

FRONT END

Creating an API using delete method

Step:1

Update **view-student.html** file by adding 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 data of stud; let $index = index">
      <!--
      <td><input [(ngModel)]="Student.id" [disabled]='$index !== expandedIndex' value=
      {{Student.id}}></td> -->
      <td><input [(ngModel)]="data.firstname" [disabled]='$index !== expandedIndex' value={{data.firstname}}></td>
      <td><input [(ngModel)]="data.lastname" [disabled]='$index !== expandedIndex' value={{data.lastname}}></td>
      <td><input [(ngModel)]="data.address" [disabled]='$index !== expandedIndex' value={{data.address}}></td>
```

```

    <td><input [(ngModel)]="data.age" [disabled]='$index !== expandedIndex' value={
{data.age}}></td>
    <td><input [(ngModel)]="data.class" [disabled]='$index !== expandedIndex' value
={{data.class}}></td>
    <td><input [(ngModel)]="data.section" [disabled]='$index !== expandedIndex' val
ue={{data.section}}></td>
    <td> <button (click)="edit(data,$index)">edit</button></td>
    <td><button (click)="delete(data)">Delete</button></td>
    <td><button (click)="save(data)">save</button></td>
  </tr>

</tbody>
</table>

```

Step:2

Update the view-student.ts file as below and write a promise in delete()

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) { }

  delete(val){
    var v={
      "_id":val._id
    }
    console.log(v)
    this.service.delrow(v).subscribe(res => {

```

```

        console.log(res)
    })
}
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);
    })
}
//edit the data
public edit(data, index) {
    this.expandedIndex = index;
}
//saves edited data
public save(data) {
    this.service.add(data).subscribe((data) => {
        console.log(data)
        this.val = true
    })
}
}

```

Step:3

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

api.service.ts

```
public delrow(v){
  return this._http.delete(this.rootUrl+"read/del", {
    observe: 'body',
    params: v
  });
}
```

Step:4 run the project

The output of the above code is

1.This is the screen we get, when we click on delete button that particular row can will be deleted from database

create profile

First Name	Last Name	Address	Age	Class	Section			
Deekshitha	Rayapati	dsp	21	B.Tech	A*	edit	Delete	save
						edit	Delete	save
Deekshitha						edit	Delete	save
Deekshitha	ww					edit	Delete	save
Deekshitha	Rayapati	DTL	21	B.Tech	A	edit	Delete	save
ms	dhoni	DTL	50	B.Tech	A	edit	Delete	save
						edit	Delete	save

Angular is running in the development mode. Call enableProdMode() to enable the production mode.
hello
hi
▶ (7) [[-], [-], [-], [-], [-], [-], [-]]
[WDS] Live Reloading enabled.

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, useUnifiedTopology: true}, (error, client) => {
  if(error){
    console.log("Error occurred 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);
app.use('/edit', editRouter);

// 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: update/replace a file 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");
})
router.delete('/del', (req, res) => {
  // console.log(req)
  if (!req.query._id)
    return res.status(400).send(`No record with given id : ${req.query._id}`);

  ReadStudent.findByIdAndRemove(req.query._id, (err, doc) => {

```

```
        if (!err) { res.send(doc); }  
        else { console.log('Error in Employee Delete :' + JSON.stringify(err, undefined, 2)); }  
    });  
});  
  
module.exports = router;
```

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