Configuration Lab

Perform these labs on your own computer using Visual Studio 2022 to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Add Configuration Settings to JSON File

Open the **appsettings.Development.json** file and add the following JSON object shown in **bold** below.

```
{
  "AdvWorksAPI": {
    "InfoMessageDefault": "Problem Attempting to {Verb}
using the {ClassName} API. Please Contact Your System
Administrator.",
    "DefaultTitle": "Ms.",
    "DefaultEmail": "FirstName.LastName@AdvWorks.com"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  }
}
```

Open the **AdvWordsAPIDefaults.cs** file in the **EntityLayer** folder

Add the property **InfoMessageDefault** to this class.

```
public string InfoMessageDefault { get; set; }
```

Modify the constructor with the change shown in **bold** below.

```
public AdvWorksAPIDefaults()
{
  Created = DateTime.Now;
  InfoMessageDefault = string.Empty;
  DefaultTitle = "Mr.";
  DefaultEmail = "LastName.FirstName@adventure-
works.com";
}
```

Lab 2: Read Settings in Program.cs

Open the **Program.cs** file and insert the code shown in **bold** below.

```
// ********************************************
// Add and Configure Services
// ********************************************
builder.Services.AddSingleton<AdvWorksAPIDefaults,
AdvWorksAPIDefaults>();

// Read "AdvWorksAPI" section and add as a singleton
AdvWorksAPIDefaults settings = new();
builder.Configuration.GetSection("AdvWorksAPI").Bind(set
tings);
builder.Services.AddSingleton<AdvWorksAPIDefaults>(setti
ngs);

builder.Services.AddScoped<IRepository<Customer>,
CustomerRepository>();
```

Open the **CustomerRouter.cs** file and add a new private field

```
private readonly AdvWorksAPIDefaults _Settings;
```

Modify the constructor as shown in **bold** below.

```
public CustomerRouter(IRepository<Customer> repo,
ILogger<CustomerRouter> logger, AdvWorksAPIDefaults
settings) : base(logger)
{
  UrlFragment = "api/Customer";
  TagName = "Customer";
  _Repo = repo;
  _Settings = settings;
}
```

Locate the Get() method and in the catch block where you set the **InfoMessage** property, replace it with the following code shown in **bold**.
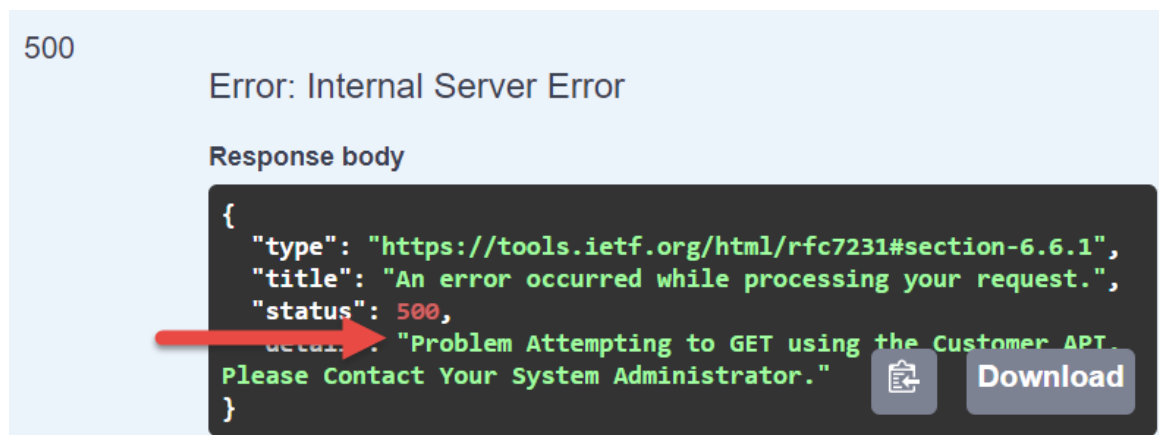
```
catch (Exception ex) {
  // Return generic message for the user
  InfoMessage = _Settings.InfoMessageDefault
      .Replace("{Verb}", "GET")
      .Replace("{ClassName}", "Customer");

  ErrorLogMessage = "Error in CustomerRouter.Get()";

  ret = HandleException(ex);
}
```

Try it Out

Run the application and click on the **GET /api/Customer** button.

Remove the Throwing of the Exception

In the Get() method and **COMMENT OUT** the *throw new ApplicationException("ERROR!");* line.