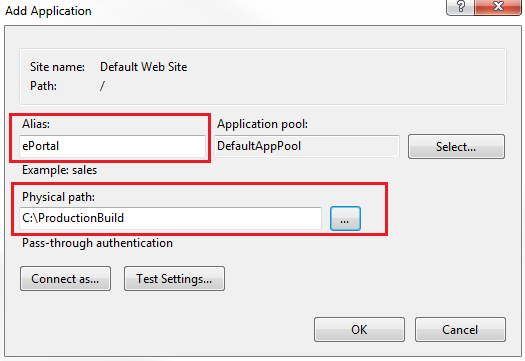
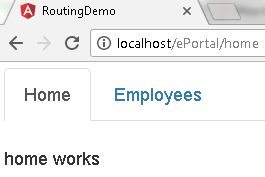
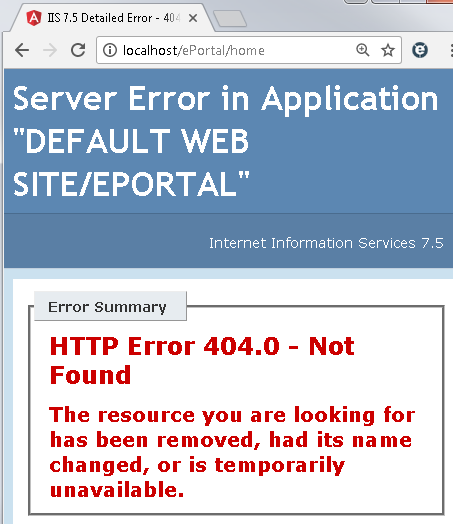
**Step 1 :** Build your angular application.   
  
If you want to deploy a development build do a development build using the following Angular CLI command. The base-href option on the build command sets the base-hrefelement in index.html to "/ePortal/" instaed of "/". In the IIS server, we will create an application with name "ePortal" in just a bit.  
ng build --base-href /ePortal/   
  
  
If you want to deploy a production build do a production build using the following Angular CLI command.   
ng build --prod --base-href /ePortal/  
  
In our case let's deploy a production build. After the build is complete, you will notice a folder with name **"dist"** in your Angular project folder. This folder contains all the build files. These build files need to be copied to a folder on the server where we have IIS installed.  
  
**Step 2 :** Create a folder on the server where you have IIS installed. You can name the folder anything you want. I am going to name the folder **"ProductionBuild"** and I am creating it in **C:\** drive.  
  
**Step 3 :** Now copy all the contents of the **"dist"** folder into **"ProductionBuild"** folder  
  
**Step 4 :** Open IIS. There are several ways to do this. One way is to type **"inetmgr"** in the **"Run"** window and click **"OK"**  
  
**Step 5 :** Create an application in IIS. Name it "ePortal". This name has to match the value we have specified for the --base-href option in Step 1.  

* Exapand the root IIS node
* Expand **Sites**
* Right click on **"Default Web Site"** and select **"Add Application"** from the context menu
* In the **"Alias"** textbox, type **"ePortal"**
* Set the **"Physical Path"** to folder that contains the build files. In our case it is **"ProductionBuild"** folder in **C:\** drive

   
  
At this point, if you launch the browser and navigate to http://localhost/ePortal/home, you will see the **"home works"** message as expected. When you click on the **"Employees"** tab it also works as expected.   
  
   
  
However, when you **"Refresh"** the page by pressing F5, you will see the following HTTP 404 error   
  
   
  
**Step 6 :**  To fix this Page Refresh issue in Angular, include the following URL rewrite rule in you web.config file. This web.config file should be in copied the **"ProductionBuild"** folder where we have the rest of the build files.

<?xml version="1.0" encoding="utf-8"?>

<configuration>

  <system.webServer>

    <rewrite>

      <rules>

        <rule name="AngularJS Routes" stopProcessing="true">

          <match url=".\*" />

          <conditions logicalGrouping="MatchAll">

            <add input="{REQUEST\_FILENAME}" matchType="IsFile" negate="true" />

            <add input="{REQUEST\_FILENAME}" matchType="IsDirectory" negate="true"/>

            <add input="{REQUEST\_URI}" pattern="^/(api)" negate="true" />

          </conditions>

          <action type="Rewrite" url="/ePortal" />

        </rule>

      </rules>

    </rewrite>

  </system.webServer>

</configuration>

**Please note :** You may also point the IIS application directly to the **"dist"** folder in **RoutingDemo**project folder. The downside of this is every time you rebuild your application, the **"dist"** folder is deleted and recreated.  This means you will loose the **web.config** file and you have to create it again. 