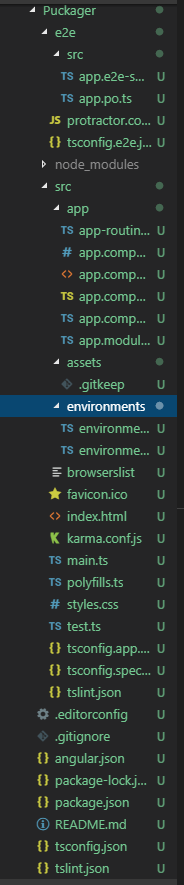
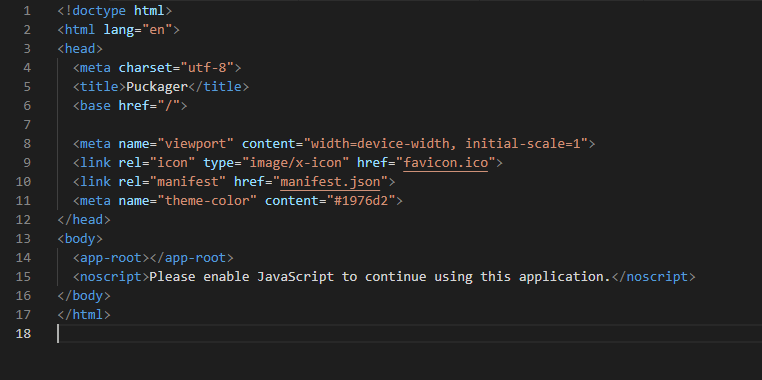
Puckager in Angular

1. Created new branch and deleted all files
2. Ran command “ng new Puckager”
   1. This creates the whole angular project;
   2. The only part we’re concerned about right now is the “src” folder. This contains all our app files;

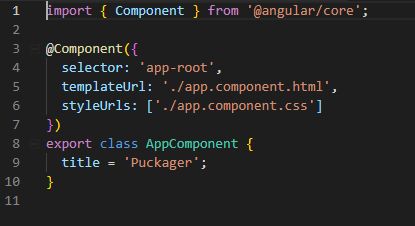


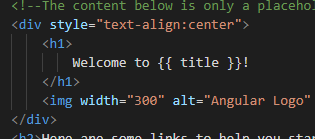
1. Cleaned up folders and ran command “ng add @angular/pwa”
   1. This adds a few files, and a couple lines of code – usually denoted by “service worker” or “ngsw” – don’t worry about this for now, but it has to do with making our app into a PWA in the future.
2. Src
   1. Index.html – this is our main html page; from here, the app loads our app component (denoted by <app-root></app-root>); this is like our global html file, so anything we want to be in the whole application we can put here



* 1. Main.ts – this is our main typescript page; same as with the index, anything we want to code globally we can do here;
  2. Styles.css – same, anything we want to style globally we can put here

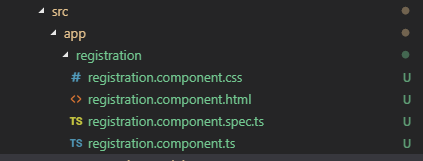
1. Src>app
   1. App.component.html – this will be our main body component; this will be the starting point for our app at the moment (we can change this by changing the <app-root> portion in the index.html to the new module name); this is our main body html, so anything we want to be global we can add here;
   2. App.component.ts – this is our main code typescript for the component. In here, we have an “export” class, this will export anything we put here to be usable in the html. As you see, we have exported a variable “title” which has the value “Puckager”. Below this is the app html, which makes use of the variable;



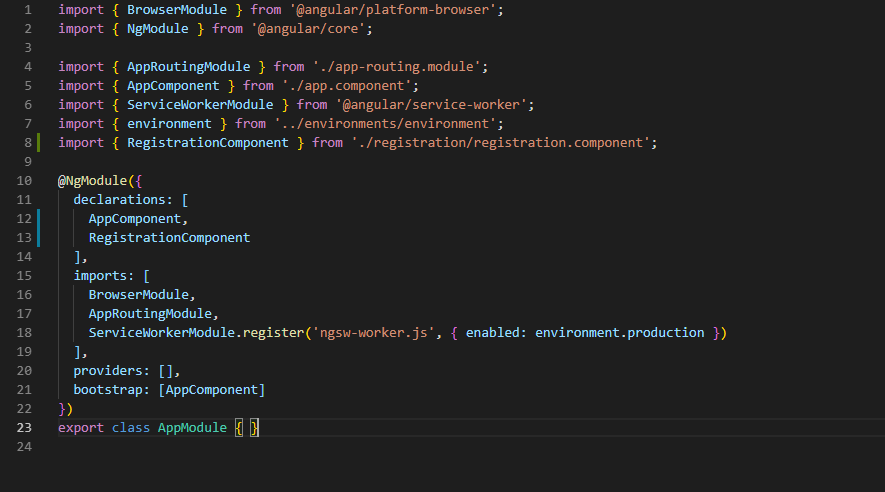


* 1. App.component.css – this is our main css for the component;
  2. App.module.ts – this is our main module component; all modules will be declared here (I’ll touch on this in the next point);
  3. App-routing.module.ts – this module holds the definitions of all of our routes (will touch on this again in 6);
  4. Important to remember about the above – because of the structure of the angular application, any css you apply to the App Module will only be applied to it and its children, but will not be applied to the parts outside of the body (which would be done in the styles.css). But, if you change css defined in the styles.css, it will be overwrite because it is the newest loaded css.

1. Adding a Module – run command “ng generate component {componentName}”
   1. Running this command will generate a new component folder with 4 files under “src>app” – a component html, css, ts and spec file (don’t worry about the spec file);



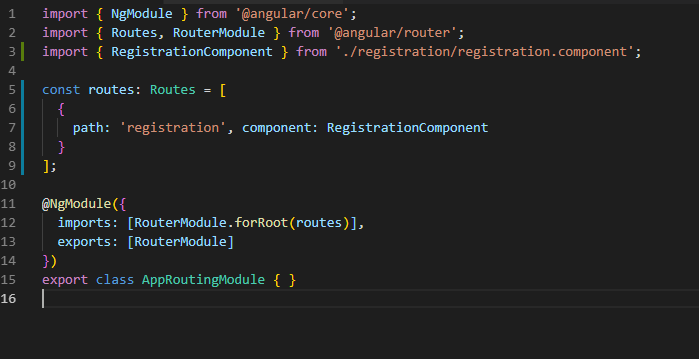
* 1. For this I generated a “registration component” – when generated, it automatically adds some things to the app.module.ts. First, you see it added an import statement for the Registration Component by giving it a variable defined in the component (it’s almost always the name followed by “Component”) and the path to the file. Second, it then added this variable under the “Declarations” array:



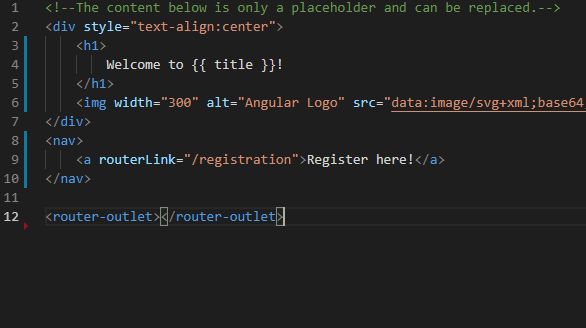
* 1. Registration.module.ts – lastly, looking at this shows some possible things to do when starting a component. One such thing is “ngOnInit” which will do anything in the function on initialization of the component (when the route is accessed):



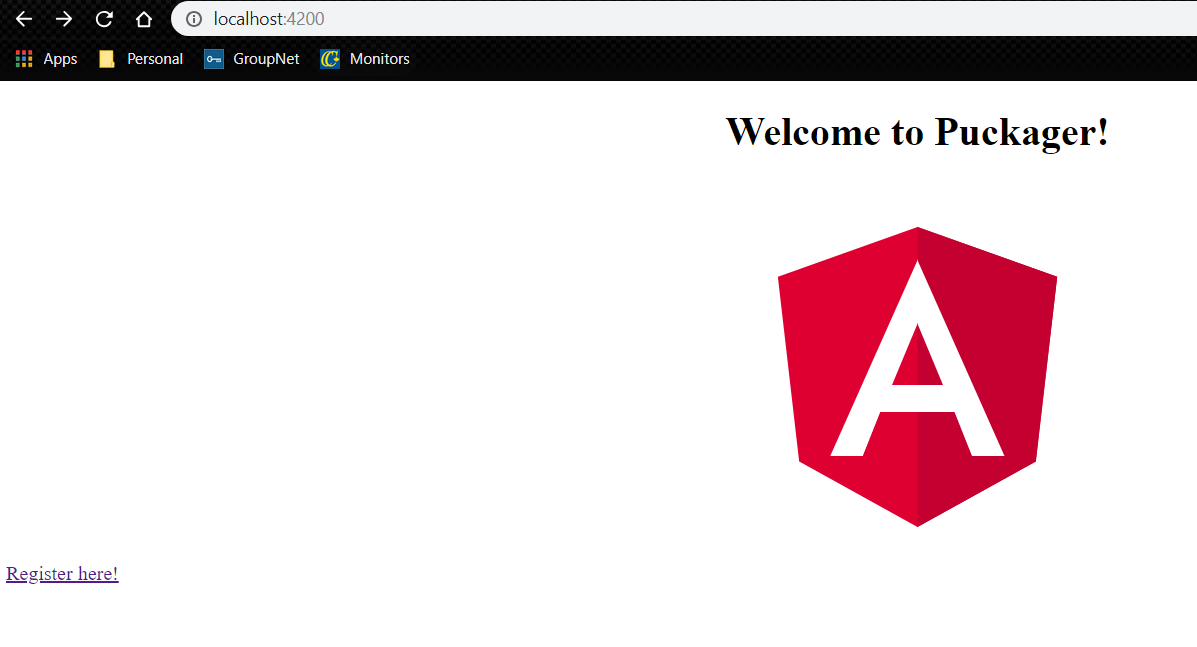
1. Routing – routing to change body based on different url endings, basically when you add a “/{whatever}” to the end of the base url, the body will changed based on what you have defined
   1. To do this, I’ve added to two files after generating the new component
      1. App-routing.module.ts – here, we add two things – we import the registration component again, and we add the path to the routes object. This gives us the endpoint to our url and tells the path where to go:

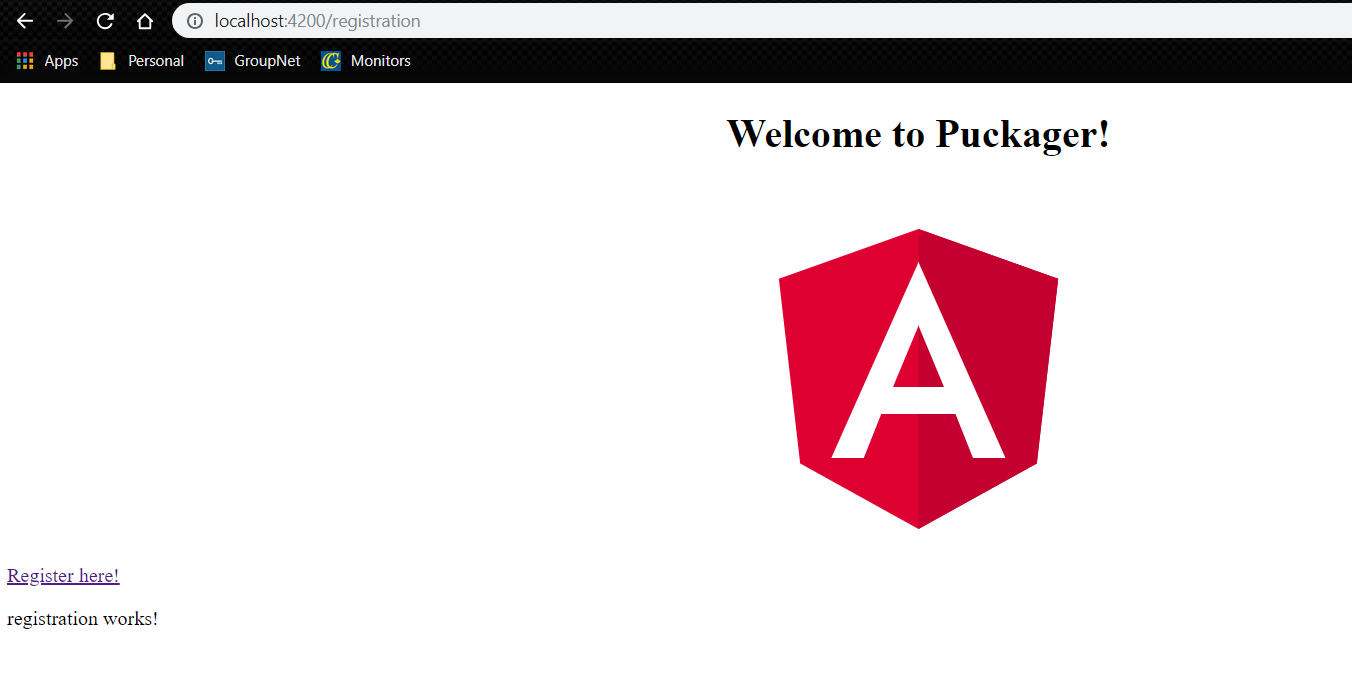


* + 1. App.component.html – here, we add a simple nav link to the registration page, and we add <router-outlet> tag to the html to let the app know that it will be routing:



Checkout the URL and the difference in the pages below





Appendix

* Starting a new app - <https://angular.io/guide/quickstart>
* Adding service worker - <https://angular.io/guide/service-worker-getting-started>
* Modules info - <https://angular.io/guide/architecture-modules>
* Routing info - <https://angular.io/tutorial/toh-pt5>