If you do not pay any relevent 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 (i) Normal (ii) Using Template (a) Direct Property (b) Processed
- 10. Formal Method Architecture
- 11. Code Explanation
- 12. Code Modification

Version-2 (Server Pagination)

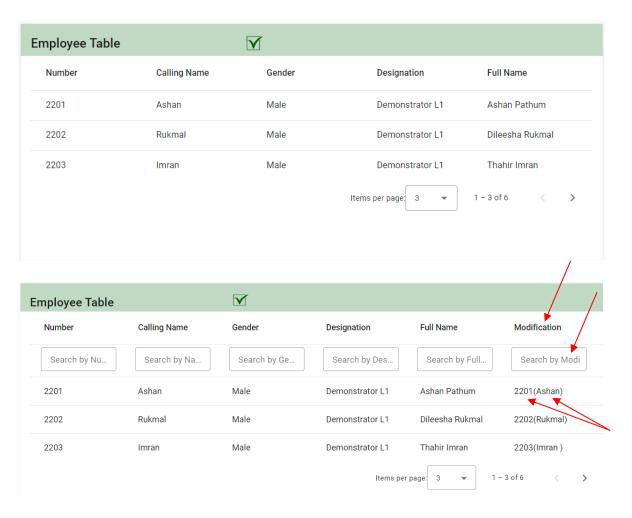
Client Search – (i) Using User Defined Template with Processing Pipe Functions

Code Modification – Add a Column at the End of the table to show "Number(Calling Name)"

Column Name - modi

Column Header - Modification

Binding Values - Processed Output with "number" & "callingname" as 2201(Ashan), 2202(Rukmal)



Here Binding Value is not just a property reference.

Hence we need to use another Pipe function to get the processed value to display

Observe the importance of the Maintainability and the strength of a design in Code Modification.

(a) Test Driven Development will start with Addition of a simple Column to Display Number Column Again

(i) Add data to the 5 Arrays for Both Display and Client Search

```
columns: string[] = ['number', 'callingname', 'gender', 'designation', 'fullname', 'modi'];
headers: string[] = ['Number', 'Calling Name', 'Gender', 'Designation', 'Full Name', 'Modification'];
binders: string[] = ['number', 'callingname', 'gender.name', 'designation.name', 'fullname', 'number'];

cscolumns: string[] = ['csnumber', 'cscallingname', 'csgender', 'csdesignation', 'csname', 'csmodi'];
csprompts: string[] = ['Search by Number', 'Search by Name', 'Search by Gender', 'Search by Designation', 'Search by Full Name', 'Search by Modi'];
public csearch!: FormGroup;

employees: Array<Employee> = [];
data!: MatTableDataSource<Employee>;
imageurl: string = '';
@ViewChild(MatPaginator) paginator!: MatPaginator;
```

(ii) Add Form Control(csmodi) into From Group to support Client Search (csearch)

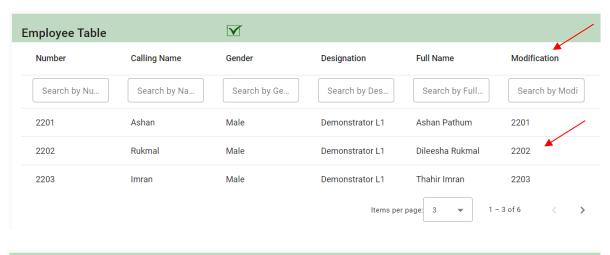
```
30
          constructor(private es: EmployeeService, private fb: FormBuilder) {
            this.imageurl = 'assets/pending.gif';
            this.es.getAll() Promise<Employee[]>
              .then((emps: Employee[]) => {
                this.employees = emps;
                this.imageurl = 'assets/fullfilled.png';
 38
 39
              }) Promise < void>
              .catch((error) => {
 41
                console.log(<u>error</u>);
                this.imageurl = 'assets/rejected.png';
              Promise < void>
               .finally( onfinally: () => {
                this.data = new MatTableDataSource(this.employees);
                this.data.paginator = this.paginator;
47
              });
49
            this.csearch = this.fb.group( controls: {
              "csnumber": new FormControl(),
               "cscallingname": new FormControl(),
              "csgender": new FormControl(),
              "csdesignation": new FormControl(),
               "csname": new FormControl(),
               "csmodi": new FormControl(), ◀
            });
58
          }
59
```

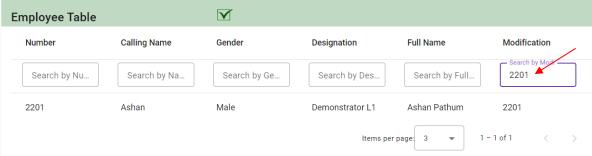
(iii) Add a Filter Criteria to support Client Search (Copy from Number Column and Change the "csmodi"

```
filterTable(): void {
           const cserchdata = this.csearch.getRawValue();
           this.data.filterPredicate = (employee: Employee, filter: string) => {
65
            return (cserchdata.csnumber == null || <u>employee</u>.number.toLowerCaxe().includes(cserchdata.csnumber)) &&
              && ((cserchdata.cscallingname == null || <u>employee</u>.callingname)). (cserchdata.cscallingname))
67
               (cserchdata.csgender == null || employee.gender.name.tacowerCase().includes(cserchdata.csgender)) &&
68
               (cserchdata.csdesignation == null || employee.designation.name.toLowerCase().includes(cserchdata.csdesignation)) &&
               (cserchdata.csname == null || employee.fullname.toLowerCase().includes(cserchdata.csname)) &&
               (cserchdata.csmodi == null || employee.number.toLowerCase().includes(cserchdata.csmodi));
           };
           this.data.filter = 'xx';
74
75
```

(iv) Keep the "getProperty()" same as the previous

(v) Test the Output for all outcome of the affected component (Regression Testing)
View All, Client Search





(b) Test Driven Development – Step 2

Convert the above column to Support Processed Output as 2201(Ashan) instead of 2201

(i) Change the binding value to a Function name as "getModi()"

You can imagine this with "getProfit()" in Item Table, where Profit will be calculated using Purchase-Price and Sale-Price

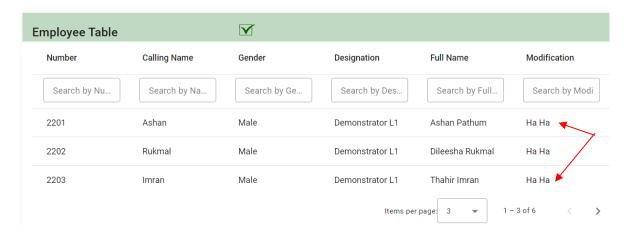
(ii) Change the "getProperty()" to support references of pipe functions in addition to property reference

```
getProperty(element: {}, reference: string): string {

if (!reference.endsWith("()")) {

const value = reference.split(separator: '.').reduce((outering: index to index t
```

(iii) Test the Output



(iv) Add a Custom Cell-Value-Factory (As in the JavaFX), here a Pipe Function in Angular to build the content to display

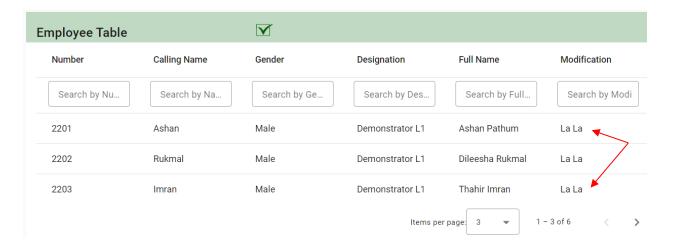
```
92
93 getModi(element: Employee) {
94 return 'La La';
95 }
```

(v) Modify the getProperty() to call this function to build the cell values

```
78
          getProperty(element: {}, reference: string): string {
79
80
            if (!reference.endsWith("()")) {
81
              const value = reference.split( separator: '.').reduce((o : {} , a : string ) => {
82
                //@ts-ignore;
83
                return o[a];
84
              }, element) as string;
85
              return value;
86
87
88
            else {
              //@ts-ignore;
89
              return this.getModi(element);
90
91
            }
92
93
       }
94
95
         getModi(element: Employee) {
96
            return 'La La';
97
98
          }
```

Discuss true/false – Any Column specified in columns array with "()" will get this Value Discuss – Why we use //@ts-ignore; here.

(vi) Observe the Output



(vii) Change the getProperty() function to specific Pipe Function for each Column as specified in the Column Definition Array

```
getProperty(element: {}, reference: string): string {
79
            if (!reference.endsWith("()")) {
81
              const value = reference.split( separator: '.').reduce((o :{} , a :string ) => {
82
                //@ts-ignore;
83
                return o[a];
84
              }, element) as string;
              return value;
86
            }
            else {
88
              reference = reference.substring(0, reference.indexOf('('));
89
90
              //@ts-ignore;
              return this[reference](element);
92
            }
93
94
```

Discuss the importance of the following line

```
(1) reference = reference.substring(0, reference.indexOf('('));
(2) return this[reference](element);
```

Discuss alternative approaches in converting String reference of a Function into real executable function call in Type Script

Discuss the Importance of Java Reflection API in referencing Methods and Attributes in Java

(viii) Change the "getModi()" as follows,

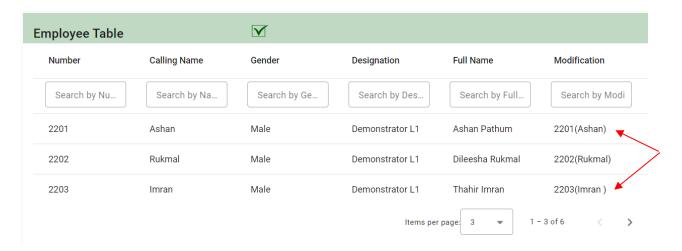
(ix) Observe the Output

Number	Calling Name	Gender	Designation	Full Name	Modification
Search by Nu	Search by Na	Search by Ge	Search by Des	Search by Full	Search by Mo
2201	Ashan	Male	Demonstrator L1	Ashan Pathum	Na Na
2202	Rukmal	Male	Demonstrator L1	Dileesha Rukmal	Na Na
2203	Imran	Male	Demonstrator L1	Thahir Imran	Na Na

(x) Change the "getModi()" as follows

```
96
97  getModi(element: Employee) {
98
99  return element.number + '(' + element.callingname + ')';
100
101  }
```

(xi) Observe the Output



(xii) Correcting in Client Search

```
filterTable(): void {
61
62
                         const cserchdata = this.csearch.getRawValue();
                         this.data.filterPredicate = (employee: Employee, filter: string) => {
66
                             return (cserchdata.csnumber == null || employee.number.toLowerCase().includes(cserchdata.csnumber)) &&
67
                                  (cserchdata.cscallingname == null || employee.callingname.toLowerCase().includes(cserchdata.cscallingname)) &&
                                   (cserchdata.csgender == null || <u>employee</u>.gender.name.toLowerCase().includes(cserchdata.csgender)) &&
                                  (cserchdata.csdesignation == null \mid \mid \underline{employee}.designation.name.toLowerCase().includes(cserchdata.csdesignation)) \& (cserchdata.csdesignation) & (cserchdata
70
                                   (cserchdata.csname == null || employee.fullname.toLowerCase().includes(cserchdata.csname)) &&
                                  (cserchdata.csmodi == null || employee.number.toLowerCase().includes(cserchdata.csmodi));
                         };
                         this.data.filter = 'xx';
75
76
                     filterTable(): void {
                         const cserchdata = this.csearch.getRawValue();
64
65
                         this.data.filterPredicate = (employee: Employee, filter: string) => {
66
                              return (cserchdata.csnumber == null || <a href="mailto:employee">employee</a>.number.toLowerCase().includes(cserchdata.csnumber)) &&
                                   (cserchdata.cscallingname == null || employee.callingname.toLowerCase().includes(cserchdata.cscallingname)) &&
                                   (cserchdata.csgender == null || employee.gender.name.toLowerCase().includes(cserchdata.csgender)) &&
                                   (cserchdata.csdesignation == null || employee.designation.name.toLowerCase().includes(cserchdata.csdesignation)) &&
70
                                   (cserchdata.csname == null || employee.fullname.toLowerCase().includes(cserchdata.csname)) &&
                                   (cserchdata.csmodi == null || this.getModi(employee).toLowerCase().includes(cserchdata.csmodi));
                         };
                         this.data.filter = 'xx';
76
```

(xiii) Test Client Search on newly add Column

