CHIANG MAI UNIVERSITY

Bachelor of Science (Software Engineering)

College of Arts, Media and Technology

2nd Semester / Academic Year 2017

SE 331 Component Based Software Development

Separation and Data binding

Name ..... ID .....

Objective In this session, you will experience how to separate more concerns, and tip and trick for the data binding

Suggestion you should read the instructions step by step. Please try to answer a question by question without skipping some questions which you think it is extremely difficult.

Hint The symbol + and – in front of the source code is to show that you have to remove the source code and add the source code only. There are not the part of the source code

0. Setting

0.1. Open the folder `C:\lab`

0.2. Open the command window and then go to the folder `C:\lab\`

0.3. Type `git clone https://github.com/chartchai/SE331-lab02 lab02`

0.4. Run `npm install` and `start` to check that you get the correct application

0.5. The title is not related to our class, change the title of the web application to anything that you think it should match and then

Show what happen in the web browser

Lecture's Signature _____

1. Add the `css`

1.1. The `css` can be added to the angular component as well.

Open the `students.component.ts` and then updated the template part, and add the styles as given.

```
-         <p> {{student. name | uppercase}} {{student.surname}}</p>
+         <p class="student-name"> {{student. name | uppercase}}
       {{student.surname}}</p>
          <p *ngIf="student.gpa > 2.5">Good Student who get grade
       {{student.gpa | number:'1.2-2'}}</p>
          <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
       {{student.gpa  | number:'1.2-2'}}</p>
       </li>
  </ul>

- <p> The average gpa is {{averageGpa()}} </p>`
+ <p class="grade"> The average gpa is {{averageGpa()}} </p>`,
+ styles:[`
+ .student-name{
+      color: #FF0000;
+      font-size: x-large;
+ }
```

```
+ .grade {
+      font-weight: bold;
+ }
+ `
+ ]
```

Hint you may find a lot of red underline warn that your code is not conform to the TSlint, to remove this, Open the settings menu, search for the “TSLint”, and then remove the enable checkpoint. We will come back to the issue later in this course.

1.2. Run the application and show how it changes?

Lecture’s Signature _____

1.3. In the web browser, right click on the student name, and then select “inspect element”. In the inspector is there anything that it has been generated? Write the generated code here

……

1.4. Then clicked in the style editor tab. Open the inline style sheet, describe the relation between what you have written, what has been generated, and how they link?

……

……

2. Similar to the other components, in each file should contains only one concerns. The html snippet and the css snippet should be extract to other file.

2.1. Create `students.component.html`, and `students.component.css`

2.2. copy the content which is the html template from the `students.component.ts` to In the `students.component.html`

2.3. copy the content which is the css template from the `students.component.ts` to In the `students.component.css`, only one css in the list is copied.

2.4. Update the `students.coponents.ts` by removing the template, and styles attribute in the `@Component`, and replaced with this

```
+ templateUrl: 'app/students.component.html',
+ styleUrls:['app/students.component.css']
```

2.5. Run the application to show how it changed

Lecture’s Signature _____

2.6. Extract the html from the time component, and then add any css style to the time component. Show staff the file structures, and the result.

Lecture’s Signature _____

2.7. Extract the html from the app component. Show staff the file structures, and the result.

Lecture’s Signature _____

2.8. To make the source code more tidy, create a new folder name `students`. And then copy all the `students.*` file to the new folder. Do not for get to relink the files that you have to import in each component. And then show the result.

Lecture's Signature _____

3. Creating a model

Type script is the programming language which can create type. Type is used to defined what we can store in our model.

3.1. Create model by creating the `student.ts` file in the `students` folder.

3.2. Add the given content to the file.

```
+export class Student{
+       id: number;
+       studentId: string;
+       name: string;
+       surname: string;
+       gpa: number;
+
+}
```

Hint after you have type or update the file, you may need to press Alt-Ctr-L to reformat the code so the code look better

3.3. Define type to the students list in the `students.component.ts` file

```
export class StudentsComponent {
-       students = [{
+       students: Student[] = [{
                "id": 1,
```

3.4. Check that the source code is still working, the import may be required. You can use auto import to fix it.

4. The data is also required to extract from the component files. +

4.1. Create `mocks.ts` in the app folder to handle all the mock data. The `mocks.ts` file must contain this information

```
+import {Student} from './students/student'
+export const STUDENTS:Student[] = [{
+               'id': 1,
+               'studentId':'562110507',
+               'name': 'Prayuth',
+               'surname':'Tu',
+               'gpa':4.00
+       },{
+               'id': 2,
+               'studentId':"562110509",
+               'name":'Pu',
+               'surname':'Priya',
+               'gpa':0}
+       ];
```

4.2. Update the `students.component.ts` to import the `const` we have created in the `mocks.ts`

```
+import {STUDENTS} from '../mocks';
```

What is "`../`" for? What is the different comparing to "`./`"

.....

.....

4.3. Remove the hard code declaration using only variable declaration.

```
export class StudentsComponent {
-       students: Student[] = [{
```

```
-                "id": 1,
-                "studentId":"562110507",
-                "name":"Prayuth",
-                "surname":"Tu",
-                "gpa":4.00
-        },{
-                "id": 2,
-                "studentId":"562110509",
-                "name":"Pu",
-                "surname":"Priya",
-                "gpa":0}];
+        students: Student[];
```

4.4. Initialize information for the students by overriding `ngOnInit` method in the `StudentsComponent` class as given

```
+        ngOnInit(){
+                this.students = STUDENTS;
+        }
```

4.5. The application must work as same as it is. Now you create a new mock data set, and inject to the `students.component` and show on the browser.

Lecture's Signature _____

5. Data binding

The data from the TypeScript class can inject to the html code using the proper api. Now we will link data from our component to the html file.

5.1. Copy the file from the given link `http://bit.ly/331lab02image` , and copy to the `images` folder. You have to create the `images` folder in the same level as app folder.

5.2. Add a new attribute to the `Student` model to contain the url of the image

```
        gpa: number;
+       image: string;
```

5.3. Update the mock objects to hold the images location by updating the `mocks.ts`

```
                'name':'Prayuth',
                'surname': 'Tu',
-               'gpa':4.00
+               'gpa':4.00,
+               'image': 'images/tu.jpg'
        },{
                'id': 2,
                'studentId':"562110509",
                'name":'Pu',
                'surname': 'Priya',
-               'gpa':0}
+               'gpa':0,
+               'image': 'images/pu.jpg'}
```

5.4. Add the `img` tag to show the image of each student in the students.component.html

```
                        <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
{{student.gpa  | number:'1.2-2'}}</p>
+                       <div>
+                               <img src={{student.image}}>
+                       </div>
                </li>
```

What happen in the web page in the browser?

.....

.....

5.5. What we have done previously is to put the value to the html attribute. But we can bind the value directly to each attribute using [ ] (Block brackets). Update the `img` tag as given

```
-                                   <img src={{student.image}}>
+                                   <img [src]="student.image">
```

5.6. Other attributes can be bound with the class attribute or method using block brackets, as well. For example, to hind some students whose grade is less than 1.5 we can update our `students.component.html` as followed:

```
-               <li *ngFor="let student of students">
+       <div *ngFor="let student of students">
+               <li [hidden]="student.gpa < 1.5" >
                        <h2>{{student.studentId }}</h2>
                </li>
+       </div>
        </ul>
```

Using your mock data to show that you have hind some students whose gpa is not good.

Lecture's Signature _____

5.7. Update your html to show the student name when the mouse is over the image.

Hint there is a tag attribute name `title` to show the text when the mouse is over the image

Lecture's Signature _____

6. Not only attribute, the css class can be selected using the information of the JavaScript object too.

6.1. Add a new css style in the `students.component.css` using the given information

```
+.featured{
+       background: linear-gradient(to bottom right, red, yellow);
+}
```

6.2. Update the `Student` class by inserting the featured boolean to control whether to show the style or not

```
        gpa: number;
        image: string;
+       featured:boolean;

 }
```

6.3. Add the featured value in the mock data

```
                'gpa':4.00,
-               'image':'images/tu.jpg'
+               'image":'images/tu.jpg',
+               'featured':true
        },{
                'id': 2,
                'studentId': '562110509',
                'name': 'Pu',
                'surname': 'Priya",
                'gpa':0,
-               'image': 'images/pu.jpg'}
+               'image': 'images/pu.jpg',
+               'featured':false
+       }
```

6.4. And then remove the hidden attribute and replaced with the code to select the feature to show in the html file.

```
 <div *ngFor="let student of students">
-               <li [hidden]="student.id > 1.5"->
+               <li [class.featured]="student.featured">
                        <h2>{{student.studentId }}</h2>
                        <p class="student-name"> {{student. N  ...
```

Show your final work

Lecture's Signature _____

7. Working with other components.

To work with other components, we need to import the component.

7.1. What is twitter's Bootstrap ?

.....

.....

7.2. Import the Bootstraps component to the project by adding

```
"bootstrap": "^3.3.7",
```

in the dependencies part in the `package.json`, and then call npm install to download the Bootstrap component.

7.3. Link the Bootstrap components to the application by adding the given snippet in the `index.html`

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="styles.css">
-
+       <!-- Bootstrap -->
+    <link href="node_modules/bootstrap/dist/css/bootstrap.min.css"
          rel="stylesheet">
    <!-- Polyfill(s) for older browsers -->
```

In the head tag

And

```
  <body>
    <my-app>Loading AppComponent content here ...</my-app>
+       <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
+    <script
          src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.m
          in.js"></script>
+    <!-- Include all compiled plugins (below), or include individual files
          as needed -->
+    <script src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
  </body>
```

7.4. Then update the `app.component.ts` to use the template file as given

```
  selector: 'my-app',
+ templateUrl : 'app/app.component.html'
 })
 export class AppComponent {
```

7.5. Crete the `app.component.html` using the given html snippet with Bootstrap css

```
+<div class="container-fluid">
+<div class="row page-header">
```

```
+<h1 class="col-md-offset-2">Hello {{name}}</h1>
+</div>
+<div class="row">
+       <students></students><br/>
+</div>
+<div class="row">
+</div>
+</div>
```

7.6. Update the `students.components.html` by replacing the html code with the given code

```
<div *ngFor="let student of students" class="row">
    <div class="panel panel-primary col-md-offset-2 col-md-6" >
            <div class="panel-heading">
            <h2 class="panel-title">{{student.studentId }}</h2>
            </div>
            <div class="panel-body row">
                <div class="col-md-6">
                <p class="student-name"> {{student. name | uppercase}}
                       {{student.surname}}</p>
                <p *ngIf="student.gpa > 2.5">Good Student who get grade
                       {{student.gpa | number:'1.2-2'}}</p>
                <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
                       {{student.gpa  | number:'1.2-2'}}</p>
                </div>
                <div class="col-md-4">
                     <img [src]="student.image" [title] = "student.name">
                </div>
            </div>
    </div>
</div>

<div class="alert alert-success col-md-offset-2 col-md-6"> The average gpa
                       is {{averageGpa()}} </div>
```

7.7. Where can you check for the Bootstraps api?

.....

.....

Run the application

Lecture's Signature _____