Creating an API using get method

FRONT END

Step:1

Create a component with name **view-student** and add the below code

view-student.html

```
<table style="border:1px solid;text-align:center">

  <thead>
  <tr>
  <th>Id </th>
  <th>First Name</th>
  <th>Last Name</th>
  <th>Address</th>
  <th>Age </th>
  <th>Class</th>
  <th>Section</th>
  </tr>
  </thead>
  <tbody>
  <tr *ngFor="let Student of stud; let $index  = index">
  <td><input [(ngModel)]="Student.id" [disabled]='$index !== expandedIndex'value= {{Student.id}}></td>
  <td><input [(ngModel)]="Student.firstname" [disabled]='$index !== expandedIndex' value={{Student.firstname}}></td>
  <td><input [(ngModel)]="Student.lastname" [disabled]='$index !== expandedIndex' value={{Student.lastname}}></td>
  <td><input [(ngModel)]="Student.address" [disabled]='$index !== expandedIndex' value={{Student.address}}></td>
  <td><input [(ngModel)]="Student.age" [disabled]='$index !== expandedIndex' value={{Student.age}}></td>
  <td><input [(ngModel)]="Student.class" [disabled]='$index !== expandedIndex' value={{Student.class}}></td>
  <td><input [(ngModel)]="Student.section" [disabled]='$index !== expandedIndex' value={{Student.section}}></td>
  </tr>

  </tbody>
  </table>
```

Step:2

Import the service file to view-student.ts file as below and write a promise in getdetails()

view-student.ts

```
import { Component, OnInit } from '@angular/core';
import { ApiService } from '../api.service';
import { Student } from '../create-student/student';
@Component({
  selector: 'app-view-students',
  templateUrl: './view-students.component.html',
  styleUrls: ['./view-students.component.css']
})
export class ViewStudentsComponent implements OnInit {
  stud
  val:boolean = true ;
  expandedIndex: any
  constructor(private service: ApiService) { }
  ngOnInit() {
    console.log('hello');
   this.getAllStudents();
  }
//gets data from db
  getAllStudents () {
    console.log('hi');
    this.service.getstudents().subscribe((res: any) => {
      this.stud = res
      console.log(res);
    }, err => {
      console.log(err);
    })
  }
  }
```

Step:3

Add an read API by using get method as below to the existing service file

api.service.ts

```
getstudents(){
  return this._http.get(this.rootUrl+"read/read", {
    observe: 'body'
});
}
```

Step:4 update the app.module.ts by import the view-student component as below

```
import   BrowserModule   from '@angular/platform-browser'
import   NgModule   from '@angular/core'
import   NgbModule   from '@ng-bootstrap/ng-bootstrap'
import   AppRoutingModule   from './app-routing.module'
import   AppComponent   from './app.component'
import   CreateStudentComponent   from './create-student/create-student.component'
import   FormsModule  ReactiveFormsModule   from '@angular/forms'
 import   HttpModule   from '@angular/http'
import   HttpClientModule   from '@angular/common/http'
import   ViewStudentsComponent   from './view-students/view-students.component'
```

```
  NgModule
 declarations:
   AppComponent
   CreateStudentComponent
   ViewStudentsComponent


 imports:
   BrowserModule
   NgbModule
   HttpModule
   HttpClientModule
   FormsModule

   ReactiveFormsModule
   AppRoutingModule

 providers:
 bootstrap:  AppComponent

export class AppModule
```

Step:5 add button with routerlink for profile in app.component.html with below code to display the buttons with router link

```
    <button>
            <a  [routerLink]="['/edit']">

              <p>profile</p>
            </a>
        </button>
```

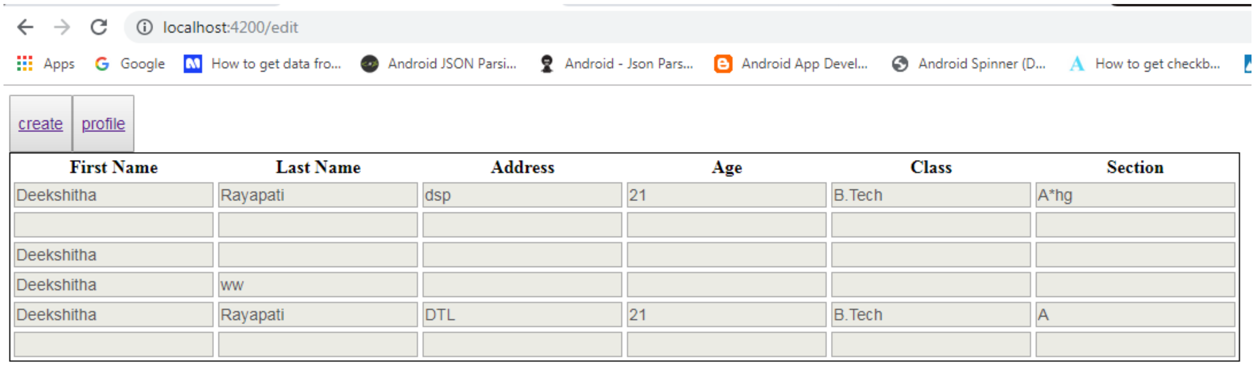
Step:6 add routing in app-routing.module.ts as below

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { CreateStudentComponent } from './create-student/create-student.component';
import { ViewStudentsComponent } from './view-students/view-students.component'
const routes: Routes = [
 { path: 'create',    component: CreateStudentComponent },
 { path: 'edit',    component: ViewStudentsComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})export class AppRoutingModule { 
```

The UI part of the above code is

1.This is the screen we get when we click on profile button it will displays all the student details



First Name	Last Name	Address	Age	Class	Section
Deekshitha	Rayapati	dsp	21	B.Tech	A*hg
Deekshitha					
Deekshitha	ww				
Deekshitha	Rayapati	DTL	21	B.Tech	A

BACK END

Step 1: update/replace the below code in index.js file

```
var express = require('express');
var path = require('path');

var createRouter = require('./routes/create');
var readRouter = require('./routes/read');
var editRouter = require('./routes/edit');
var bodyParser = require('body-parser')
var app = express();
app.use(bodyParser.urlencoded({ extended: false }))

// parse application/json
app.use(bodyParser.json())
// add mongoose
var mongoose = require('mongoose');
mongoose.connect("mongodb://127.0.0.1:27017/training", {useNewUrlParser: true, us
eUnifiedTopology: true}, (error, client) => {
  if(error){
      console.log("Error occured not able to connect");
  }
  });


//add cors
var cors = require('cors');
app.use(cors({
  origin:'http://localhost:4200'
}));



// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use('/create', createRouter);
app.use('/read', readRouter);


// catch 404 and forward to error handler
app.use(function(req, res, next) {
    next(createError(404));
  });

// error handler
app.use(function(err, req, res, next) {
    // set locals, only providing error in development
    res.locals.message = err.message;
    res.locals.error = req.app.get('env') === 'development' ? err : {};

    // render the error page
    res.status(err.status || 500);
    res.render('error');
  });

//server listening
app.listen(3005, ()=> console.log("listening on port 3005"));
module.exports = app;
```

Step 2: create a file in routes folder with name read.js and place the below code in that file

```
var express = require('express');
var router = express.Router();
var ReadStudent = require('../models/C-stud');
router.get('/read', function (req, res) {
    // find is used to retrieve data from db
    ReadStudent.find(function (err, result) {
        if (err) {
            res.send('Failed to fetch data');
        }
        else {
            result =
                res.json(result);
        }
    })
    console.log("reached get api");
})

module.exports = router;
```

Step 3: start the server by using **node index.js** **command in terminal**