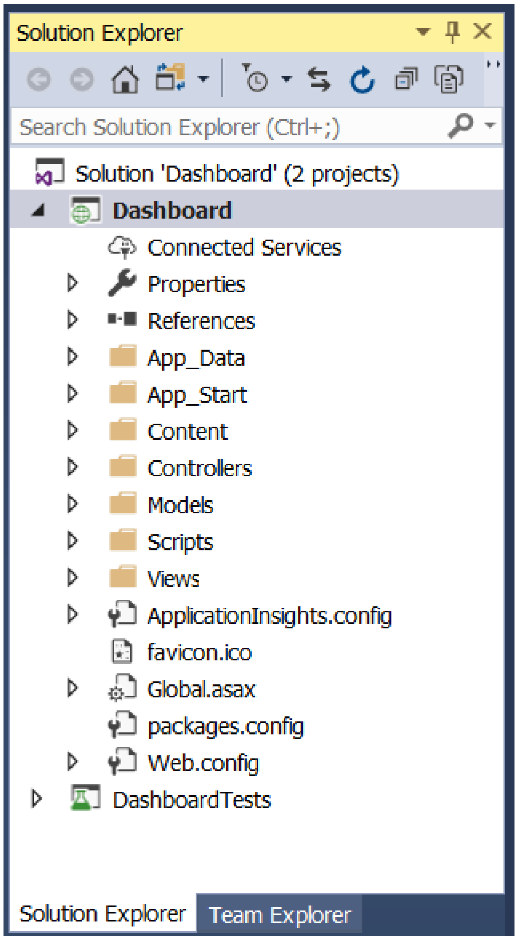
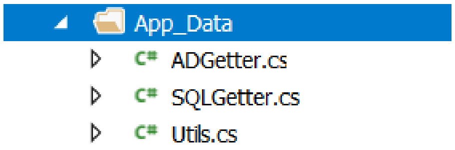
Dashboard Website Documentation



- Solution 'Dashboard' (2 projects)
  - Dashboard
    - Connected Services
    - Properties
    - References
    - App_Data
    - App_Start
    - Content
    - Controllers
    - Models
    - Scripts
    - Views
    - ApplicationInsights.config
    - favicon.ico
    - Global.asax
    - packages.config
    - Web.config
  - DashboardTests

Open the project in visual studio – this is the layout in the Solution Explorer. I will give a basic run down of features; **there are lots of comments in code** to help also.



- App_Data
  - ADGetter.cs
  - SQLGetter.cs
  - Utils.cs

App_Data folder contains ADGetter.cs, SQLGetter.cs, and Utils.cs

ADGetter performs the work of querying data from Active Directory, SQLGetter queries from SQL Databases, and Utils contains functions both these classes use

ADGetter:

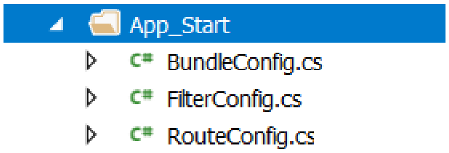
- parseData(string query): function which takes a query and executes either the GetComputer or GetUser function based on the query
- GetComputer(string name): searches for ComputerPrincipal object in Active Directory then returns data in table to be rendered on search page
- GetUser(string name): searches for UserPrincipal object in Active Directory then returns data in table to be rendered on search page

SQLGetter:

- GetSCSM(), GetEPO(), GetSCCM(): these functions populate DataTables with data from SQL
- getLastUpdate(string username): searches SCCM network logins to retrieve latest login time
- Search(string query): searches the 3 databases for query and sets up table for rendering on search page
- GetDashData(): performs all required queries for dashboard page and sends all information to be sent to webpage for display

Utils:

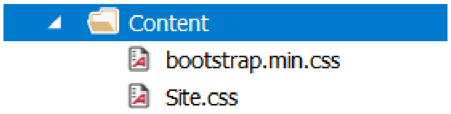
- ConvertDataTableToHTML(DataTable dt, string id): this function takes a C# DataTable and converts into string which marks up valid HTML table
- ConvertUTCtoEST(DateTime utc): takes a DateTime object which is in UTC format and returns a DateTime converted to EST
- GetNumbers(string input): takes an alphanumeric string and returns a new string with only the numeric characters



App_Start
BundleConfig.cs
FilterConfig.cs
RouteConfig.cs

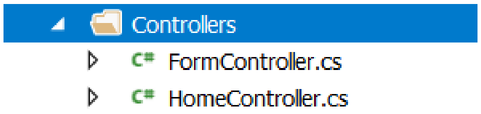
App_Start folder contains BundleConfig.cs, FilterConfig.cs, and RouteConfig.cs

These classes are initiated when the application first starts. BundleConfig is a technique used to reduce load time my decreasing number of requests to server by bundling CSS and JavaScript files together. RouteConfig is used to register routes (urls). FilterConfig is used to register global filters to all actions and controllers.



Content
bootstrap.min.css
Site.css

The Content folder contains the CSS the application uses.



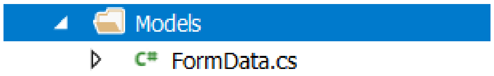
Controllers
FormController.cs
HomeController.cs

Controllers are responsible for processing incoming HTTP requests, handling user input, retrieving and saving data, and determining the response to send back to the client.

Form Controller – handles post requests

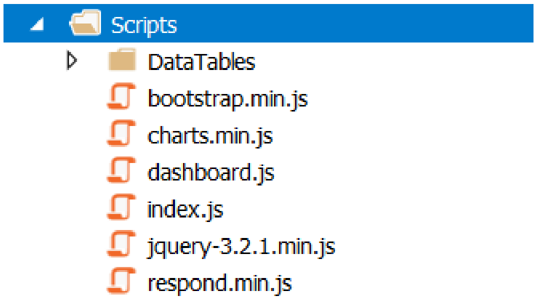
- FormOne: used to initiate a search
- FormTwo: sends all relevant dashboard data

Home Controller – handles all http requests and returns appropriate page



Models
FormData.cs

The Models folder contains a FormData.cs which is a simple model for the post data consisting of a string.



Scripts
DataTables
bootstrap.min.js
charts.min.js
dashboard.js
index.js
jquery-3.2.1.min.js
respond.min.js

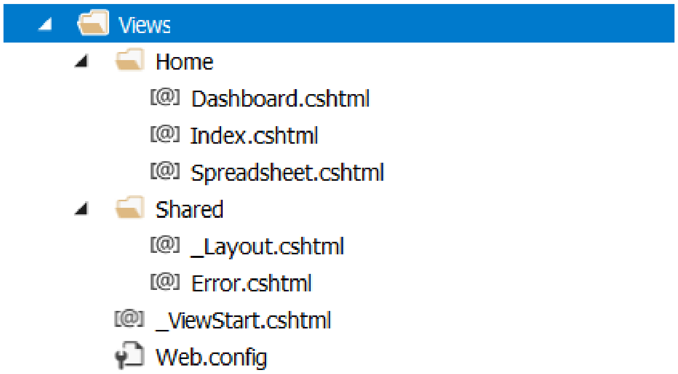
The Scripts folder contains all the JavaScript files required for the web application. Except for dashboard.js and index.js they are all libraries used: jQuery, Charts.js, DataTables.js, Respond.js, and Bootstrap.js

Index.js

- Submit(): post search query to server and initiate search functions

Dashboard.js

- Contains functions to setup pie charts



```
Views
  Home
    Dashboard.cshtml
    Index.cshtml
    Spreadsheet.cshtml
  Shared
    _Layout.cshtml
    Error.cshtml
  _ViewStart.cshtml
  Web.config
```

The view folder contains the page data to be rendered. The shared folder contains the base layout for each page using Layout.cshtml.

Layout:

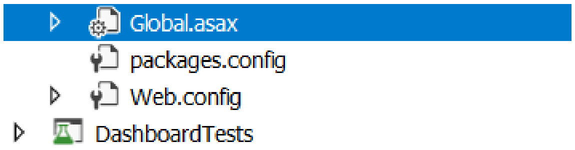
- The <head> tag contains linked JavaScript and CSS files to be used by all pages of application
- Contains the setup for the navigation bar
- To create a new page: Add function to HomeController, create cshtml page in Views, and add to navigation bar (see Step 1, Step 2 in comments)

Index:

- This is the main search page which opens at application startup
- Contains input and submit button to initiate searches
- <div> id=tableDiv is used to hold tables created when searching

Dashboard:

- Contains links to SCCM, EPO pages under <head>
- Shows loader while waiting for SQL data
- <div> with id=hiddenDiv is used to display information when clicking pie slices
- <div< with id=tail is the percent tool tip which is shown when hovering over pie charts
- Pie charts are shown in 2 columns indicated by the two divs with style=float:right or style=float:left
- Each pie chart has a label and canvas with a line break in-between each for spacing



```
  Global.asax
  packages.config
  Web.config
DashboardTests
```

In the main directory of the project these files will reside. The Global.asax contains functions for Application_Start and Session_Start. Application_Start runs every function within once at the startup of the application. Session_Start is initiated every time a new session connects to the webpage (includes reload). So essentially to refresh data from SQL you can refresh page manually because Session_Start calls the initiate database functions. Web.config contains the setup for authorization should It ever need to be changed.