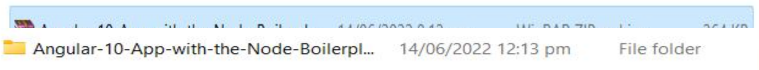
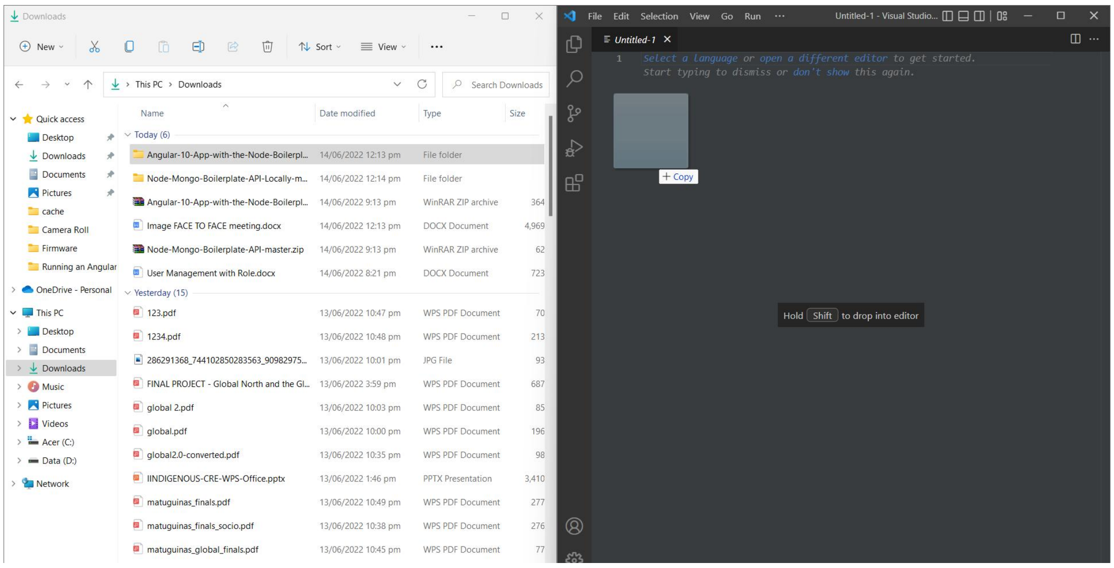
Angular-10-App-with-the-Node-Boilerplate-API-master

1. **Go to Github.com download the zip file and Extract the File**



Name	Date modified	Type
Angular-10-App-with-the-Node-Boilerpl...	14/06/2022 12:13 pm	File folder

2. **Open you Visual Studio Code as your text editor and drag the extracted file after dragging to to terminal and type "npm install"**

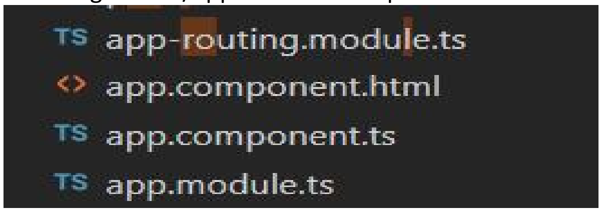


Name	Date modified	Type	Size
Today (6)			
Angular-10-App-with-the-Node-Boilerpl...	14/06/2022 12:13 pm	File folder	
Node-Mongo-Boilerplate-API-Locally-m...	14/06/2022 12:14 pm	File folder	
Angular-10-App-with-the-Node-Boilerpl...	14/06/2022 9:13 pm	WinRAR ZIP archive	364
Image FACE TO FACE meeting.docx	14/06/2022 12:13 pm	DOCX Document	4,969
Node-Mongo-Boilerplate-API-master.zip	14/06/2022 9:13 pm	WinRAR ZIP archive	62
User Management with Role.docx	14/06/2022 8:21 pm	DOCX Document	723
Yesterday (15)			
123.pdf	13/06/2022 10:47 pm	WPS PDF Document	70
1234.pdf	13/06/2022 10:48 pm	WPS PDF Document	213
286291368_744102850283563_90982975...	13/06/2022 10:01 pm	JPG File	93
FINAL PROJECT - Global North and the Gl...	13/06/2022 3:59 pm	WPS PDF Document	687
global 2.pdf	13/06/2022 10:03 pm	WPS PDF Document	85
global.pdf	13/06/2022 10:00 pm	WPS PDF Document	196
global2.0-converted.pdf	13/06/2022 10:35 pm	WPS PDF Document	98
IINDIGENOUS-CRE-WPS-Office.pptx	13/06/2022 1:46 pm	PPTX Presentation	3,410
matuguinas_finals.pdf	13/06/2022 10:49 pm	WPS PDF Document	277
matuguinas_finals_socio.pdf	13/06/2022 10:38 pm	WPS PDF Document	276
matuguinas_global_finals.pdf	13/06/2022 10:45 pm	WPS PDF Document	77

```
Select a language or open a different editor to get started.
Start typing to dismiss or don't show this again.
```

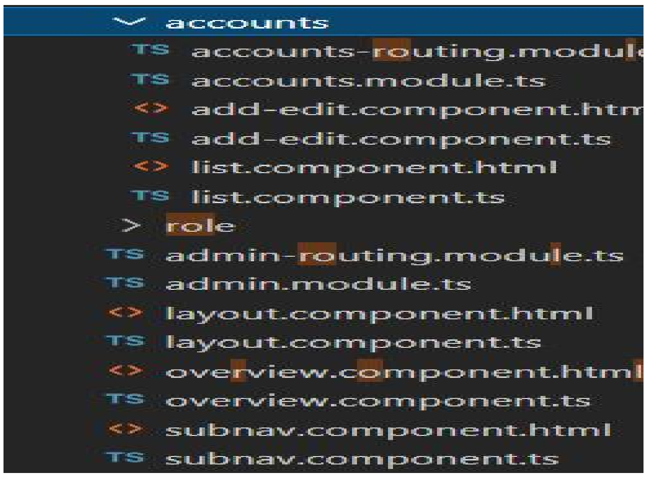
Hold Shift to drop into editor

3. First go to scr/appctrl + f or find profile



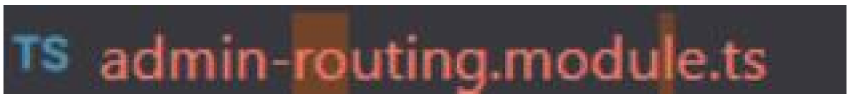
```
TS app-routing.module.ts
<> app.component.html
TS app.component.ts
TS app.module.ts
```

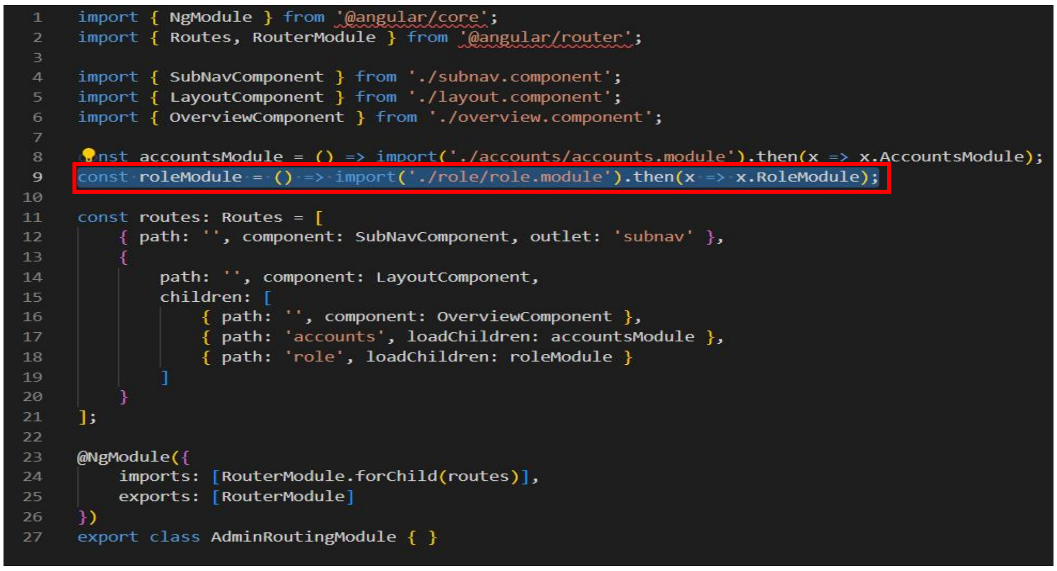
4. Add folder roles in the admin and copy the same from accounts to your role folder
You can edit the files and code and make changes in your account role



```
∨ accounts
    TS accounts-routing.module
    TS accounts.module.ts
    <> add-edit.component.htm
    TS add-edit.component.ts
    <> list.component.html
    TS list.component.ts
  > role
  TS admin-routing.module.ts
  TS admin.module.ts
  <> layout.component.html
  TS layout.component.ts
  <> overview.component.html
  TS overview.component.ts
  <> subnav.component.html
  TS subnav.component.ts
```

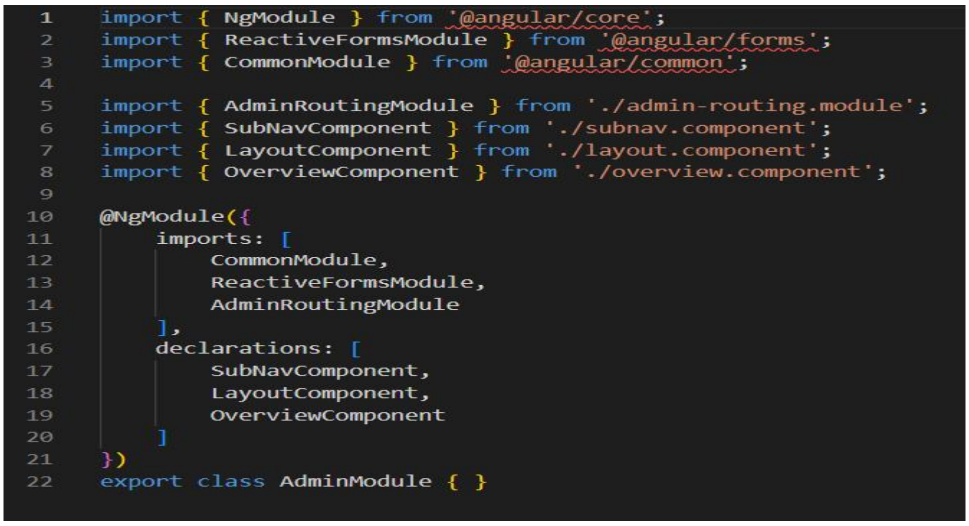
5. Lets Make some changes , Find admin-routing.module.ts and add the code " const roleModule = () => import('./role/role.module').then(x => x.RoleModule); "





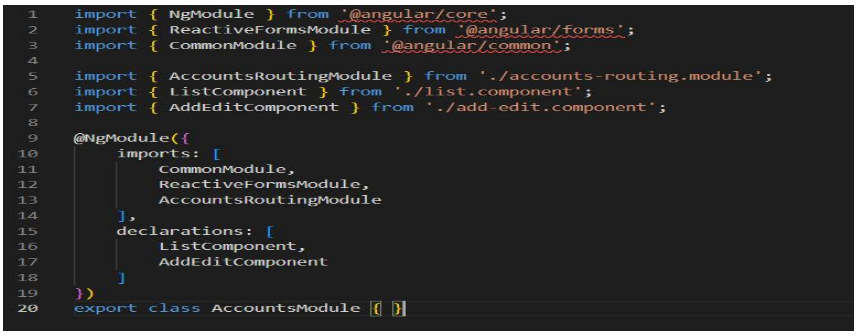
```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { SubNavComponent } from './subnav.component';
import { LayoutComponent } from './layout.component';
import { OverviewComponent } from './overview.component';

const accountsModule = () => import('./accounts/accounts.module').then(x => x.AccountsModule);
const roleModule = () => import('./role/role.module').then(x => x.RoleModule);

const routes: Routes = [
    { path: '', component: SubNavComponent, outlet: 'subnav' },
    {
        path: '', component: LayoutComponent,
        children: [
            { path: '', component: OverviewComponent },
            { path: 'accounts', loadChildren: accountsModule },
            { path: 'role', loadChildren: roleModule }
        ]
    }
];

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

6. Your role.module.ts should be



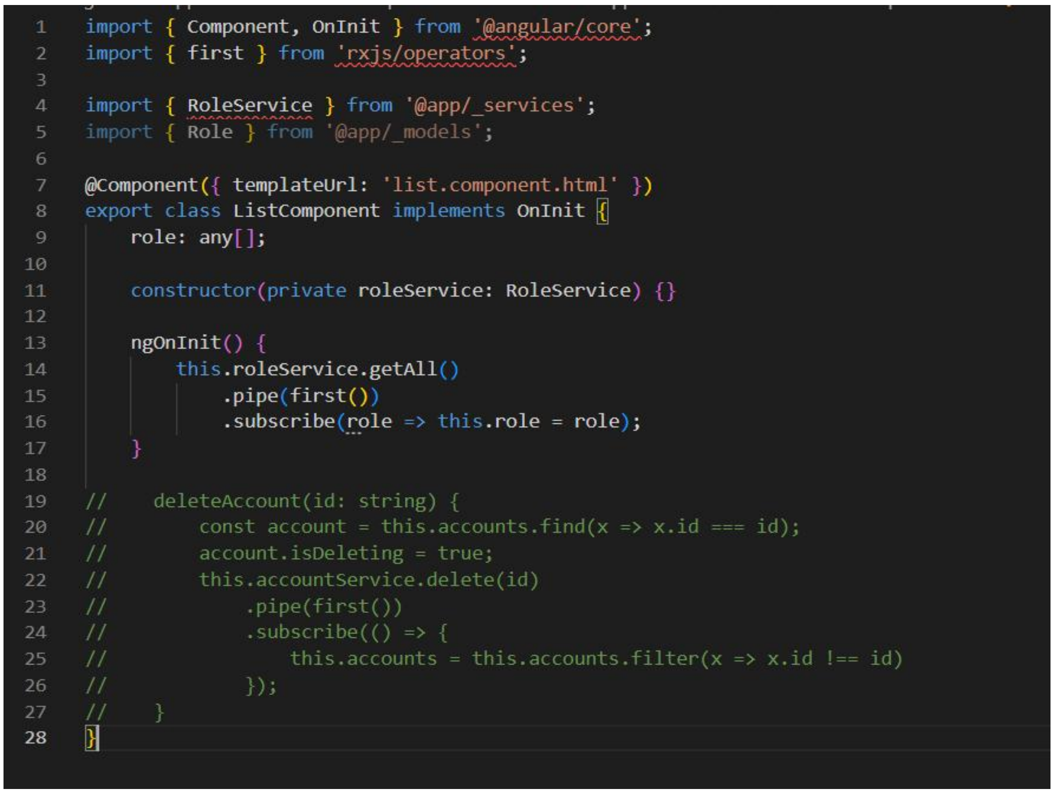
```
import { NgModule } from '@angular/core';
import { ReactiveFormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';

import { AdminRoutingModule } from './admin-routing.module';
import { SubNavComponent } from './subnav.component';
import { LayoutComponent } from './layout.component';
import { OverviewComponent } from './overview.component';

@NgModule({
    imports: [
        CommonModule,
        ReactiveFormsModule,
        AdminRoutingModule
    ],
    declarations: [
        SubNavComponent,
        LayoutComponent,
        OverviewComponent
    ]
})
export class AdminModule { }
```

7. Your role.routing.module.ts



```
import { NgModule } from '@angular/core';
import { ReactiveFormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';

import { AccountsRoutingModule } from './accounts-routing.module';
import { ListComponent } from './list.component';
import { AddEditComponent } from './add-edit.component';

@NgModule({
    imports: [
        CommonModule,
        ReactiveFormsModule,
        AccountsRoutingModule
    ],
    declarations: [
        ListComponent,
        AddEditComponent
    ]
})
export class AccountsModule { }
```

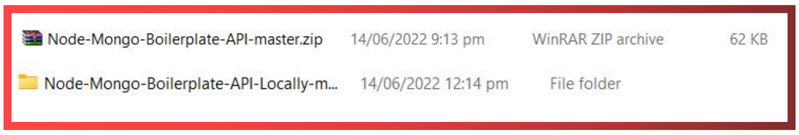
8. List.component.ts



```
import { Component, OnInit } from '@angular/core';
import { first } from 'rxjs/operators';

import { RoleService } from '@app/_services';
import { Role } from '@app/_models';

@Component({ templateUrl: 'list.component.html' })
export class ListComponent implements OnInit {
    role: any[];

    constructor(private roleService: RoleService) {}

    ngOnInit() {
        this.roleService.getAll()
            .pipe(first())
            .subscribe(role => this.role = role);
    }

//    deleteAccount(id: string) {
//        const account = this.accounts.find(x => x.id === id);
//        account.isDeleting = true;
//        this.accountService.delete(id)
//            .pipe(first())
//            .subscribe(() => {
//                this.accounts = this.accounts.filter(x => x.id !== id)
//            });
//    }
}
```

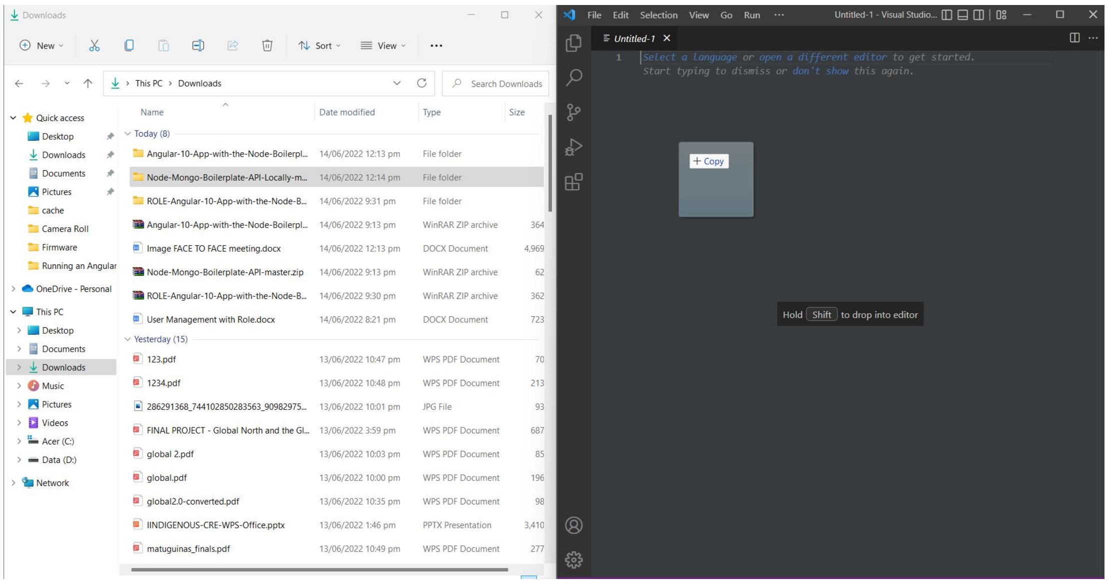
USER MANAGEMENT WITH WELL DEFINE ROLES USING Node-Mongo-Boilerplate-API-Locally

1.**Go to Github.com download the zip file and Extract the File**



Node-Mongo-Boilerplate-API-master.zip	14/06/2022 9:13 pm	WinRAR ZIP archive	62 KB
Node-Mongo-Boilerplate-API-Locally-m...	14/06/2022 12:14 pm	File folder	

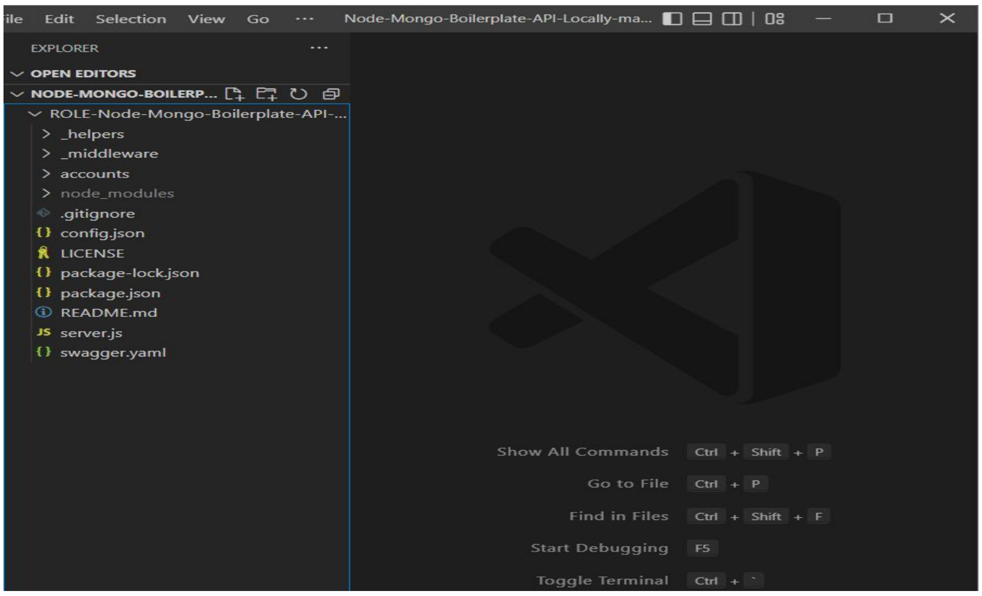
2. Open Visual Studio Code and Drag the extracted file



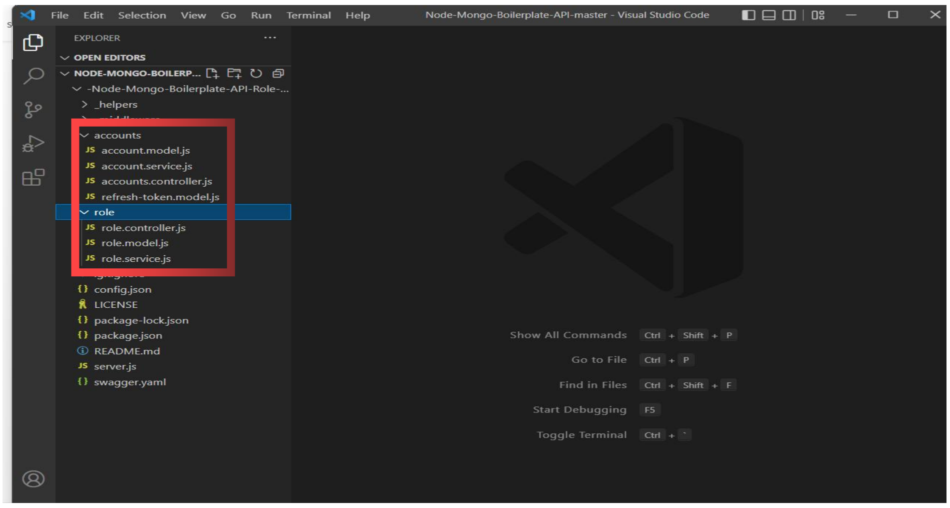
Name	Date modified	Type	Size
Today (8)			
Angular-10-App-with-the-Node-Boilerpl...	14/06/2022 12:13 pm	File folder	
Node-Mongo-Boilerplate-API-Locally-m...	14/06/2022 12:14 pm	File folder	
ROLE-Angular-10-App-with-the-Node-B...	14/06/2022 9:31 pm	File folder	
Angular-10-App-with-the-Node-Boilerpl...	14/06/2022 9:13 pm	WinRAR ZIP archive	364
Image FACE TO FACE meeting.docx	14/06/2022 12:13 pm	DOCX Document	4,969
Node-Mongo-Boilerplate-API-master.zip	14/06/2022 9:13 pm	WinRAR ZIP archive	62
ROLE-Angular-10-App-with-the-Node-B...	14/06/2022 9:30 pm	WinRAR ZIP archive	362
User Management with Role.docx	14/06/2022 8:21 pm	DOCX Document	723
Yesterday (15)			
123.pdf	13/06/2022 10:47 pm	WPS PDF Document	70
1234.pdf	13/06/2022 10:48 pm	WPS PDF Document	213
286291368_744102850283563_90982975...	13/06/2022 10:01 pm	JPG File	93
FINAL PROJECT - Global North and the Gl...	13/06/2022 3:59 pm	WPS PDF Document	687
global 2.pdf	13/06/2022 10:03 pm	WPS PDF Document	85
global.pdf	13/06/2022 10:00 pm	WPS PDF Document	196
global2.0-converted.pdf	13/06/2022 10:35 pm	WPS PDF Document	98
IINDIGENOUS-CRE-WPS-Office.pptx	13/06/2022 1:46 pm	PPTX Presentation	3,410
matuguinas_finals.pdf	13/06/2022 10:49 pm	WPS PDF Document	277

```
Select a language or open a different editor to get started.
Start typing to dismiss or don't show this again.
```

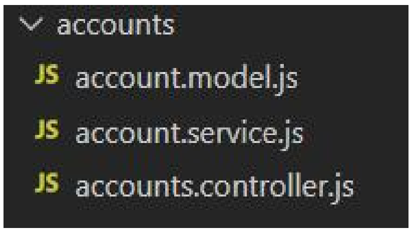
3. This is the file when you drag it , automatically open the file

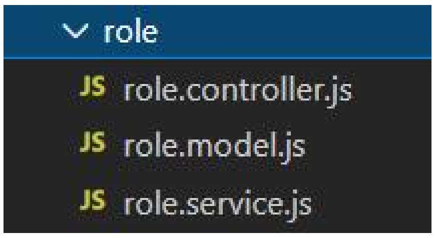


4. Open the terminal and type “npm install “ after successfully installed Copy the account file renamed it role and Replace the account name into “ role”

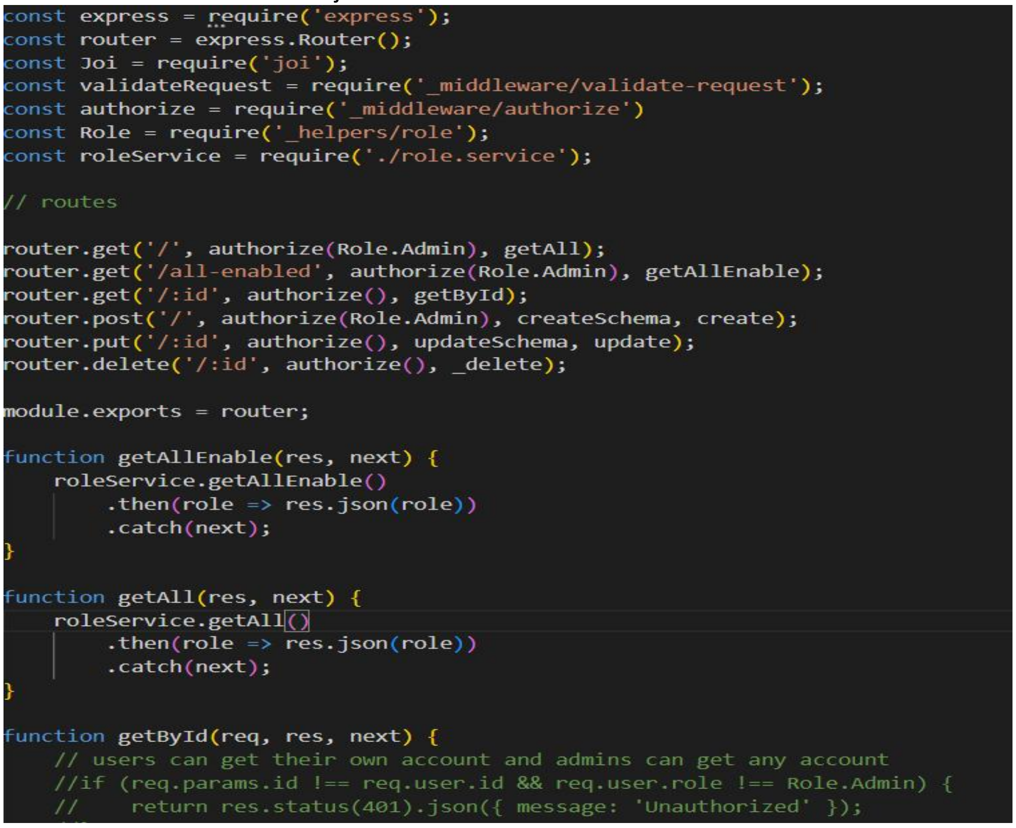


5. Here is the changes

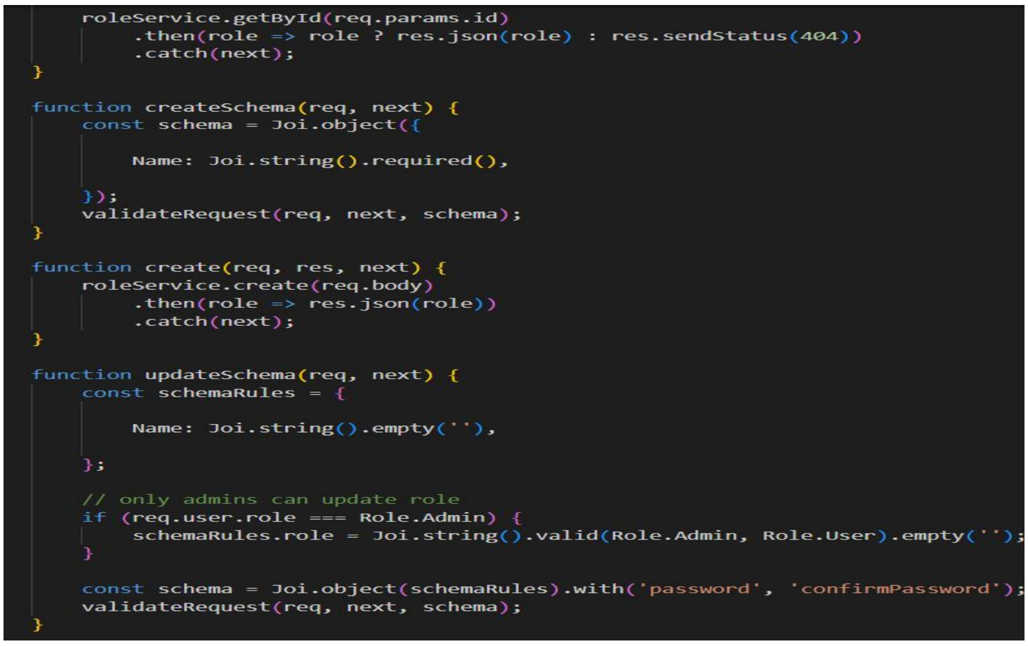




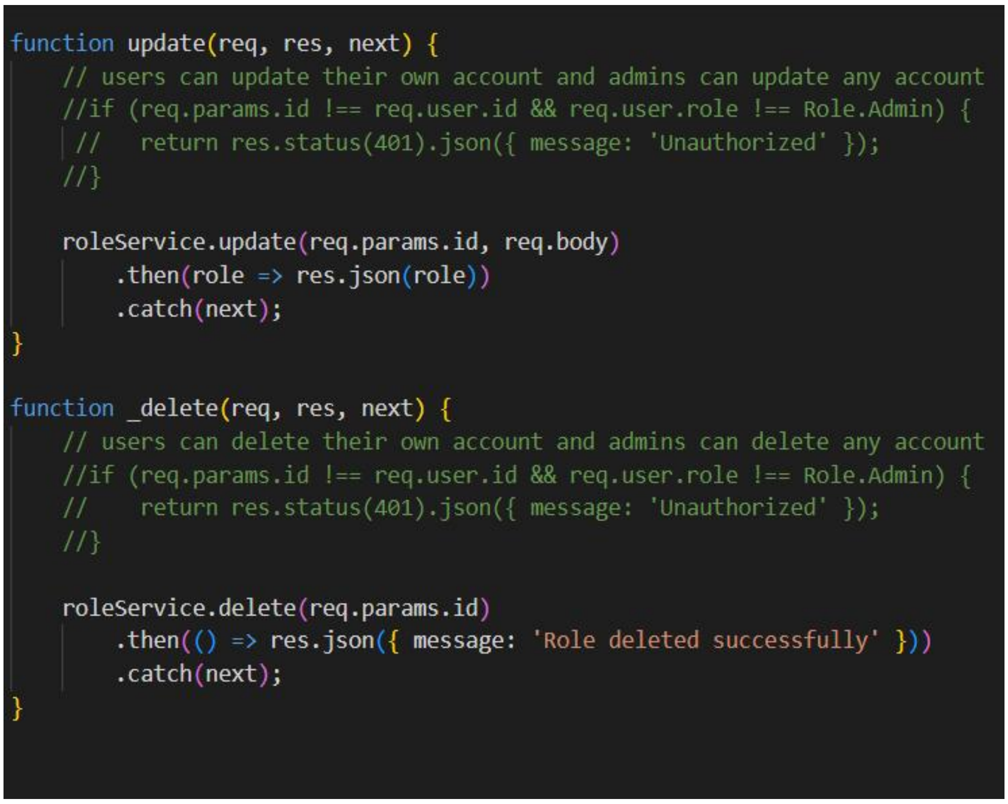
Screenshot of the role.control.js



```
const express = require('express');
const router = express.Router();
const Joi = require('joi');
const validateRequest = require('_middleware/validate-request');
const authorize = require('_middleware/authorize')
const Role = require('_helpers/role');
const roleService = require('./role.service');

// routes

router.get('/', authorize(Role.Admin), getAll);
router.get('/all-enabled', authorize(Role.Admin), getAllEnable);
router.get('/:id', authorize(), getById);
router.post('/', authorize(Role.Admin), createSchema, create);
router.put('/:id', authorize(), updateSchema, update);
router.delete('/:id', authorize(), _delete);

module.exports = router;

function getAllEnable(res, next) {
    roleService.getAllEnable()
        .then(role => res.json(role))
        .catch(next);
}

function getAll(res, next) {
    roleService.getAll()
        .then(role => res.json(role))
        .catch(next);
}

function getById(req, res, next) {
    // users can get their own account and admins can get any account
    //if (req.params.id !== req.user.id && req.user.role !== Role.Admin) {
    //     return res.status(401).json({ message: 'Unauthorized' });
```

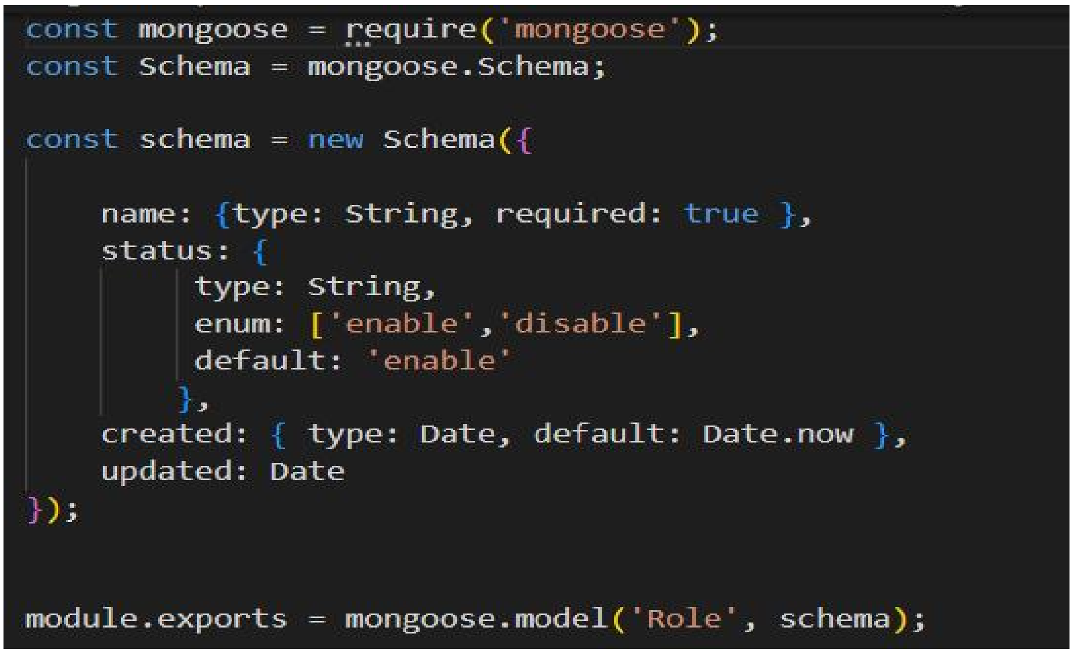


```
    roleService.getById(req.params.id)
        .then(role => role ? res.json(role) : res.sendStatus(404))
        .catch(next);
}

function createSchema(req, next) {
    const schema = Joi.object({

        Name: Joi.string().required(),

    });
    validateRequest(req, next, schema);
}

function create(req, res, next) {
    roleService.create(req.body)
        .then(role => res.json(role))
        .catch(next);
}

function updateSchema(req, next) {
    const schemaRules = {

        Name: Joi.string().empty(''),

    };

    // only admins can update role
    if (req.user.role === Role.Admin) {
        schemaRules.role = Joi.string().valid(Role.Admin, Role.User).empty('');
    }

    const schema = Joi.object(schemaRules).with('password', 'confirmPassword');
    validateRequest(req, next, schema);
}
```



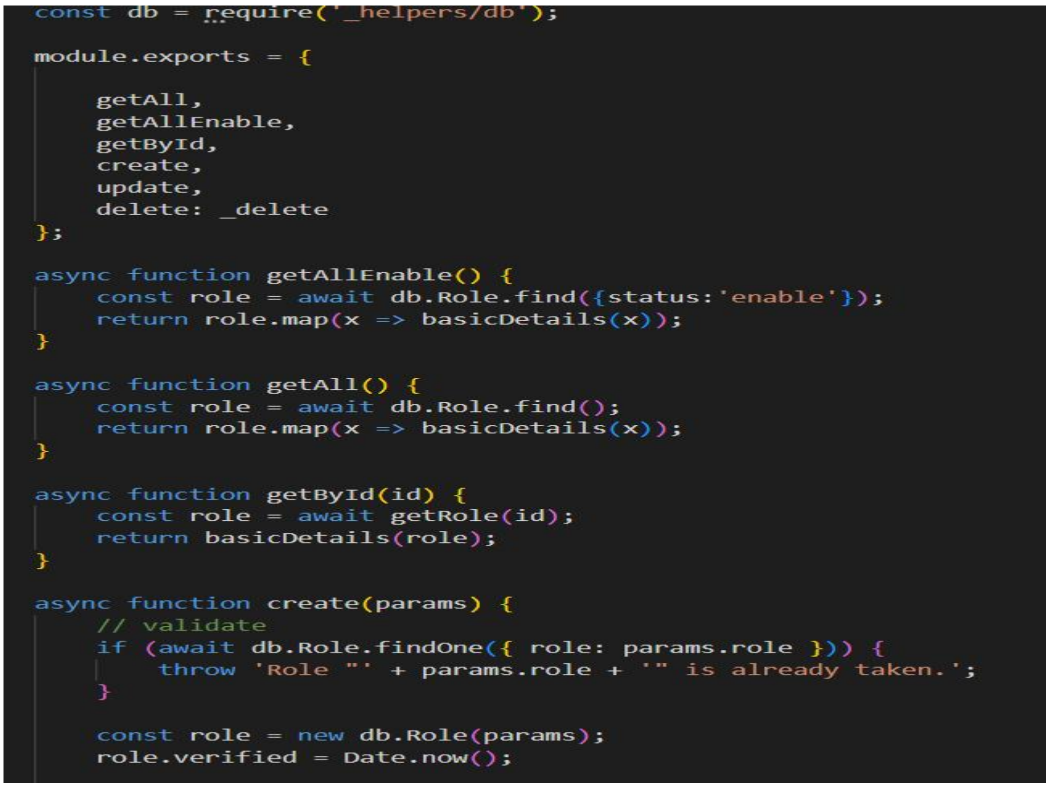
```
function update(req, res, next) {
    // users can update their own account and admins can update any account
    //if (req.params.id !== req.user.id && req.user.role !== Role.Admin) {
    //    return res.status(401).json({ message: 'Unauthorized' });
    //}

    roleService.update(req.params.id, req.body)
        .then(role => res.json(role))
        .catch(next);
}

function _delete(req, res, next) {
    // users can delete their own account and admins can delete any account
    //if (req.params.id !== req.user.id && req.user.role !== Role.Admin) {
    //    return res.status(401).json({ message: 'Unauthorized' });
    //}

    roleService.delete(req.params.id)
        .then(() => res.json({ message: 'Role deleted successfully' }))
        .catch(next);
}
```

Screenshot of role.model.js



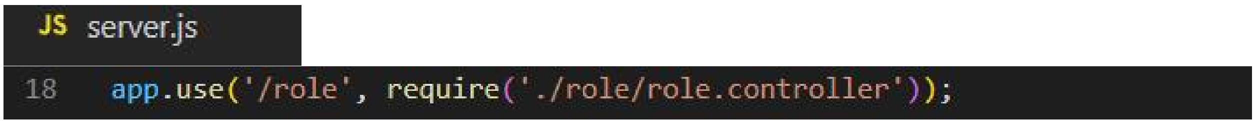
```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const schema = new Schema({

    name: {type: String, required: true },
    status: {
            type: String,
            enum: ['enable','disable'],
            default: 'enable'
        },
    created: { type: Date, default: Date.now },
    updated: Date
});


module.exports = mongoose.model('Role', schema);
```

Screenshot of role.service.js



```
const db = require('_helpers/db');

module.exports = {

    getAll,
    getAllEnable,
    getById,
    create,
    update,
    delete: _delete
};

async function getAllEnable() {
    const role = await db.Role.find({status:'enable'});
    return role.map(x => basicDetails(x));
}

async function getAll() {
    const role = await db.Role.find();
    return role.map(x => basicDetails(x));
}

async function getById(id) {
    const role = await getRole(id);
    return basicDetails(role);
}

async function create(params) {
    // validate
    if (await db.Role.findOne({ role: params.role })) {
        throw 'Role "' + params.role + '" is already taken.';
    }

    const role = new db.Role(params);
    role.verified = Date.now();
```

Edit with the code



```
app.use('/role', require('./role/role.controller'));
```

Your done with the API