```csharp
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Threading.Tasks;
 5  using ExploreHorsens.Models;
 6  using Microsoft.AspNetCore.Builder;
 7  using Microsoft.AspNetCore.Hosting;
 8  using Microsoft.AspNetCore.Http;
 9  using Microsoft.AspNetCore.Identity;
10  using Microsoft.AspNetCore.Mvc;
11  using Microsoft.EntityFrameworkCore;
12  using Microsoft.Extensions.Configuration;
13  using Microsoft.Extensions.DependencyInjection;
14
15
16  namespace ExploreHorsens
17  {
18      public class Startup
19      {
20          public Startup(IConfiguration configuration)
21          {
22              Configuration = configuration;
23          }
24
25          public IConfiguration Configuration { get; }
26          // This method gets called by the runtime. Use this method to add
                services to the container.
27          // For more information on how to configure your application, visit
                https://go.microsoft.com/fwlink/?LinkID=398940
28          public void ConfigureServices(IServiceCollection services)
29          {
30              services.AddMvc().SetCompatibilityVersion
                    (CompatibilityVersion.Version_2_1);
31
32              services.AddDbContext<BlogDataContext>(options =>
33              options.UseSqlServer(Configuration
                    ["ConnectionStrings:DefaultConnection"]));
34
35          }
36
37          // This method gets called by the runtime. Use this method to
                configure the HTTP request pipeline.
38          public void Configure(IApplicationBuilder app, IHostingEnvironment
                env)
39          {
40
41              var configuration = new ConfigurationBuilder()
42                                  .AddEnvironmentVariables()
43
44                                  .Build();
45
46              if (configuration.GetValue<bool>("EnableDeveloperExceptions"))
47              {
```

```
48                    app.UseDeveloperExceptionPage();
49                }
50
51            app.Use(async (context, next) =>
52            {
53                if (context.Request.Path.Value.Contains("invalid"))
54                    throw new Exception("ERROR!");
55
56                await next();
57            });
58
59            app.UseMvc(routes =>
60            {
61                routes.MapRoute("Default",
62                    "{controller=Default}/{action=Index}/{id?}"
63                );
64            });
65        }
66    }
67 }
68
```