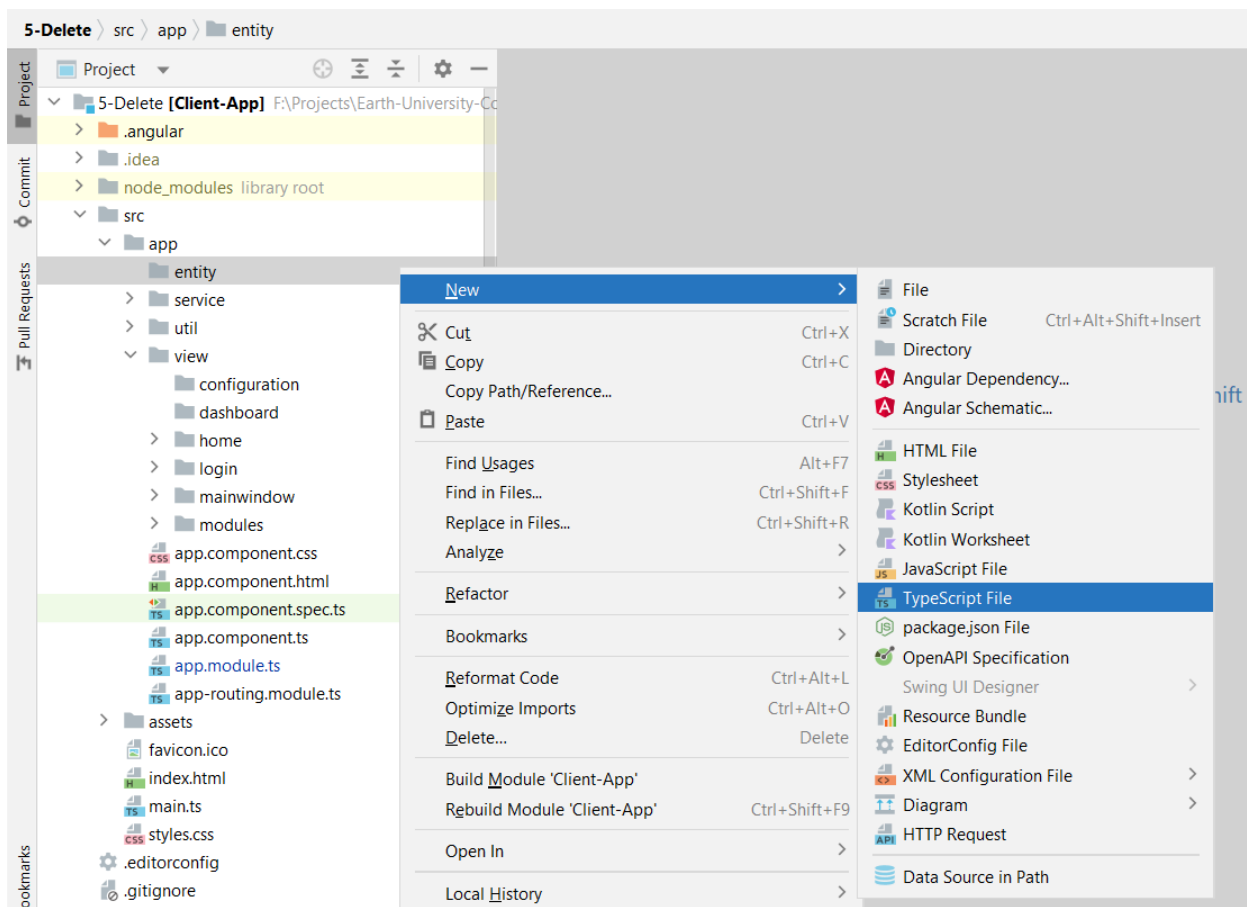
Version-1 (Client Pagination)

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
6. Data loading from Server & Testing
7. Entity Auto generating & Testing

Version-2 (Server Pagination)

1. Create Entity Class to Encapsulate Data

Simple Manual Coded Entity Classes (Later in the Sprint-2, we will remove them and replace with auto generated Entities)



Repeat with Designation, Employeestatus entities.

Code entity Employee as follows.

Try to explain codes.

```

1 import {Gender} from "../gender";
2 import {Designation} from "../designation";
3
4 export class Employee{
5     new *
6     public id!: number;
7     new *
8     public name!: string;
9     new *
10    public number!: string;
11    new *
12    public callingname!: string;
13    new *
14    public gender!: Gender;
15    new *
16    public designation!: Designation;
17
18    2 usages new *
19    constructor(id:number, name:string, number:string, callingname:string, gender:Gender, designation:Designation ) {
20        this.id=id;
21        this.name=name;
22        this.number=number;
23        this.callingname=callingname;
24        this.gender=gender;
25        this.designation=designation;
26    }
27 }

```

2. Create Employee Objects in the constructor and add them into the data array bind with the table (employee-components.ts)

Change the value of Staff Number 2201 to **2201-N** and 2202 to **2202-N** for distinguishing test results

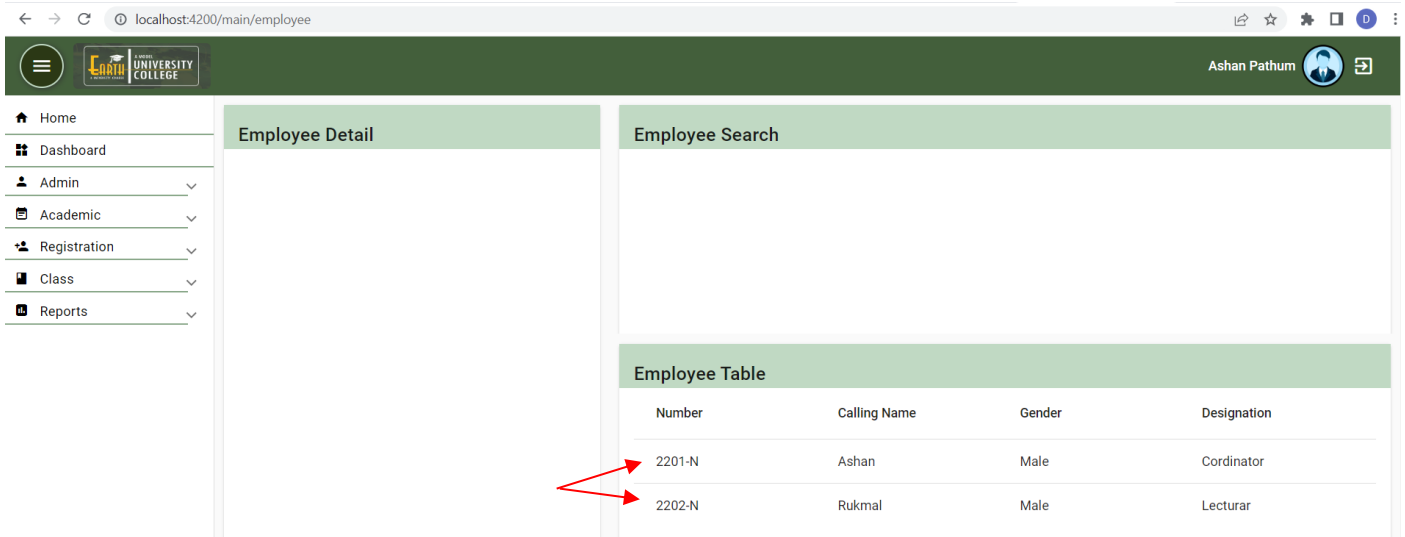
```

1 import {Component} from '@angular/core';
2 import {Gender} from "../../entity/gender";
3 import {Designation} from "../../entity/designation";
4 import {Employee} from "../../entity/employee";
5
6 @Component({
7     selector: 'app-employee',
8     templateUrl: './employee.component.html',
9     styleUrls: ['./employee.component.css']
10 })
11
12 export class EmployeeComponent {
13
14     2 usages new *
15     columns: string[] = ['number','callingname','gender', 'designation'];
16     3 usages new *
17     employees: Array<Employee> = [];
18
19     no usages new *
20     constructor() {
21
22         let g1:Gender = new Gender( id: 1, name: "Male");
23         let d1:Designation = new Designation( id: 1, name: "Coordinator");
24         let d2:Designation = new Designation( id: 2, name: "Lecturan");
25
26         let e1:Employee = new Employee( id: 1, name: "Ashan Pathum", number: "2201-N", callingname: "Ashan", g1, d1);
27         let e2:Employee = new Employee( id: 1, name: "Rukmal Dileesha", number: "2202-N", callingname: "Rukmal", g1, d2);
28
29         this.employees.push(e1);
30         this.employees.push(e2);
31     }
32 }

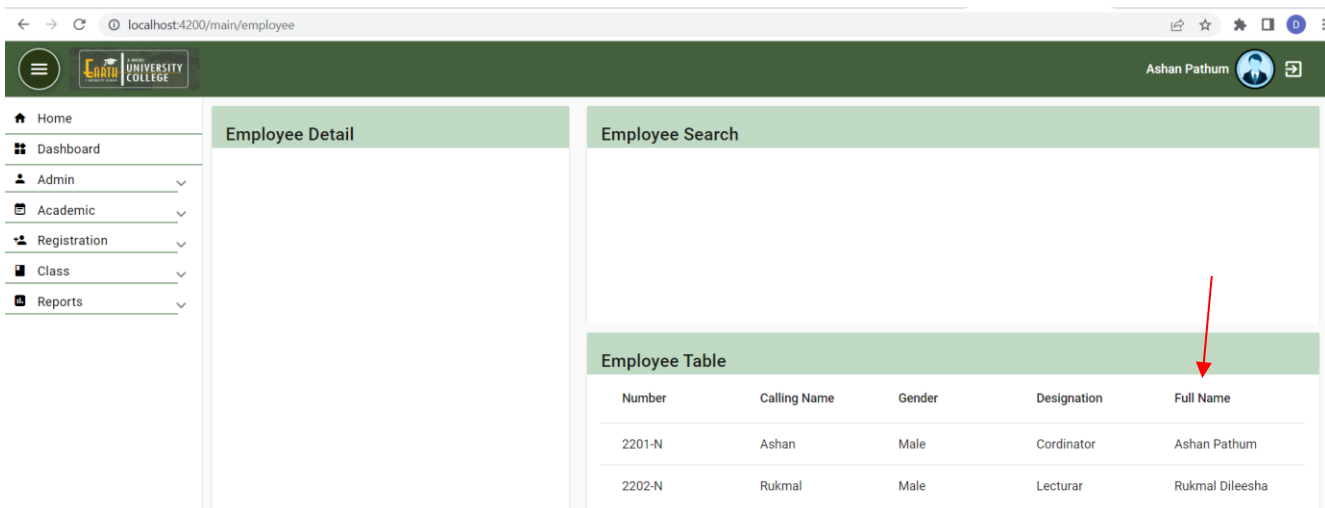
```

Discuss the usages of “this” and “let” in the above codes

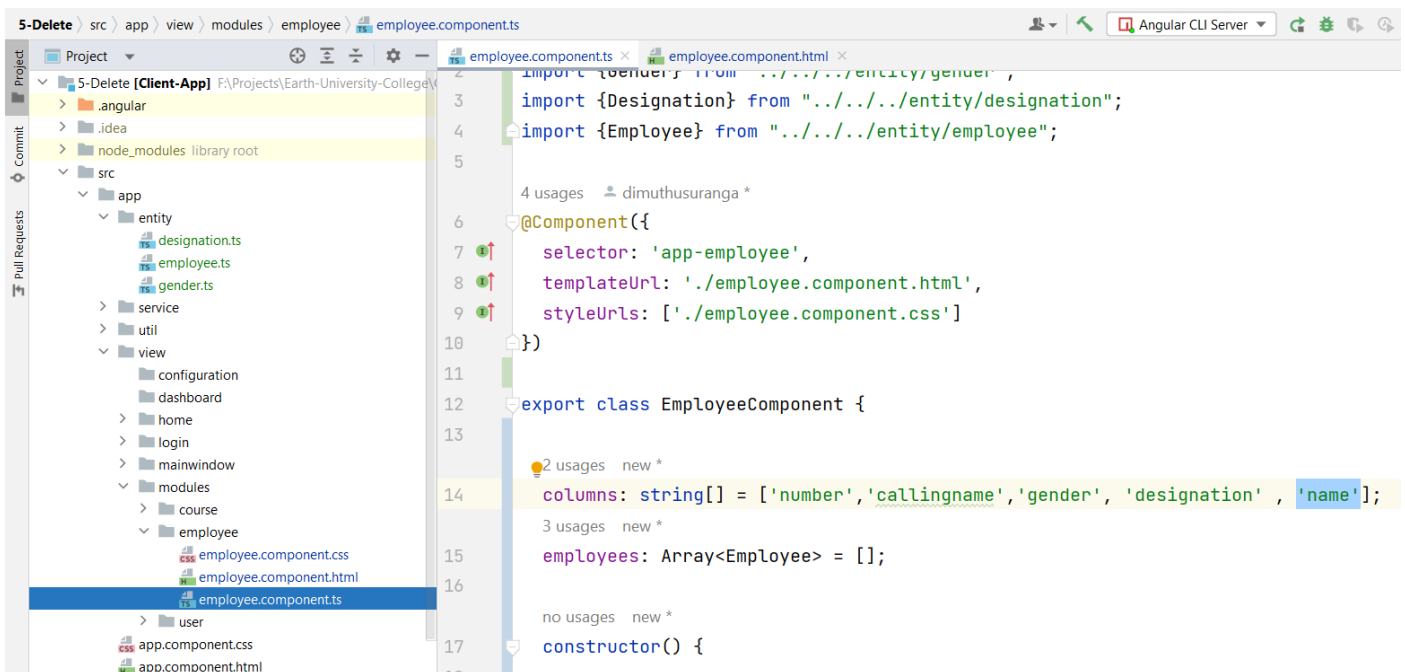
3. Re run the project if the project is not running and observe the output with Staff Number Changes



4. Adding a new column to display the “name” of the employee available in entity objects



- (i) Add a column into the column array in the UI-Controller (.ts file)
- (ii) Add a column into the Employee-UI (.html file) with binding codes (object property to be displayed)



(iii) Test and Observe the Output with the New Table