```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Threading.Tasks;
 5  using Microsoft.AspNetCore.Builder;
 6  using Microsoft.AspNetCore.Hosting;
 7  using Microsoft.Extensions.Configuration;
 8  using Microsoft.Extensions.DependencyInjection;
 9  using DatabaseConnect;
10  using Microsoft.EntityFrameworkCore;
11  using Microsoft.AspNetCore.Authentication.JwtBearer;
12  using Microsoft.IdentityModel.Tokens;
13  using System.Text;
14  using Microsoft.AspNetCore.Rewrite;
15  using Microsoft.IdentityModel.Protocols.OpenIdConnect;
16  using Swashbuckle.AspNetCore.Swagger;
17  using Microsoft.Extensions.FileProviders;
18  using System.IO;
19
20  namespace LibraryAppMVC
21  {
22      public class Startup
23      {
24          public Startup(IConfiguration configuration)
25          {
26              Configuration = configuration;
27          }
28
29          public IConfiguration Configuration { get; }
30
31          // This method gets called by the runtime. Use this method to add
              services to the container.
32          public void ConfigureServices(IServiceCollection services)
33          {
34              /* JWT Resources
35               * https://auth0.com/blog/securing-asp-dot-net-core-2-applications-
                   with-jwts/
36               * https://blogs.msdn.microsoft.com/webdev/2017/04/06/jwt-validation-
                   and-authorization-in-asp-net-core/
37               * https://docs.microsoft.com/en-us/aspnet/core/migration/1x-to-2x/
                   identity-2x
38               */
39              services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
40                  .AddJwtBearer(opt =>
41                  {
42                      opt.Audience = "http://localhost:5001/";
43                      opt.Authority = "http://localhost:5000/";
44                      opt.Configuration = new OpenIdConnectConfiguration();
45                      // Stackexchange https://stackoverflow.com/
                         questions/37693516/unable-to-obtain-configuration-from-well-
                         known-openid-configuration/37973711
46                      opt.TokenValidationParameters = new TokenValidationParameters
```

```
47                      {
48                          ValidateIssuer = true,
49                          ValidateAudience = true,
50                          ValidateLifetime = true,
51                          ValidateIssuerSigningKey = true,
52                          ValidIssuer = Configuration["Jwt:Issuer"],
53                          ValidAudience = Configuration["Jwt:Issuer"],
54                          IssuerSigningKey = new SymmetricSecurityKey
                        (Encoding.UTF8.GetBytes(Configuration["Jwt:Key"])),
55                          ClockSkew = TimeSpan.Zero  // Lets tokens expire when
                        they say they will
56                      };
57                  });
58
59              services.AddSingleton(Configuration);
60
61              services.AddDbContext<Context>(options => options.UseSqlServer
                  (Configuration.GetConnectionString("TheDatabase")));
62              services.AddMvc()
63                  .AddJsonOptions(options =>
                      options.SerializerSettings.ReferenceLoopHandling =
                      Newtonsoft.Json.ReferenceLoopHandling.Ignore);
64              services.AddSwaggerGen(c =>
65              {
66                  c.SwaggerDoc("v1", new Info { Title = "FBLA Mobile App", Version
                      = "v1" });
67              });
68          }
69
70          // This method gets called by the runtime. Use this method to configure
                the HTTP request pipeline.
71          public void Configure(IApplicationBuilder app, IHostingEnvironment env)
72          {
73              var redirOpt = new RewriteOptions();
74              if (env.IsDevelopment())
75              {
76                  app.UseDeveloperExceptionPage();
77                  app.UseBrowserLink();
78              }
79              else
80              {
81                  app.UseExceptionHandler("/Home/Error");
82                  // HTTPS redirector
83                  redirOpt.AddRedirectToHttps();
84              }
85              app.UseRewriter(redirOpt);
86              app.UseAuthentication();
87              app.UseFileServer(new FileServerOptions
88              {
89                  FileProvider = new PhysicalFileProvider(
90              Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/images")),
91                  RequestPath = "/images",
```

```
92                  EnableDirectoryBrowsing = true
93              });
94              app.UseSwagger();
95              app.UseSwaggerUI(c =>
96              {
97                  c.SwaggerEndpoint("/swagger/v1/swagger.json", "FBLA Mobile App
                      v1");
98              });
99
100             app.UseMvc(routes =>
101             {
102                 routes.MapRoute(
103                     name: "default",
104                     template: "{controller=Home}/{action=Index}/{id?}");
105             });
106         }
107     }
108 }
109
```