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.

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

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 delete the builder.Services.Singleton<AdvWorksAPIDefaults, AdvWorksAPIDefaults>() call.

Where that line of code was, add the following code.

```
// Read "AdvWorksAPI" section
// Use the IOptionsMonitor<AdvWorksAPIDefaults> in
controller's constructor
builder.Services.Configure<AdvWorksAPIDefaults>(builder.
Configuration.GetSection("AdvWorksAPI"));
```

This code should **be placed before** adding the Scoped() CustomerRepository.

Inject IOptionsMonitor<T>

Open the **CustomerController.cs** file and add a new readonly field.

```
private readonly AdvWorksAPIDefaults _Settings;
```

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

```
public CustomerController(IRepository<Customer> repo,
ILogger<CustomerController> logger,
IOptionsMonitor<AdvWorksAPIDefaults> settings) :
base(logger)
{
  _Repo = repo;
  _Settings = settings.CurrentValue;
}
```

In the catch block of the Get() method, change the setting of the **InfoMessage** property to look like the following:

```
InfoMessage =
_Settings.InfoMessageDefault.Replace("{Verb}",
"GET").Replace("{ClassName}", "Customer");
```

Try it Out

Set a breakpoint on the above line.

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

Check to make sure the InfoMessage is set with the data from the settings file.

Remove Throwing of Exception

You can now **COMMENT OUT** the *throw new ApplicationException("ERROR!");* line in the Get() method of the CustomerController class.