First Angular app 1.html

1. Create directory D:\Projects\angular1-intro
2. Open Visual Studio Code
3. Open

File->Open Folder..-> D:\Projects\angular1-intro Select Folder

1. Create a file 1.html
2. Enter the following Angular 1.6.1 code:



1. Open up Windows Explorer and navigate to 1.html
2. Double click 1.html – which should open and run the code in a browse
3. Enter your name – and as you type either ‘forename’ and/or ‘surname’ they should be dynamically re-bound in ‘Hello’ message

More Directives and Filters 2.html

1. Back in Visual Studio Code, create a file 2.html
2. Enter the following code snippet (don’t forget to surround it with the html, & head tags …) – data from <https://en.wikipedia.org/wiki/List_of_countries_by_population_(United_Nations)>

<div ng-init="countries=[{capital:'Washington',name:'United States',population:324},

{capital:'Beijing',name:'China',population:1382},

{capital:'London',name:'United Kingdom',population:65},

{capital:'Paris',name:'France',population:64}]">



Note: add this style (in the <head> section) to style the table

<style>

table, th, td {

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd) {

background-color: #f2f2f2;

}

table tr:nth-child(even) {

background-color: #ffffff;

}

</style>

1. Navigate and run 2.html
2. Modify ‘countries’ and add a field ‘population2015’ for each of the countries to each hold population from 2015.
3. Modify <table>, add table header <th> and table data <td> to show population difference
4. Check your results

Modules & Controllers 3.html

1. Back in Visual Studio Code, create a file 3.html
2. Enter the following code snippet:



1. Test your app by running in browser and ensure you can see ‘Hello World’ message.

Controller Initialising Data 4.html

1. Back in Visual Studio Code, create a file 4.html
2. Enter the following code snippet (use the <style> from 3.html):



1. Test your app by running in browser.

SPA Routing With Express 5.html

Angular routing requires a server/Express to be running

1. In Visual Studio Code select View->Integrated Terminal
2. Run following to create a package for express

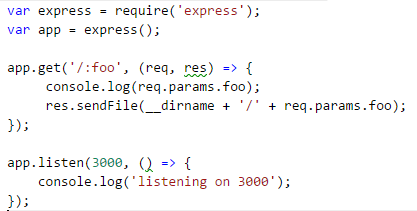
npm init

Accept all defaults. This creates a package.json file.

1. In Integrated Terminal run

npm install express --save

1. Create a file called app.js
2. Enter following:



1. Create a file 5.html
2. Copy contents of 4.html into this file 5.html (it will save on typing)
3. Modify 5.html by adding
   1. New Page loading & href
   2. New ng-vew
   3. Optionally modified <script .. angular.min.js -- changed to use min version
   4. New <script .. angular-route.js
   5. Modified .module
   6. New .config



1. Create a file external-page 5.html
2. Add a simple message such as: (note do not add html/head sections)



1. To test, in the Integrated Terminal start the node application app.js by typing:

node app

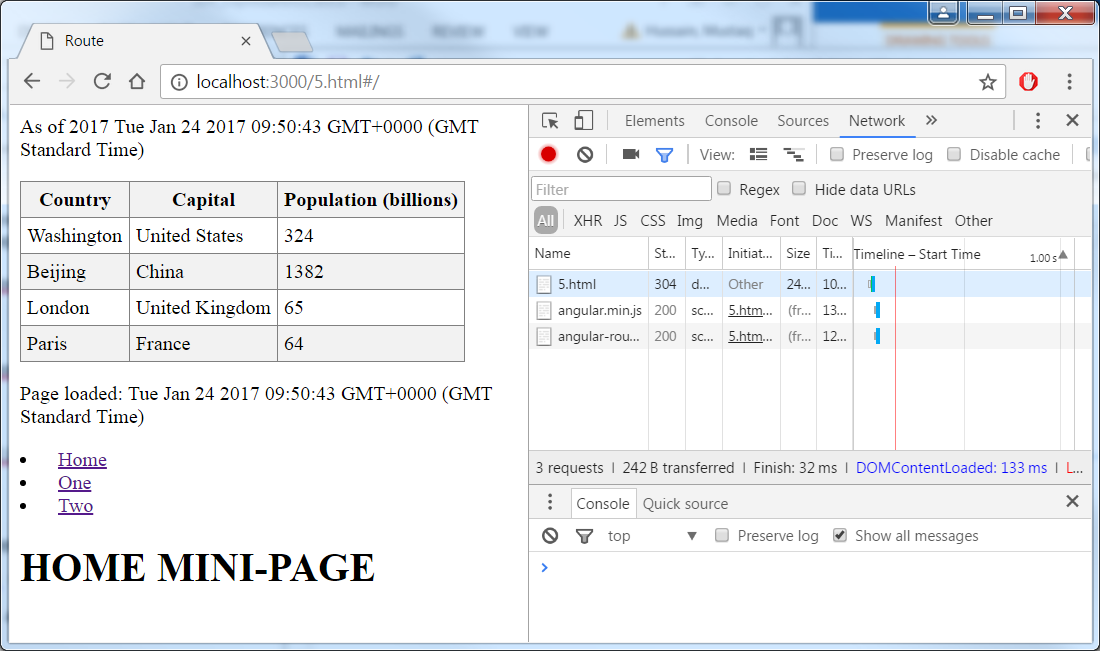
1. Open a browser (examples for Chrome) and press F12 to bring up developer tools.
2. Press Network tab and press Clear to remove traffic history.
3. Navigate to localhost:3000/5.html, and notice the traffic being sent across the wire.

5.html

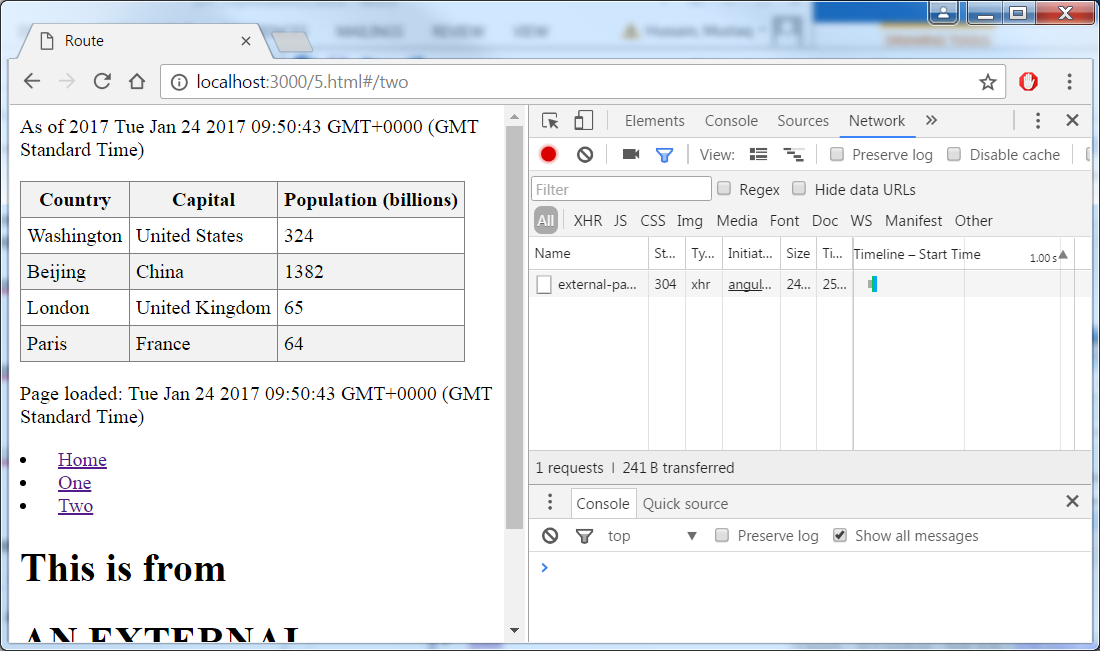
angular.min.js

angular-router.js

and the time e.g. 09:50:43



1. Press Clear on Network tab to clear history
2. Press link One – result is no Network traffic
3. Press link Two – result is Network traffic to load external-file.html, however enclosing page not refreshed (time still 09:50:43)



1. Press Clear on Network tab to clear history.
2. In browser enter localhost:3000/5.html this should force re-fresh (new time) and traffic being sent across the wire again.