

April 12, 2022

Angular App with Routing Assignment

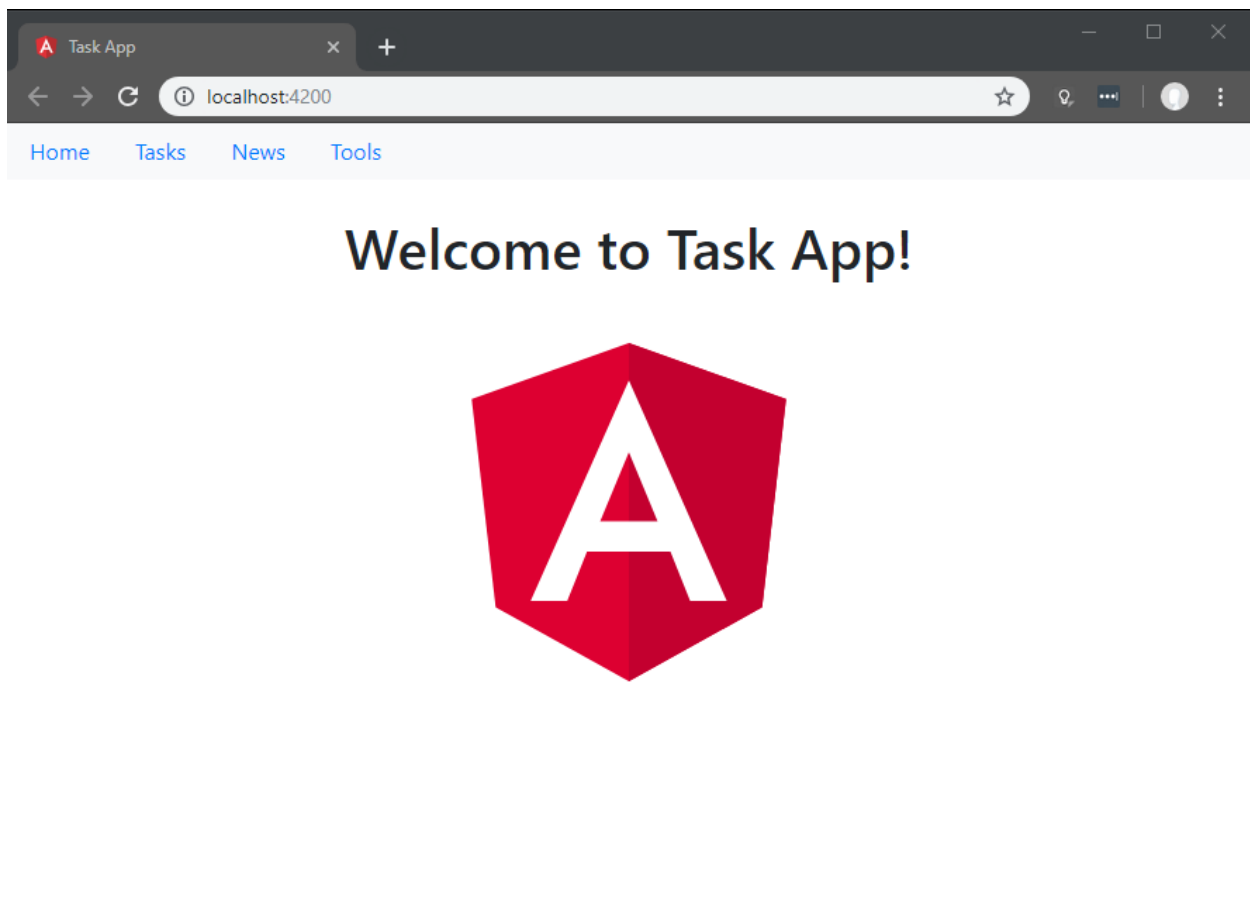
Table of Contents

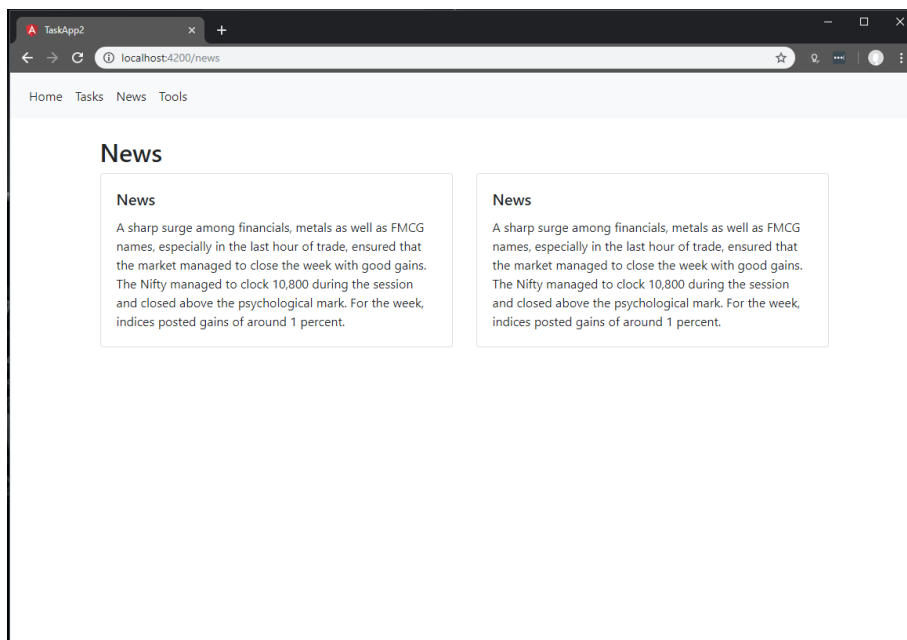
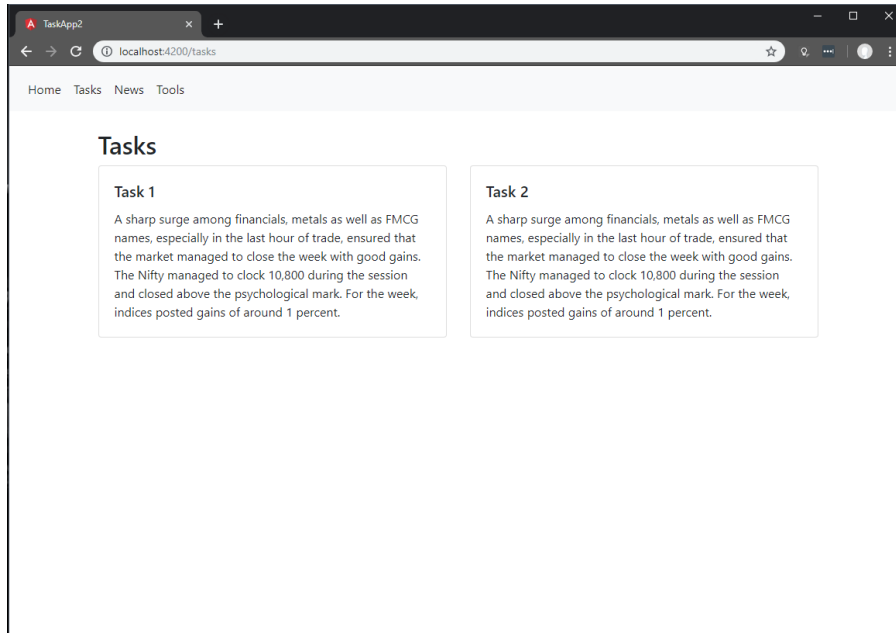
Table of Contents	2
Create the App	5
Install and Configure Bootstrap	5
App Component	5
Home Component.....	6
Remaining Components.....	6

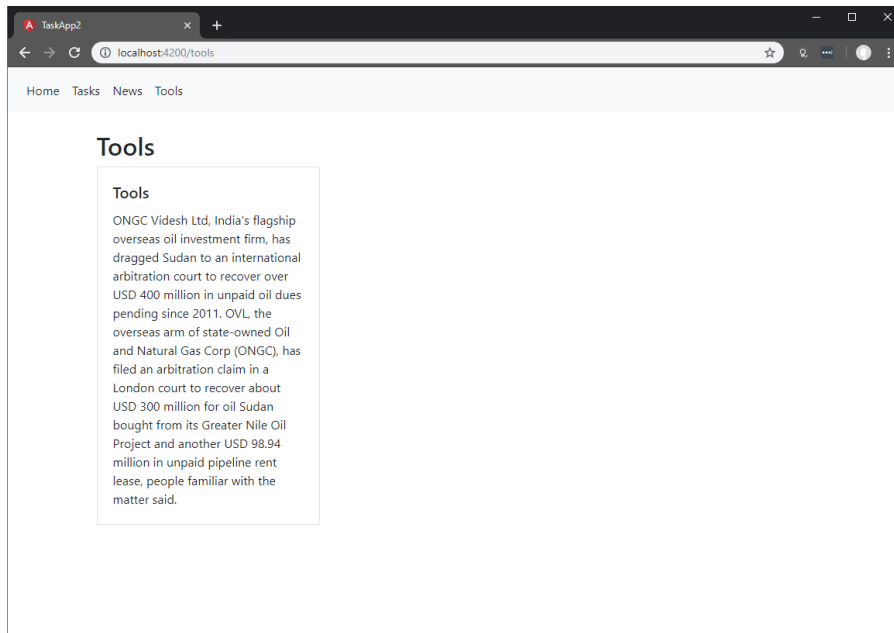
Angular App with Routing Assignment

This task is to create simple app that has 4 main feature components (Home, Tasks, News and Tools) and supports routing between components. Routing utilizes lazy loading of each of the components so that the components will not be downloaded until the user selects the feature. Each of the features just display a simple html page with some data. We will not be building out the features because this assignment is more about getting the application structure in place and getting the routing between components working. In addition, the application will incorporate bootstrap for styling.

Here is what the application looks like.







Create the App

Create a new app using the cli with routing

Install and Configure Bootstrap

Add bootstrap to the application

```
npm install bootstrap
```

Add the following to styles.css

```
@import "~bootstrap/dist/css/bootstrap.min.css";
```

Run the app and you should see the bootstrap styling.

App Component

app.component.html

This page should have a bootstrap top level menu that will router link to the 4 areas/components: Home, Tasks, News and Tools. So essentially the html for app component will be the nav and the router outlet. Wrap the router outlet in a bootstrap div container. Look at the bootstrap site for creating a nav menu with an `` and `` elements or simply with anchors and the appropriate nav classes.

Here is the general layout of the file. The title and the angular logo html will be used for the html for the home component below and you can get rid of the h2 and the set of links that the default page has.

```
<nav>...</nav>  
  
<div container>  
  
<router-outlet></router-outlet>  
  
</div>
```

The nave should have anchors with routerlink attributes for each of the areas " for home, 'tasks', 'news' and 'tools'

Home Component

The Home component is just going to display the home page with a title and the angular logo. Add a home component using the cli in the root folder (--flat) You can get the title and the Welcome message from the current app.component.html. This component should be loaded as the default empty path route in the routing module.

Add a variable for the variable called title that the html template expects.

Remaining Components

For each of the other components, use the cli to

generate a module with routing

then generate a component.

There should be 3 more components: TaskComponent, NewsComponent and ToolsComponent.

They should end up in their own folders. /task, /news, /tools

Add each feature module to the app routing file using the lazy loading syntax loadChildren instead of just specifying the component like you did with the home component. Remember to remove the component declaration from the app module because we want the component lazy loaded and not loaded at the app level.

Add a default empty path in each areas feature module routing class to load that feature component.

Add some html to each components html like I did with each of them having a title in an <h2> and bootstrap card layouts for the content. The content does not have to be exactly like mine and other than the correct title it can be whatever you want.