Best of Both Worlds

About Austin McDaniel

Software Engineer focused on large-scale web applications. JavaScript enthusiast.

Follow me on Twitter @amcdnl

Follow me on Github github.com/amcdnl

Do you Angular?

AngularJS is a up and coming, its a great choice because:

- Huge community (Stackoverflow / Plugins / Stability)
- Support by Google
- On the cutting edge (v3 will do ES6)

Why .NET?

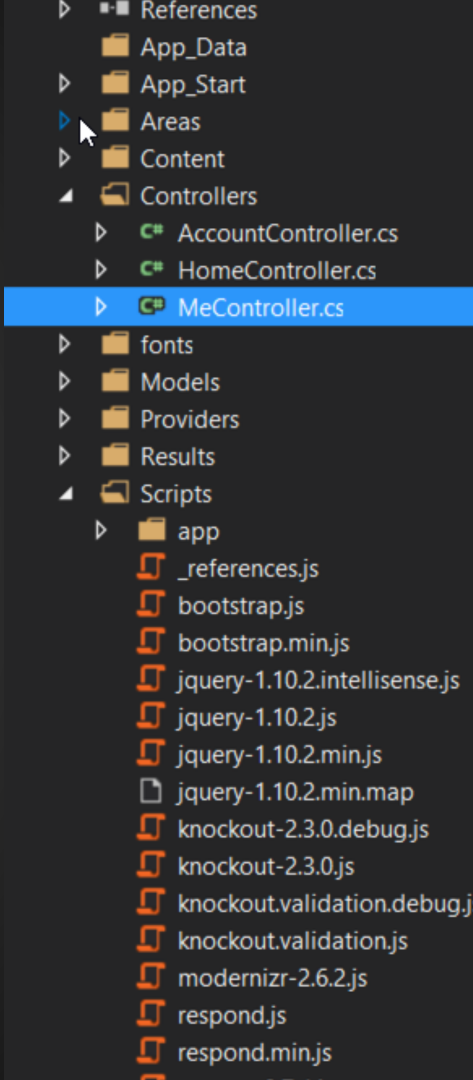
I <3 NodeJS as much as the next guy, but .NET has the stability and ‘trust’ enterprise customers want.

Web API 2 has won me over with its great built in security, ease of use, and .net reliability.

Why .NET does SPA Wrong!

Create a new SPA template in VS2013 and you get this ->

- MVC + SPA?!
- One folder for all JS?
- Defaults to Knockout?



```
References
App_Data
App_Start
Areas
Content
Controllers
    AccountController.cs
    HomeController.cs
    MeController.cs
fonts
Models
Providers
Results
Scripts
    app
    _references.js
    bootstrap.js
    bootstrap.min.js
    jquery-1.10.2.intellisense.js
    jquery-1.10.2.js
    jquery-1.10.2.min.js
    jquery-1.10.2.min.map
    knockout-2.3.0.debug.js
    knockout-2.3.0.js
    knockout.validation.debug.j
    knockout.validation.js
    modernizr-2.6.2.js
    respond.js
    respond.min.js
```

Github

Follow along with me in code on via Github:

https://github.com/amcdnl/angularpreso

Structuring and Plumbing

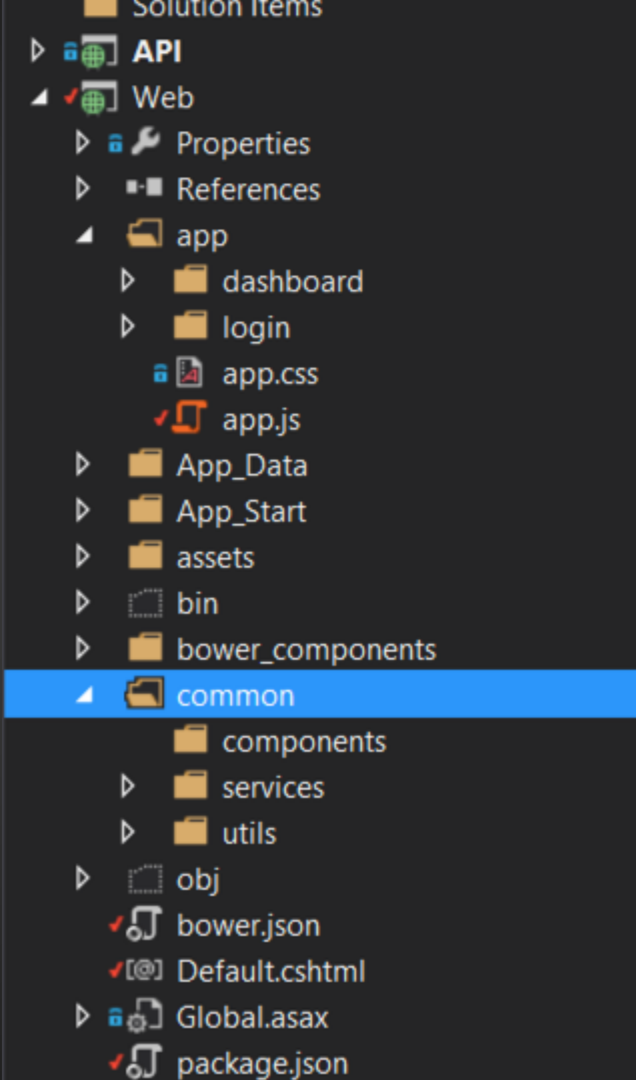
Structuring your app

The Web API and Web should be 2 separate projects.

- Multiple Consumers (Mobile / Desktop / Web)
- Better organization
- Forces RESTful 3-tier architecture

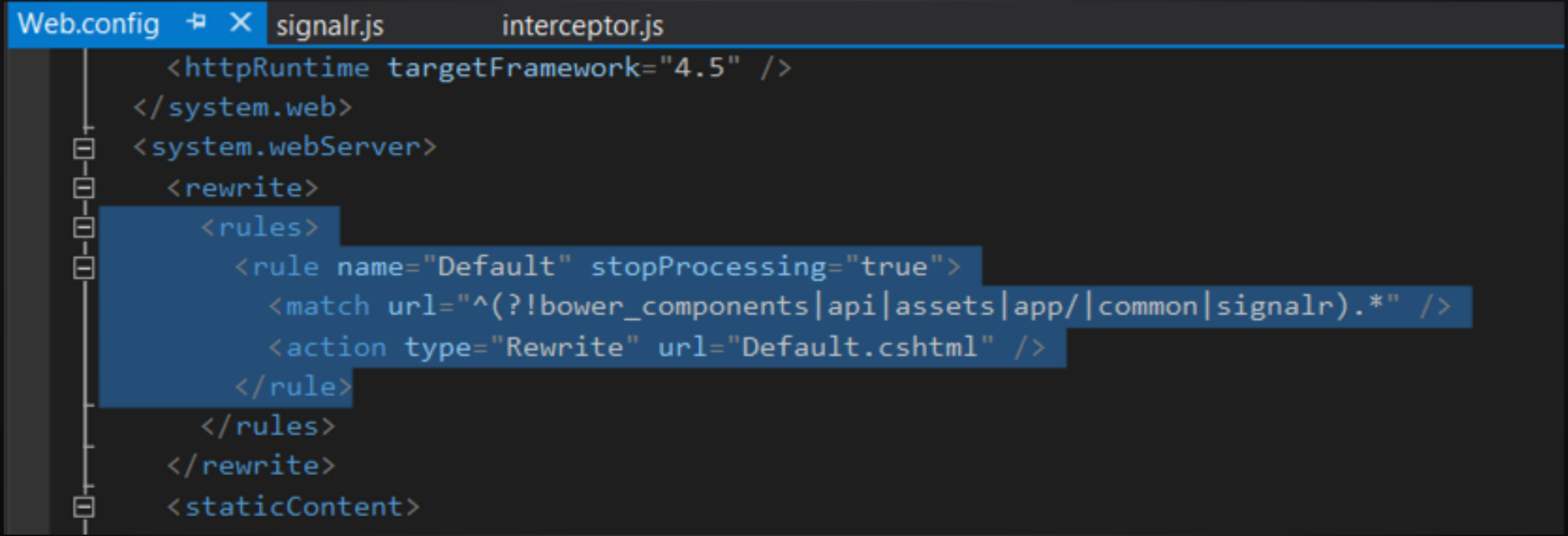
The Web Project

- Blank MVC Project
  - Allows for OWIN/SignalR startup
  - Easy debugging / prod deploy
  - Use bundler, etc



URL Rewriting 2.0

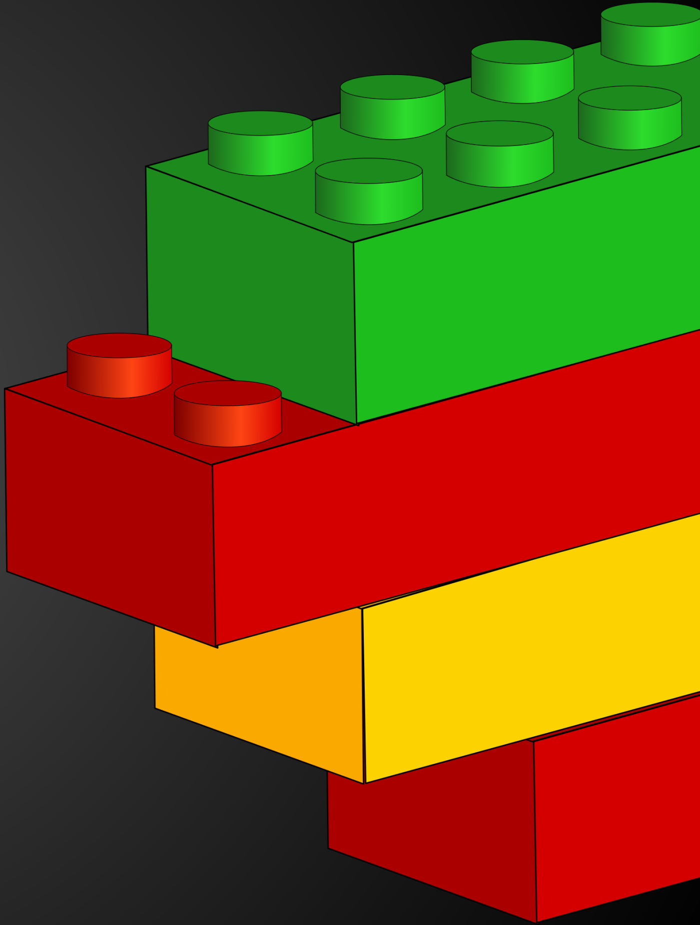
Stop MVC processing and redirect to Default.cshtml



```xml
    <httpRuntime targetFramework="4.5" />
  </system.web>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="Default" stopProcessing="true">
          <match url="^(?!bower_components|api|assets|app/|common|signalr).*" />
          <action type="Rewrite" url="Default.cshtml" />
        </rule>
      </rules>
    </rewrite>
    <staticContent>
```

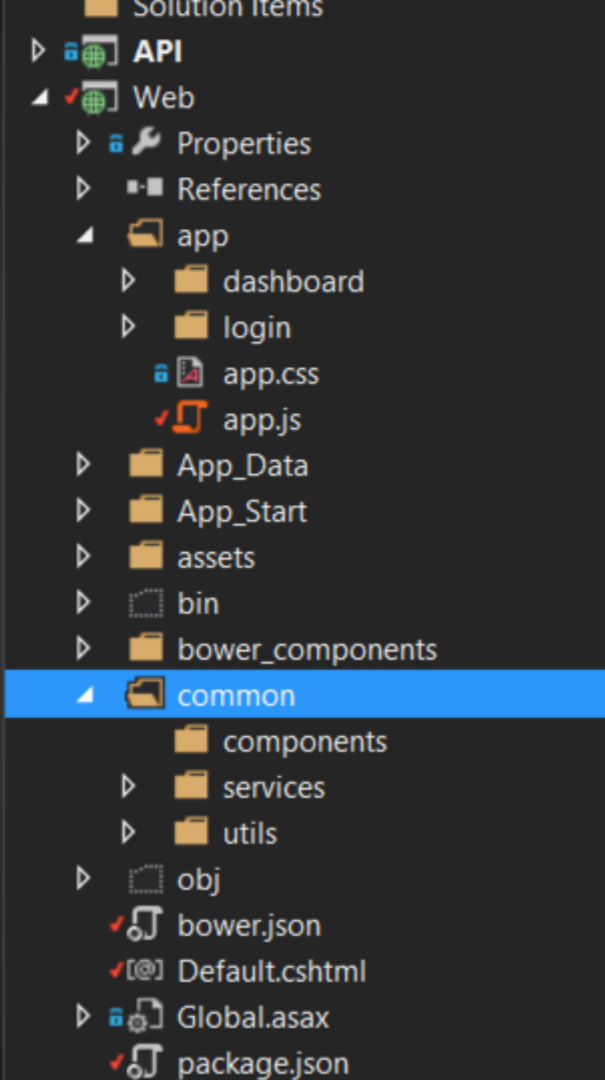
So how do you build a BIG JavaScript application?

The secret: You don’t! You build lots of small components that are glued together by one main ‘controller’.

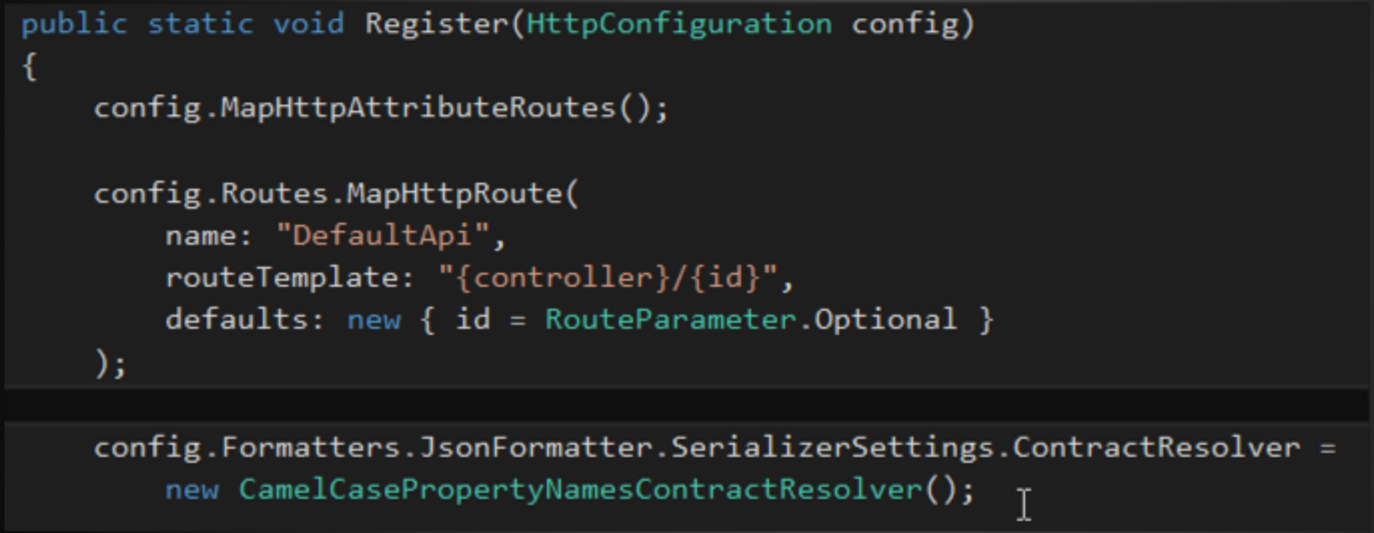


Structuring your JS

- Split your ‘areas’ up into separate folders.
    - Cousins shouldn’t be coupled.
    - Child Components aren’t coupled to parents (use EOA)
    - Common services / utilts / components grouped in own container



API Configuration

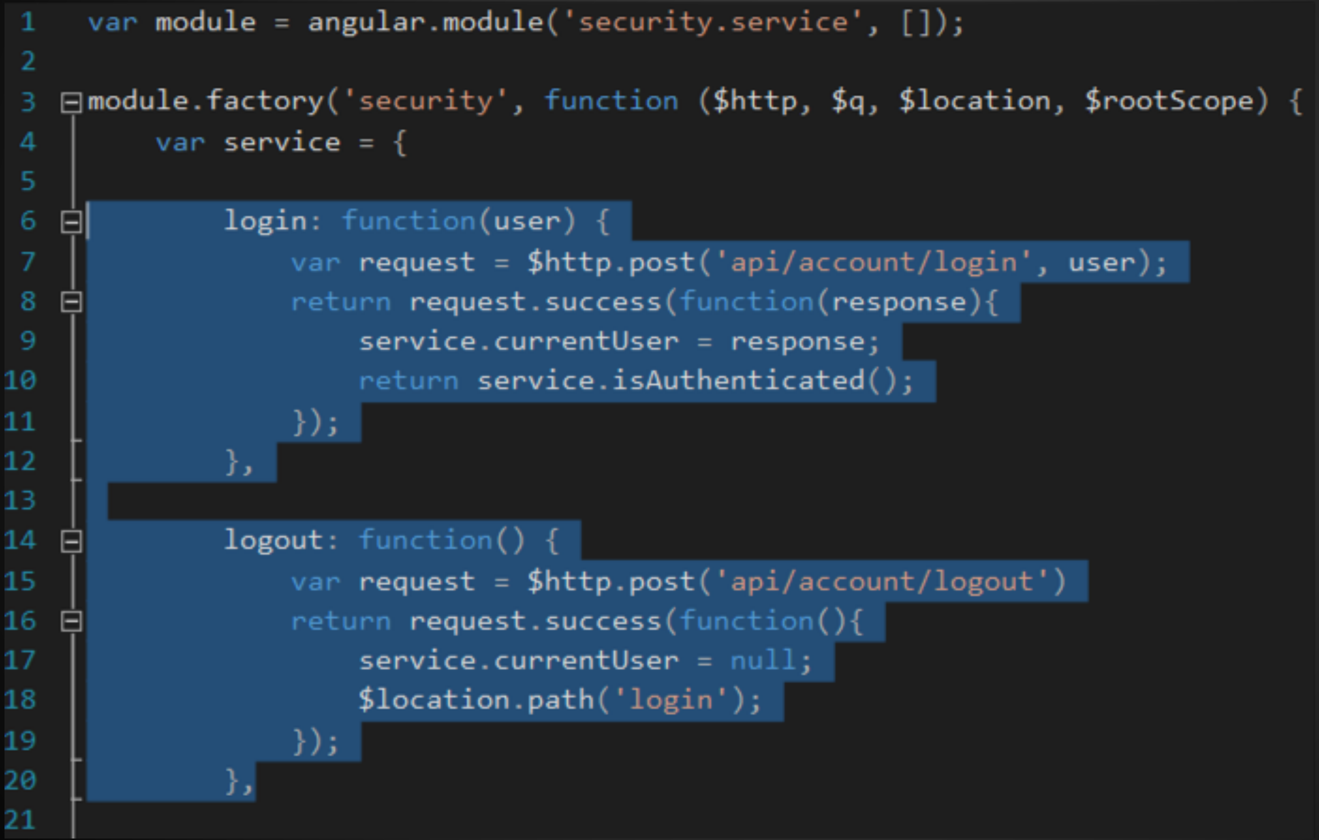


```
public static void Register(HttpConfiguration config)
{
    config.MapHttpAttributeRoutes();

    config.Routes.MapHttpRoute(
        name: "DefaultApi",
        routeTemplate: "{controller}/{id}",
        defaults: new { id = RouteParameter.Optional }
    );

    config.Formatters.JsonFormatter.SerializerSettings.ContractResolver =
        new CamelCasePropertyNamesContractResolver();
```

Authentication

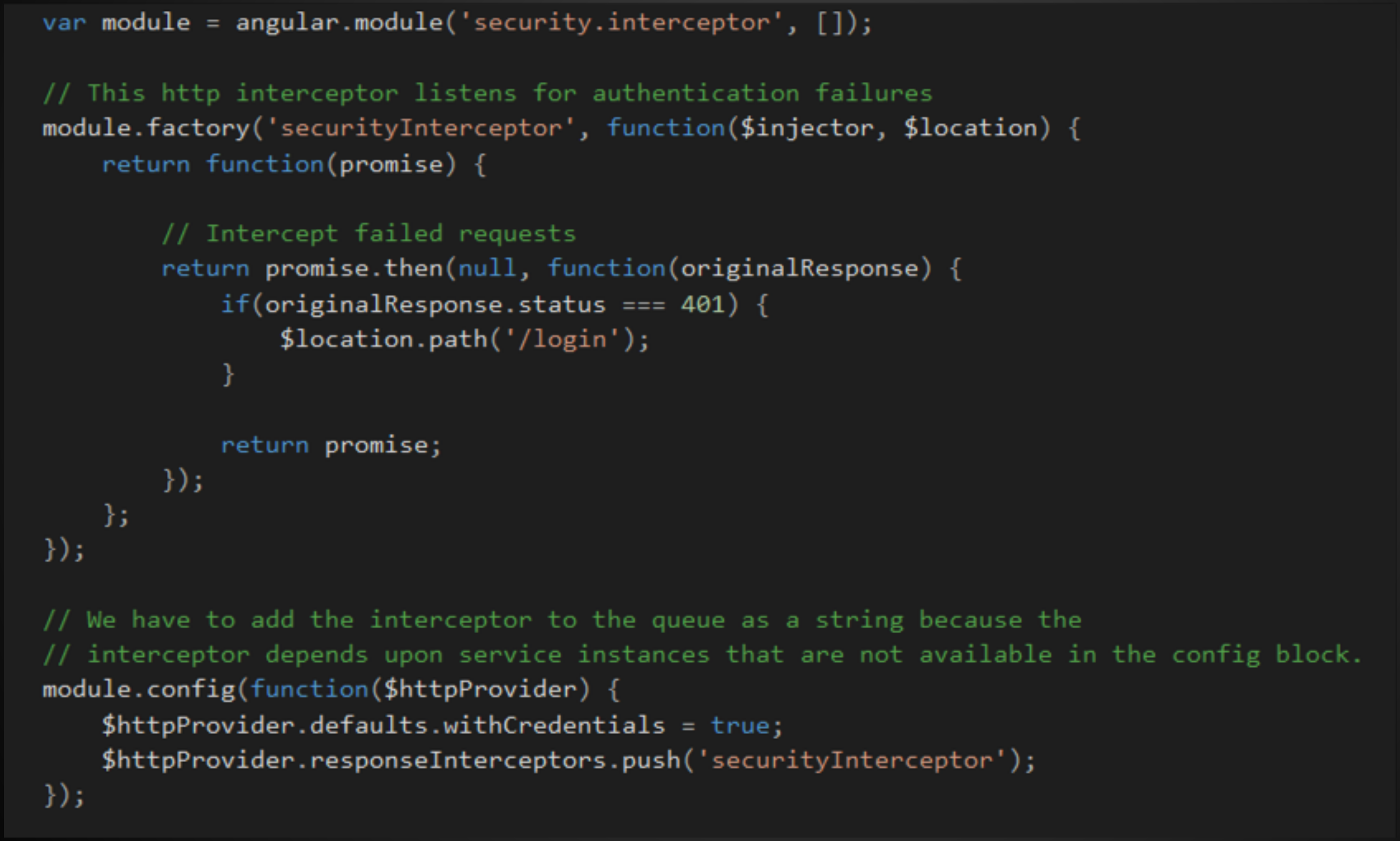
AngularJS + Web API + Owin

Angular Service



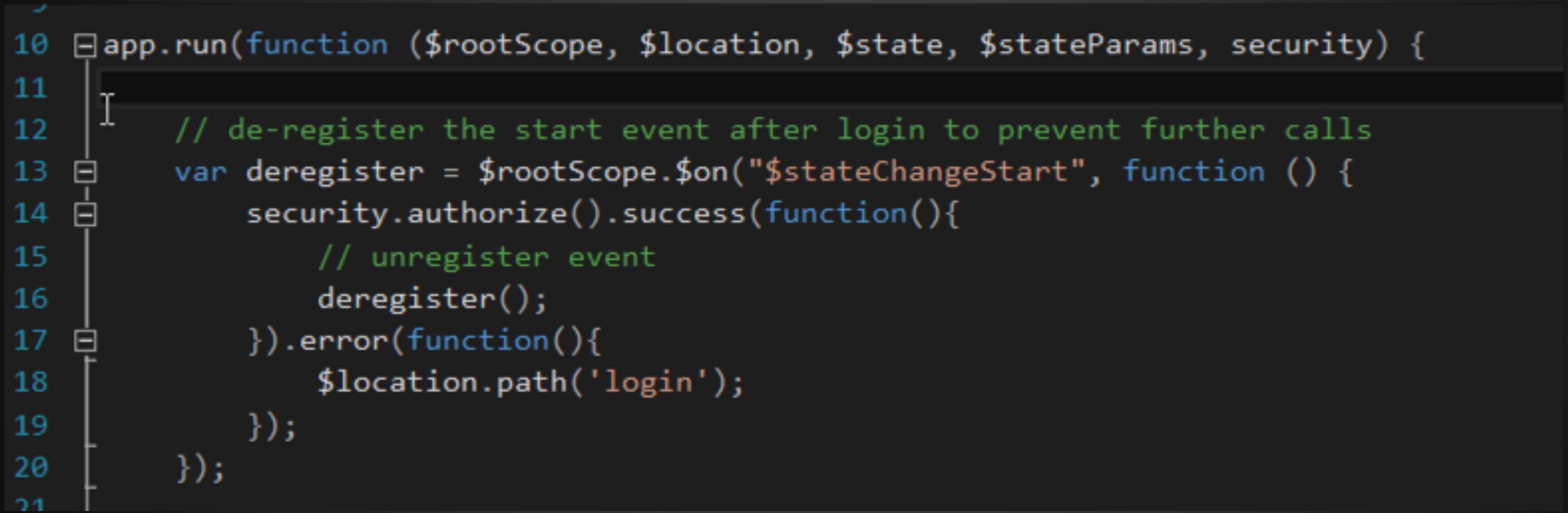
```javascript
var module = angular.module('security.service', []);

module.factory('security', function ($http, $q, $location, $rootScope) {
    var service = {

        login: function(user) {
            var request = $http.post('api/account/login', user);
            return request.success(function(response){
                service.currentUser = response;
                return service.isAuthenticated();
            });
        },

        logout: function() {
            var request = $http.post('api/account/logout')
            return request.success(function(){
                service.currentUser = null;
                $location.path('login');
            });
        },

```

Angular Interceptor



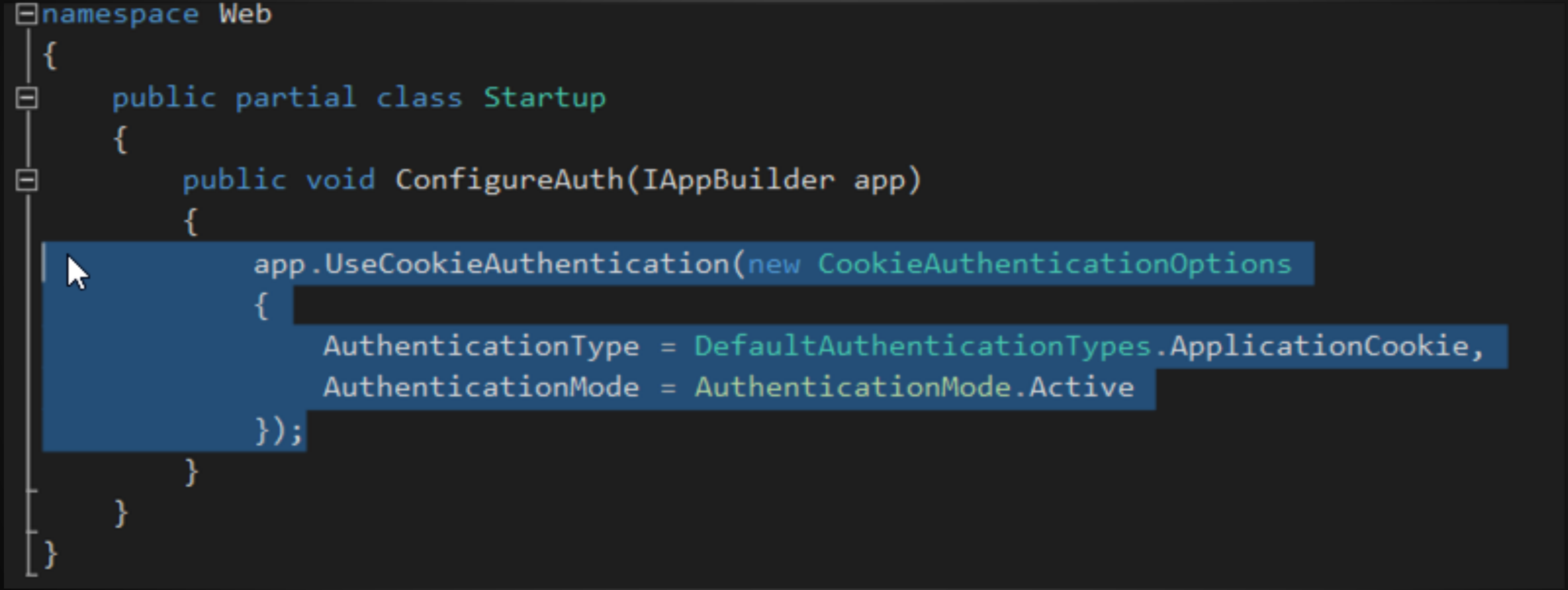
```
var module = angular.module('security.interceptor', []);

// This http interceptor listens for authentication failures
module.factory('securityInterceptor', function($injector, $location) {
    return function(promise) {

        // Intercept failed requests
        return promise.then(null, function(originalResponse) {
            if(originalResponse.status === 401) {
                $location.path('/login');
            }

            return promise;
        });
    };
});

// We have to add the interceptor to the queue as a string because the
// interceptor depends upon service instances that are not available in the config block.
module.config(function($httpProvider) {
    $httpProvider.defaults.withCredentials = true;
    $httpProvider.responseInterceptors.push('securityInterceptor');
});
```

Angular App Start



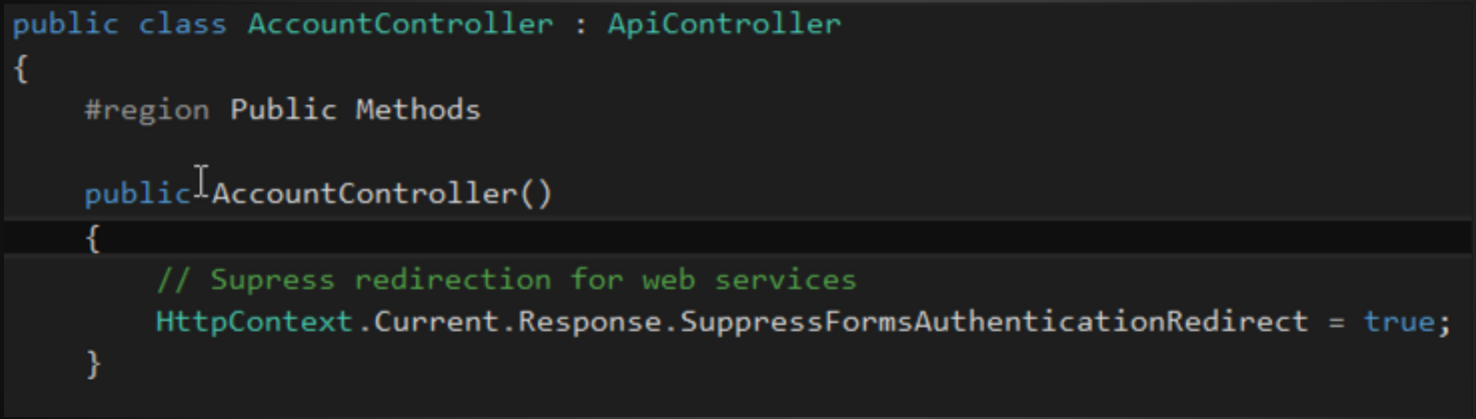
```
app.run(function ($rootScope, $location, $state, $stateParams, security) {

    // de-register the start event after login to prevent further calls
    var deregister = $rootScope.$on("$stateChangeStart", function () {
        security.authorize().success(function(){
            // unregister event
            deregister();
        }).error(function(){
            $location.path('login');
        });
    });
```

Web Startup Auth



```
namespace Web
{
    public partial class Startup
    {
        public void ConfigureAuth(IAppBuilder app)
        {
            app.UseCookieAuthentication(new CookieAuthenticationOptions
            {
                AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
                AuthenticationMode = AuthenticationMode.Active
            });
        }
    }
}
```

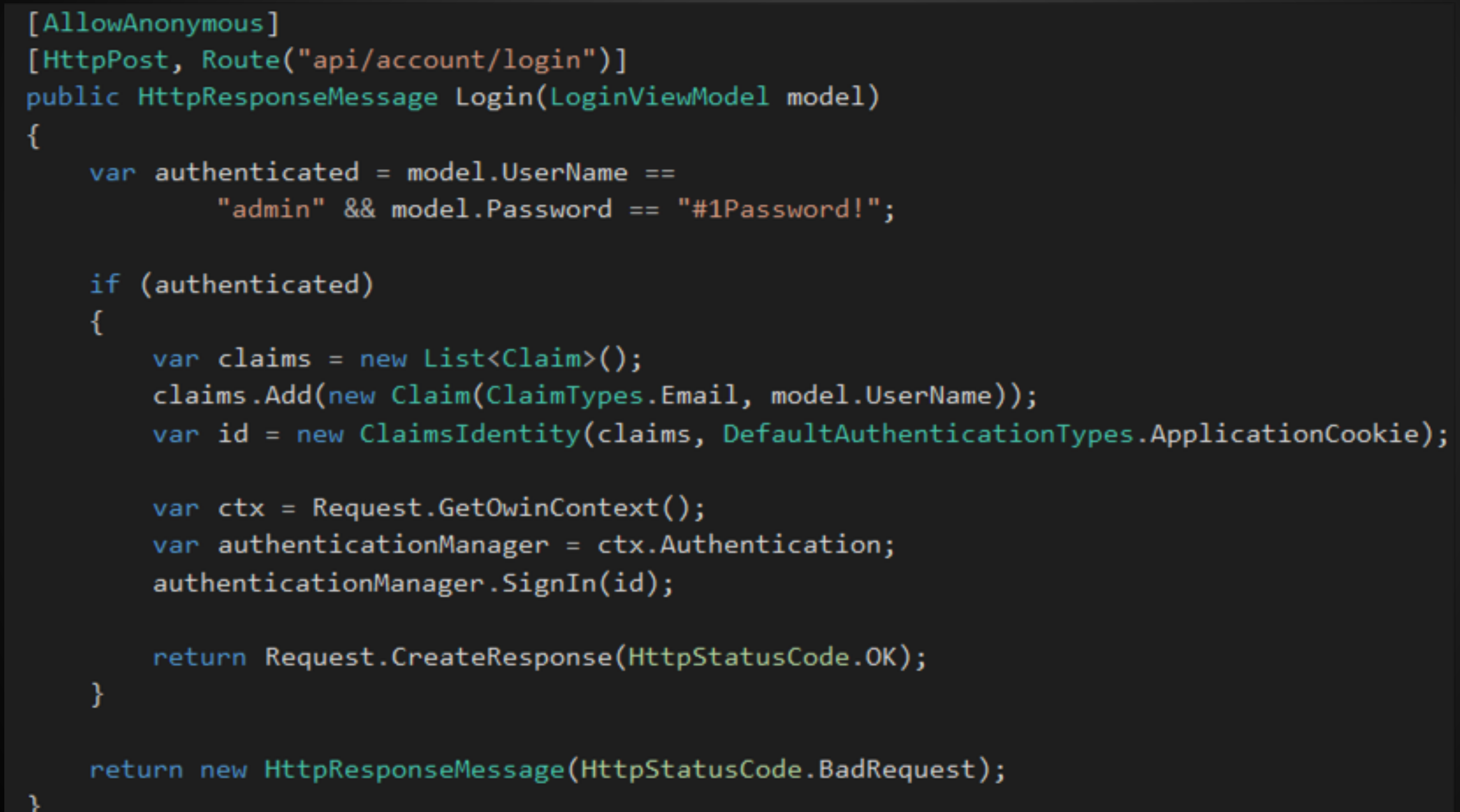
API AccountController



```
public class AccountController : ApiController
{
    #region Public Methods

    public AccountController()
    {
        // Supress redirection for web services
        HttpContext.Current.Response.SuppressFormsAuthenticationRedirect = true;
    }
```

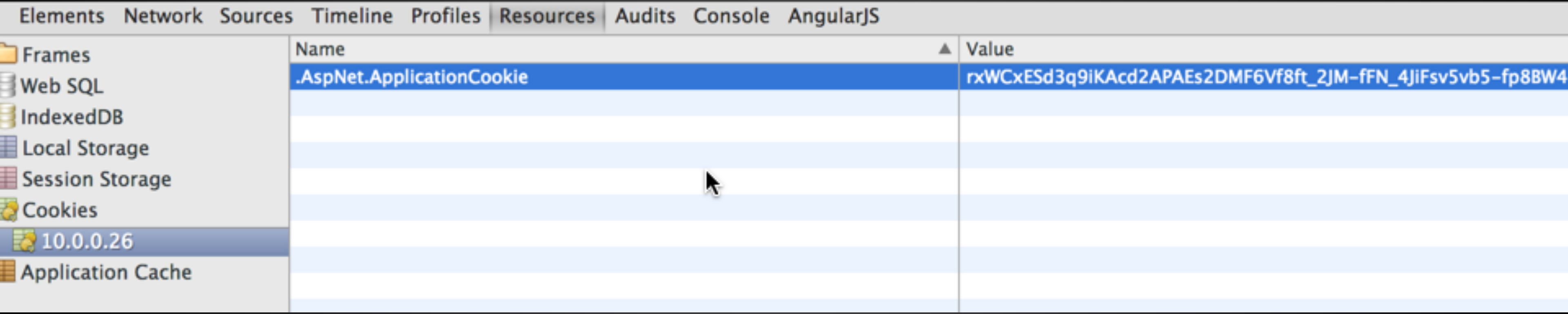
Stop from sending back HTML pages

API AccountController Login



```
[AllowAnonymous]
[HttpPost, Route("api/account/login")]
public HttpResponseMessage Login(LoginViewModel model)
{
    var authenticated = model.UserName ==
            "admin" && model.Password == "#1Password!";

    if (authenticated)
    {
        var claims = new List<Claim>();
        claims.Add(new Claim(ClaimTypes.Email, model.UserName));
        var id = new ClaimsIdentity(claims, DefaultAuthenticationTypes.ApplicationCookie);

        var ctx = Request.GetOwinContext();
        var authenticationManager = ctx.Authentication;
        authenticationManager.SignIn(id);

        return Request.CreateResponse(HttpStatusCode.OK);
    }

    return new HttpResponseMessage(HttpStatusCode.BadRequest);
}
```

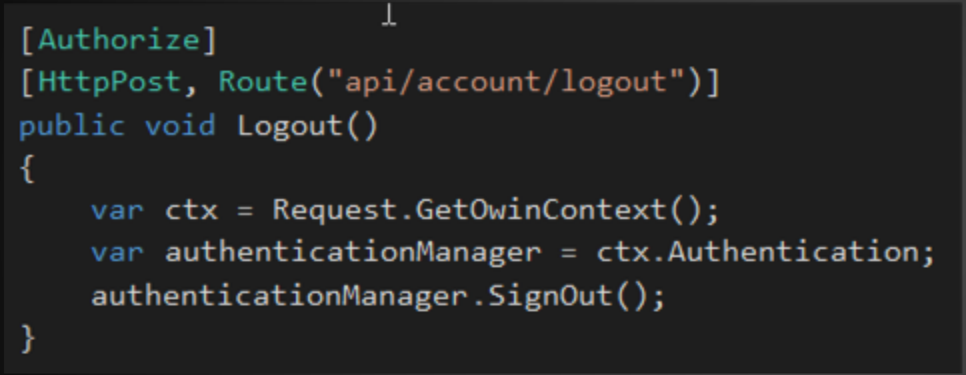
Token from Login



Name	Value
.AspNet.ApplicationCookie	rxWCxESd3q9iKAcd2APAEs2DMF6Vf8ft_2JM-fFN_4JiFsv5vb5-fp8BW4

.NET automatically generates a AspNetCookie for you

API AccountController Logout

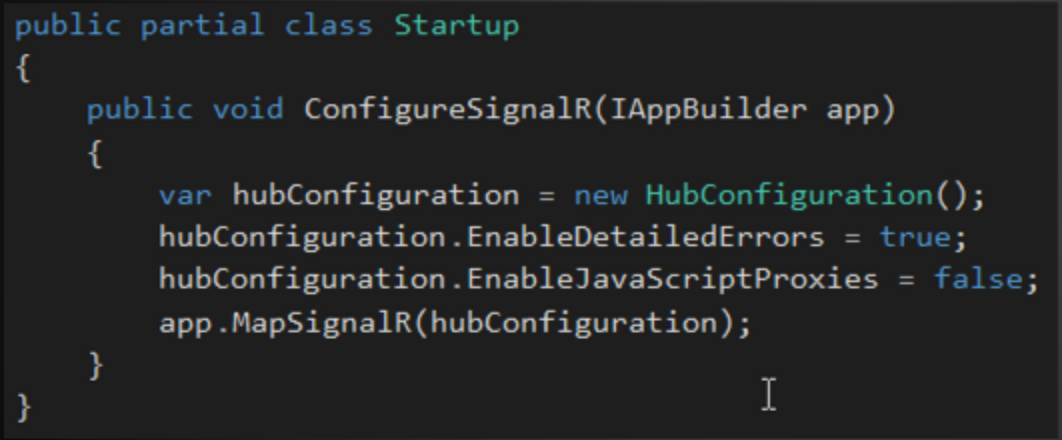


```
[Authorize]
[HttpPost, Route("api/account/logout")]
public void Logout()
{
    var ctx = Request.GetOwinContext();
    var authenticationManager = ctx.Authentication;
    authenticationManager.SignOut();
}
```

Authorize Tags on Requests, automatically handles token authentication.

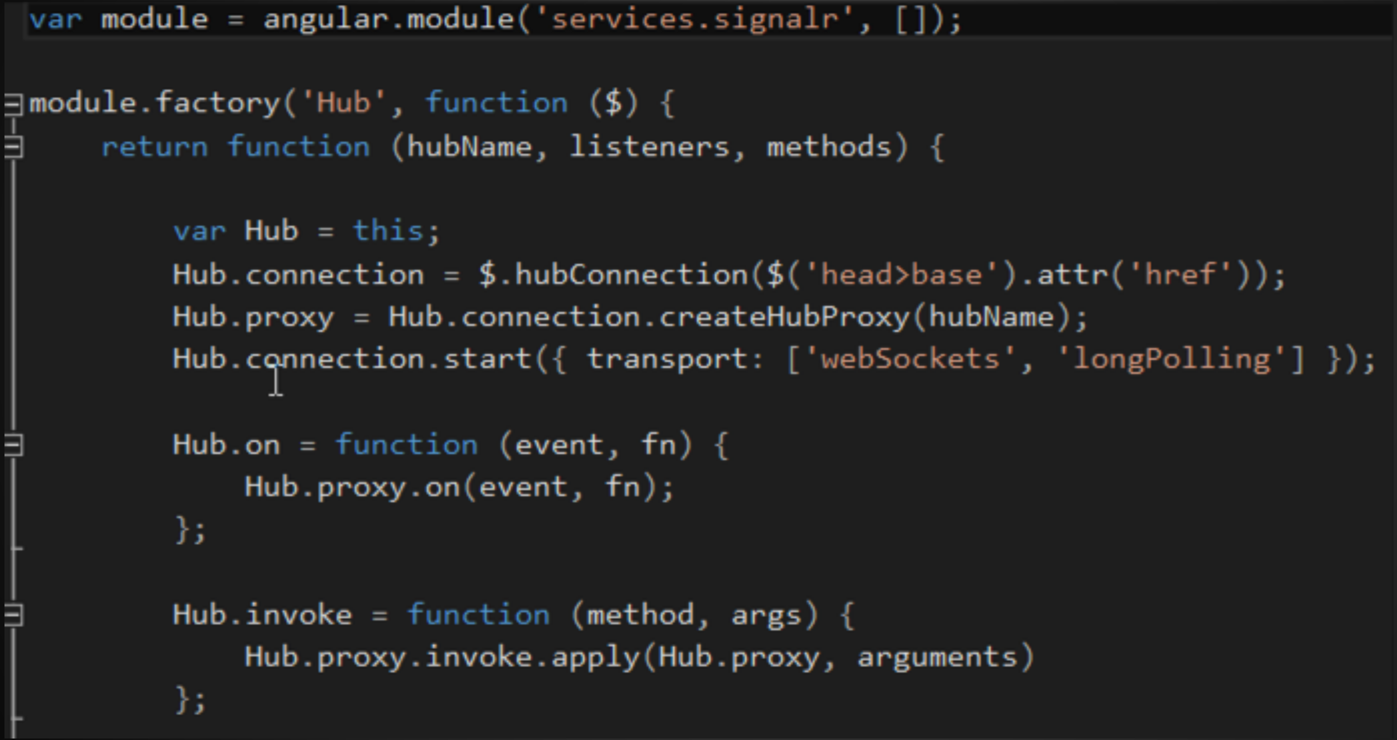
SignalR

Web SignalR Startup



```
public partial class Startup
{
    public void ConfigureSignalR(IAppBuilder app)
    {
        var hubConfiguration = new HubConfiguration();
        hubConfiguration.EnableDetailedErrors = true;
        hubConfiguration.EnableJavaScriptProxies = false;
        app.MapSignalR(hubConfiguration);
    }
}
```

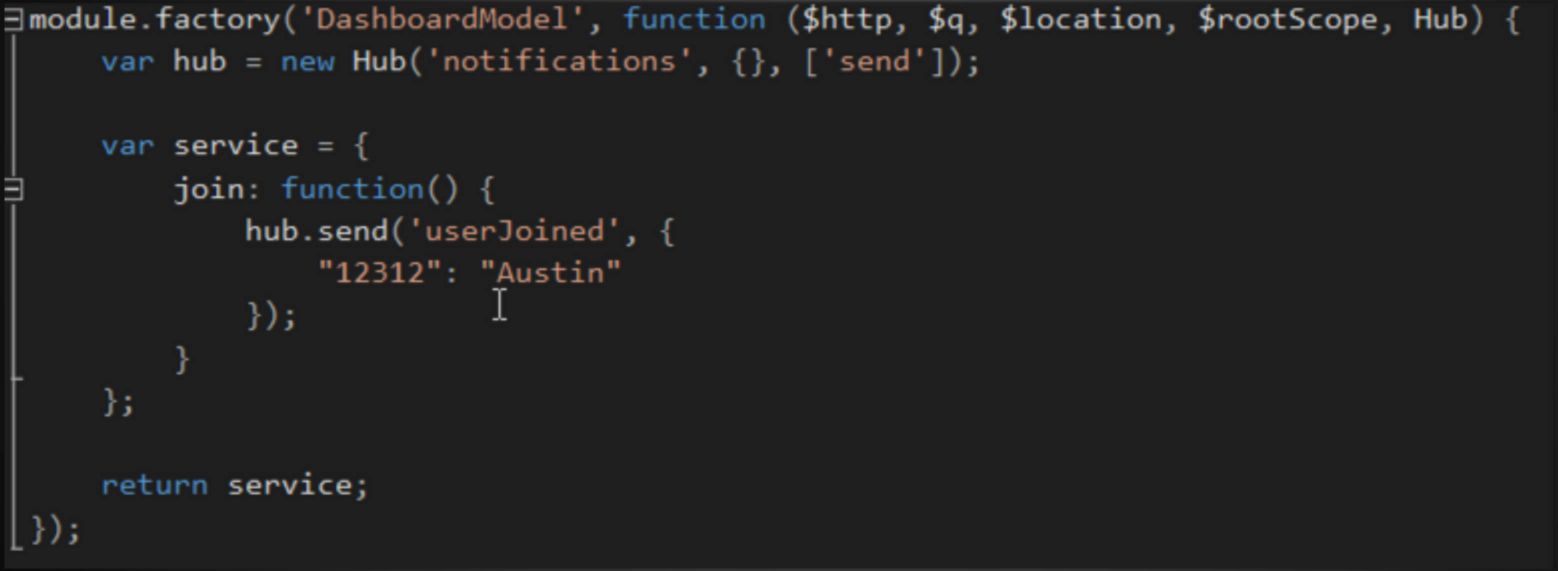
Angular SignalR Service



```
var module = angular.module('services.signalr', []);

module.factory('Hub', function ($) {
    return function (hubName, listeners, methods) {

        var Hub = this;
        Hub.connection = $.hubConnection($('head>base').attr('href'));
        Hub.proxy = Hub.connection.createHubProxy(hubName);
        Hub.connection.start({ transport: ['webSockets', 'longPolling'] });

        Hub.on = function (event, fn) {
            Hub.proxy.on(event, fn);
        };

        Hub.invoke = function (method, args) {
            Hub.proxy.invoke.apply(Hub.proxy, arguments)
        };
```

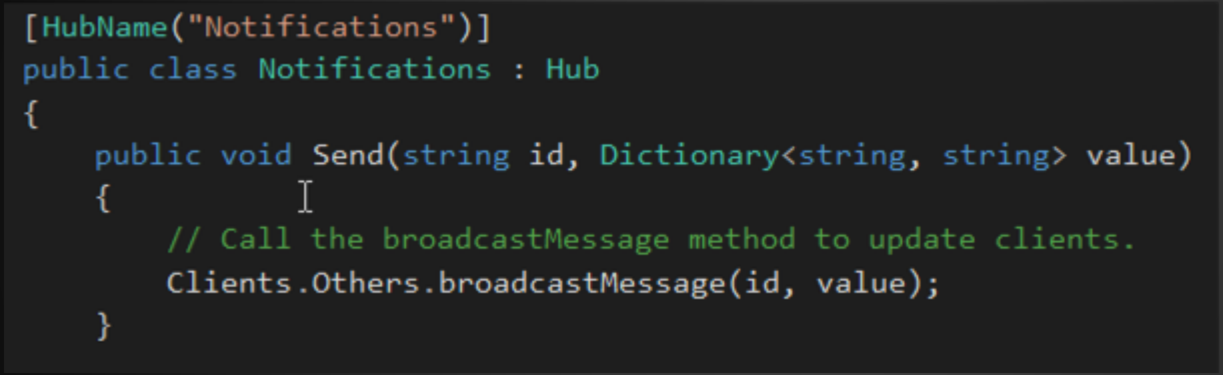
https://github.com/JustMaier/angular-signalr-hub

Angular SignalR Service Usage



```
module.factory('DashboardModel', function ($http, $q, $location, $rootScope, Hub) {
    var hub = new Hub('notifications', {}, ['send']);

    var service = {
        join: function() {
            hub.send('userJoined', {
                "12312": "Austin"
            });
        }
    };

    return service;
});
```

SignalR Hub - Distro Methods



```
[HubName("Notifications")]
public class Notifications : Hub
{
    public void Send(string id, Dictionary<string, string> value)
    {
        // Call the broadcastMessage method to update clients.
        Clients.Others.broadcastMessage(id, value);
    }
```

Define method name and arguments, declare to send to all or ‘other’ clients