Liên kết khác   toilati123vn@gmail.com   Bảng điều khiển   Đăng xuất

Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

Support us   **.Net Basics**   **C#**   **SQL**   **ASP.NET**   **ADO.NET**   **MVC**   **Slides**   **C# Programs**   **Subscribe**   **Buy DVD**

Angular 6 routing tutorial

Suggested Videos
Part 1 - Creating angular 6 project | Text | Slides
Part 2 - Install Bootstrap for Angular 6 | Text | Slides

In this video we will discuss **setting up routing in a separate routing module**.

At the moment, in our angular project, we only have one component and that is the root component - **AppComponent**. Now, let's create the following 2 components, so we can setup routing to navigate between these components.

- CreateEmployeeComponent
- ListEmployeesComponent

Use the following 2 Angular CLI commands to create the components.

ng g c employee/create-employee --spec=false --flat=true
ng g c employee/list-employees --spec=false --flat=true

Step 1 : Set `<base href="/">` in index.html : When setting up routing in an angular application, the first step is to set the base path using the base href element. The base

PRAGIM
TECHNOLOGIES
Training + Placements

Best software training and placements in marathahalli, bangalore. For further details please call 09945699393.

Complete Tutorials

JavaScript tutorial

Bootstrap tutorial

Angular tutorial for beginners

Angular 5 Tutorial for beginners

Important Videos

The Gift of Education

Web application for your business

path tells the angular router, how to compose navigation URLs. When you create a new Angular 6 project, this is included automatically by the Angular CLI. To understand the significance of this base element, please watch Part 4 from Angular CRUD tutorial.

Step 2 : Create a separate routing module : The first question that comes to our mind is - Why should we define routes in a separate routing module.
Well, for separation of concerns and maintainability. If routing is in it's own module, it is easier to find and change routing code if required.

Now, using the Angular CLI, let's create the routing module. By convention, the routing module class name is called AppRoutingModule and the file is named app-routing.module.ts.

```
ng generate module app-routing --flat=true –module=app
```

–module=app tells the Angular CLI to import and register routing module in the application root module AppModule.

The above command can also be written using it's short-hand notation as shown below

```
ng g m app-routing --flat=true -m=app
```

Step 3 : Import the Angular RouterModule into the AppRoutingModule and configure the application routes : Here is the modified AppRoutingModule file (app-routing.module.ts). Please note that, the CommonModule is not required in the routing module, so I have deleted it's reference. We generally don't declare components in the routing module so, I also deleted declarations array from @NgModule decorator. The rest of the code is commented and self-explanatory.

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

// Import the components so they can be referenced in routes
import { CreateEmployeeComponent } from './employee/create-employee.component';
import { ListEmployeesComponent } from './employee/list-employees.component';
```

```
// The last route is the empty path route. This specifies
// the route to redirect to if the client side path is empty.
const appRoutes: Routes = [
  { path: 'list', component: ListEmployeesComponent },
  { path: 'create', component: CreateEmployeeComponent },
  { path: '', redirectTo: '/list', pathMatch: 'full' }
];


// Pass the configured routes to the forRoot() method
// to let the angular router know about our routes
// Export the imported RouterModule so router directives
// are available to the module that imports this AppRoutingModule
@NgModule({
  imports: [ RouterModule.forRoot(appRoutes) ],
  exports: [ RouterModule ]
})
export class AppRoutingModule { }
```

Step 4 : In the application root component file (**app.component.html**), create the navigation menu and tie the configured routes to it. The directive tells the router where to display routed views.

```
<div class="container">
    <nav class="navbar navbar-default">
        <ul class="nav navbar-nav">
            <li>
                <a routerLinkActive="active" routerLink="list">List</a>
            </li>
            <li>
                <a routerLinkActive="active" routerLink="create">Create</a>
            </li>
        </ul>
    </nav>
    <router-outlet></router-outlet>
</div>
```

Next video : Creating a reactive form

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Dot Net Interview Questions

Slides

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

www.PRAGIMTECH.com

CLICK HERE FOR THE FULL
ANGULAR 6 TUTORIAL PLAYLIST

facebook.com/pragimtech | twitter.com/kudevankat

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions
C#

SQL Server

Written Test

1 comment:

Khan Muzaffer March 20, 2019 at 2:29 AM

please upload angular 7 tutorials with web api and database

Reply

```
Enter your comment...
```

Comment as: toilati123vn@g ▼

[Publish]   [Preview]

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

Create a Link

Newer Post   Home   Older Post

Subscribe to: Post Comments (Atom)

Powered by Blogger.