how to stop live server in angular

ctrl + c

Installing the tools required

Node : Install the latest version of node. Here is the link to download and install the latest version

https://nodejs.org/en/download/

To check the version of node

node -v

Angular CLI : Install the latest version of Angular CLI by executing the following command from the windows command prompt.

npm install -g @angular/cli

To verify the version of Angular CLI installed on your machine execute the following command.

ng -v

Another question that you might have is, which version of Angular are we using with this course. As of this recording, the latest version is Angular 5, which is what we will be using.

After you have the latest version of Node and Angular CLI installed, launch windows command prompt as an administrator and execute the following command. This creates a new AngularProject with name AngularCrud. We do not want test files to be generated for the root component AppComponent, so we have set "skip-tests" option to true. We will discuss unit testing components in a later video.

ng new AngularCrud --skip-tests true

Once you have the Angular project opened in Visual Studio Code, open package.json file and notice that we are using Angular 5.

We will be using Bootstrap for styles in our application. So install Bootstrap by executing the following command from the command prompt.

npm install bootstrap@3 --save

for opening direct visual source code using below command

" code . "

Once Bootstrap is installed, open .angular-cli.json file and specify the path to the Bootstrap stylesheet (bootstrap.min.css) in the styles property as shown below.

"styles": [

"../node\_modules/bootstrap/dist/css/bootstrap.min.css",

"styles.css"]

==========================

create json file

ng g c name

====

<link rel="stylesheet" href="../node\_modules/bootstrap/dist/css/bootstrap.min.css">

Create a Router:

1. first create componet(including HTML CSS)

ex: ng g c employees/createEmployee --spec false -- flat true

2. open app.module.ts file and import/update the createEmployee

import { BrowserModule } from '@angular/platform-browser';

import { NgModule, Component } from '@angular/core';

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

import { AppComponent } from './app.component';

import { NameComponent } from './name/name.component';

import { ListEmployeesComponent } from './employees/list-employees.component';

import { CreateEmployeeComponent } from './employees/create-employee.component';

const appRoutes: Routes = [

{ path: 'list', component: ListEmployeesComponent},

{ path: 'create', component: CreateEmployeeComponent},

{ path:'', redirectTo: '/list', pathMatch: 'full'}

];

@NgModule({

declarations: [

AppComponent,

NameComponent,

ListEmployeesComponent,

CreateEmployeeComponent,

],

imports: [

BrowserModule,

RouterModule.forRoot(appRoutes)

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

3. updates the app.component.html root file

<div class="container">

<nav class="nav navbar-default">

<ul class="nav navbar-nav">

<li>

<a routerLink = "list">List</a>

</li>

<li>

<a routerLink = "create">Create</a>

</li>

</ul>

</nav>

<router-outlet></router-outlet>

</div>

Changing Base Href:

Method1: first will update manually in index html page

<head>

<meta charset="utf-8">

<title>AngularCrud</title>

<base href="/emp/">

after that when we will run the live server that time we will get error

how to fix the error : give the below command in cmd

ng serve -o --base-href /emp

method 2: Didn't change anything in index HTML page

directly give the command to cmd:

<base href="/">

give the below command in cmd

2.Second way to run Angular href base but coming p unknown error while runs on cmd

ng build -prod --base-href /emp/ (as of now its not working)

09.Angular bootstrap select list.

01.need to mention select in input.

</div>

<div class="form-group">

<label for="department">Department</label>

<select id="department" class="form-control" [(ngModel)]="department" name="department" disabled>

<option value="1" selected>HelpDesk</option>

<option value="2">IT</option>

<option value="H3R">HR</option>

<option value="4">Payroll</option>

</select>

</div>

</div>

as above mention can possible to changes like disable or enable, selected

Note: if [(ngModel)] doesn't support selected so make this changes need to write create-employee.component.html as a "department= "1"";

import { Component, OnInit } from '@angular/core';

import { NgForm } from '../../../node\_modules/@angular/forms';

@Component({

selector: 'app-create-employee',

templateUrl: './create-employee.component.html',

styleUrls: ['./create-employee.component.css']

})

export class CreateEmployeeComponent implements OnInit {

department = "1";

constructor() { }

ngOnInit() {

}

saveEmployee(empForm:NgForm):void{

console.log(empForm.value);

}

}

10.Angular select options from array.

Note: \*ngFor is one of the structural directives changes the DOM , like adding new in DOM, removing from DOM.

11.Angular DatePicker tutorial\\

16.Form Validation

first we will use bootstrap class for validation

has-error

form-control

help-block

for success--> has-success

<div class="form-group" [class.has-error]="fullNameControl.invalid && fullNameControl.touched"

[class.has-success]="fullNameControl.valid">

<label for="fullName" class="control-label">FullName</label>

<input type="text" required id="fullName" name="FullName" [(ngModel)]="FullName" class="form-control"

placeholder="Enter your Name" #fullNameControl ="ngModel" >

<span class="help-block" \*ngIf="fullNameControl.invalid && fullNameControl.touched">

Full Name is required</span>

</div>

for disable button:

first we will check if full form is validate then we will enable the button

<button type="submit" class="btn btn-primary" [disabled]="EmployeeForm.invalid">Submit</button>

Email Validation:

First we will make required validation same as name required validation after we will add email validation

<div class="form-group" [class.has-error]="email.invalid && email.touched" [class.has-success]="email.valid">

<label for="email" class="control-label">Email</label>

<input type="text" id="email" class="form-control" required email #email="ngModel" [(ngModel)]="employee.Email" name="Email"

placeholder="Enter your Email ID">

<span class="help-block" \*ngIf="email.errors?.required && email.touched" >Email is required!</span>

<span class="help-block" \*ngIf="email.errors?.email && email.touched" >Email is Invalid!</span>

</div>

Custom Email Validation:

in place of email valuator will use pattern validator

<div class="form-group" [class.has-error]="email.invalid && email.touched" [class.has-success]="email.valid">

<label for="email" class="control-label">Email</label>

<input type="text" id="email" class="form-control" required

pattern="^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$" #email="ngModel" [(ngModel)]="employee.Email" name="Email"

placeholder="Enter your Email ID">

<span class="help-block" \*ngIf="email.errors?.required && email.touched" >Email is required!</span>

<span class="help-block" \*ngIf="email.errors?.pattern && email.touched" >Email is Invalid!</span>

</div>