

go deploy



Module 13 ▼

Screenshots ▼

## Module 13: Implementing Web APIs

### Lab: Implementing Web APIs

#### ? Scenario

You have been asked to create a web-based restaurant application for your organization's customers. To do this you need to create a page showing all the restaurant branches, enable users to order a reservation for a selected restaurant branch, add a wanted ad page, and allow submitting an application to a selected job.

You will create a server-side Web API application and a client-side ASP.NET Core application. In the client-side application, you will call the Web API actions by using HttpClient and jQuery.

### Exercise 1: Adding Actions and Calling them by using Microsoft Edge

#### ? Scenario

In this exercise, you will first add a controller and an action to a Web API application. You will then run the application and view the outcome by using Microsoft Edge. After that, you will add a controller and an action that gets a parameter. You will then run the application and view the outcome by using Microsoft Edge. Finally, you will add a Post action to the Web API application.

The main tasks for this exercise are as follows:

- Add a controller and an action to a Web API application
- Run the application
- Add a controller and an action that gets a parameter
- Run the application
- Add a Post action to a Web API application

### Task 1: Add a controller and an action to a Web API application


- ☐ 1. Navigate to **D:\Allfiles\Mod13\Labfiles\01\_Restaurant\_begin**, and then double-click **Restaurant.sln**.

**⚠ Note:** If a **Security Warning for Server** dialog box appears, verify that the **Ask me for every project in this solution** check box is cleared, and then click OK.


## go deploy




- ☐ 3. In the **Add New Scaffolded Item** dialog box, choose **API** from the left-hand navigation, click **API Controller - Empty**, and then click **Add**.

- ☐ 4. In the **Add New Item - Server** dialog box, in the **Name:** textbox, type  **RestaurantBranchesController**, and then click **Add**.


- ☐ 5. In the **RestaurantBranchesController.cs** code window, locate the following code:

```
 using Microsoft.AspNetCore.Mvc;
```

- ☐ 6. Ensure that the cursor is at the end of the **Microsoft.AspNetCore.Mvc** namespace, press Enter, and then type the following code:


```
 using Server.Models;
```

- ☐ 7. In the **RestaurantBranchesController.cs** code window, place the cursor after the second { (opening brace) sign, press Enter, and then type the following code:


```
 private RestaurantContext _context;

public RestaurantBranchesController(RestaurantContext context)
{
    _context = context;
}
```

- ☐ 8. Ensure that the cursor is at the end of the **RestaurantBranchesController** constructor code block, press Enter twice, and then type the following code:

```
 [HttpGet]
public ActionResult<List<RestaurantBranch>> Get()
{
}
```

- ☐ 9. In the **Get** action code block, type the following code:

```
 var branches = from r in _context.RestaurantBranches
                orderby r.City
                select r;
return branches.ToList();
```

## Task 2: Run the application

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In Solution Explorer, right-click **Server**, and then click **Set as StartUp Project**.


## go deploy




 **Note:** The browser displays a list of branches in the JSON format.

- ☐ 4. In Microsoft Edge, click **Close**.

### Task 3: Add a controller and an action that gets a parameter


- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Server**, right-click **Controllers**, point to **Add**, and then click **Controller...**
- ☐ 2. In the **Add New Scaffolded Item** dialog box, choose **API** from the left-hand navigation, click **API Controller - Empty**, and then click **Add**.
- ☐ 3. In the **Add New Item - Server** dialog box, in the **Name:** textbox, type  **ReservationController**, and then click **Add**.
- ☐ 4. In the **ReservationController.cs** code window, locate the following code:

```
 using Microsoft.AspNetCore.Mvc;
```


- ☐ 5. Ensure that the cursor is at the end of the **Microsoft.AspNetCore.Mvc** namespace, press Enter, and then type the following code:

```
 using Server.Models;
```

- ☐ 6. In the **ReservationController.cs** code window, place the cursor after the second { (opening brace) sign, press Enter, and then type the following code:

```
 private RestaurantContext _context;  
  
public ReservationController(RestaurantContext context)  
{  
    _context = context;  
}
```

- ☐ 7. Ensure that the cursor is at the end of the **ReservationController** constructor code block, press Enter twice, and then type the following code:

```
 [Route("{id:int}")]  
public ActionResult<OrderTable> GetById(int id)  
{  
}
```


- ☐ 8. In the **GetById** action code block, type the following code:

go deploy



```
{  
    return NotFound();  
}  
  
return order;
```

#### Task 4: Run the application


- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 3. In **Microsoft Edge**, in the address bar, type  [http://localhost:\[port\]/api/Reservation/1](http://localhost:[port]/api/Reservation/1), and then press Enter.

 **Note:** The browser displays an order with id=1 in the JSON format.


- ☐ 4. In Microsoft Edge, click **Close**.

#### Task 5: Add a Post action to a Web API application

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Server**, expand **Controllers**, and then click **ReservationController**.
- ☐ 2. In the **ReservationController.cs** code window, ensure that the cursor is at the end of the **GetById** action code block, press Enter twice, and then type the following code:

```
 [HttpPost]  
public ActionResult<OrderTable> Create(OrderTable orderTable)  
{  
}
```

- ☐ 3. In the **Create** action code block, type the following code:

```
 _context.Add(orderTable);  
_context.SaveChanges();  
return CreatedAtAction(nameof(GetById), new { id = orderTable.Id }, orderTable);
```

✓ **Results:** In this exercise, you added controllers and actions to a Web API application, and called them by using Microsoft Edge.

#### Exercise 2: Calling a Web API by using Server-Side Code

## go deploy



In this exercise, you will call the Web API you developed in the previous exercise by using the `HttpClient` class. To do this, you will first register the `IHttpClientFactory` service in the `Program.cs` file. You will then create an MVC controller and use the `HttpClient` class to call a `Get` action in the Web API. After that, you will create another MVC controller and use the `HttpClient` class to call a `Post` action in the Web API. Finally, you will add an action to the MVC controller in which you will use the `HttpClient` class to call a `Get` action in the Web API that gets a parameter.


The main tasks for this exercise are as follows:

- Calling a Web API `Get` method
- Run the application
- Calling a Web API `Post` method
- Calling a Web API `Get` method that gets a parameter
- Run the application


## Task 1: Calling a Web API `Get` method

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, click **Program.cs**.

- ☐ 2. In the **Program.cs** code window, locate the following code:

```
 builder.Services.AddControllersWithViews();
```

- ☐ 3. Place the cursor at the end of the located code, press `Enter`, and then type the following code:


```
 builder.Services.AddHttpClient();
```

- ☐ 4. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, right-click **Controllers**, point to **Add**, and then click **Controller...**

- ☐ 5. In the **Add New Scaffolded Item** dialog box, choose **API** from the left-hand navigation, click **MVC Controller - Empty**, and then click **Add**.

- ☐ 6. In the **Add New Item - Client** dialog box, in the **Name:** textbox, type  RestaurantBranchesController, and then click **Add**.

- ☐ 7. In the **RestaurantBranchesController.cs** code window, locate the following code:

```
 using Microsoft.AspNetCore.Mvc;
```

- ☐ 8. Ensure that the cursor is at the end of the **Microsoft.AspNetCore.Mvc** namespace, press `Enter`, and then type the following code:

## go deploy



- ☐ 9. In the **RestaurantBranchesController.cs** code window, select the following code:

```
public IActionResult Index()
{
    return View();
}
```

- ☐ 10. Replace the selected code with the following code:

```
private IHttpClientFactory _httpClientFactory;

public RestaurantBranchesController(IHttpClientFactory httpClientFactory)
{
    _httpClientFactory = httpClientFactory;
}
```

- ☐ 11. Ensure that the cursor is at the end of the **RestaurantBranchesController** constructor code block, press Enter twice, and then type the following code:

```
public async Task<IActionResult> Index()
{
}
```

- ☐ 12. In the **Index** action code block, type the following code:

```
HttpClient httpClient = _httpClientFactory.CreateClient();
httpClient.BaseAddress = new Uri("http://localhost:54517");
HttpResponseMessage response = await httpClient.GetAsync("api/RestaurantBranches");
if (response.IsSuccessStatusCode)
{
    IEnumerable<RestaurantBranch> restaurantBranches = await response.Content.ReadAsJsonAsync();
    return View(restaurantBranches);
}
else
{
    return View("Error");
}
```

- ☐ 13. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, expand **Views**, expand **RestaurantBranches**, and then click **Index.cshtml**.

**Note:** Examine the file content.

## Task 2: Run the application

## go deploy



- ☐ 2. In the **launchSettings.json** code window, select the following code:

```
"profiles": {  
  "IIS Express": {  
    "commandName": "IISExpress",  
    "launchBrowser": true,  
    "launchUrl": "api/RestaurantBranches",  
    "environmentVariables": {  
      "ASPNETCORE_ENVIRONMENT": "Development"  
    }  
  },  
}
```

- ☐ 3. Replace the selected code with the following code:

```
"profiles": {  
  "IIS Express": {  
    "commandName": "IISExpress",  
    "launchBrowser": false,  
    "environmentVariables": {  
      "ASPNETCORE_ENVIRONMENT": "Development"  
    }  
  },  
}
```

- ☐ 4. In the **Restaurant - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 5. In Solution Explorer, right-click **Server**, and then click **Set as StartUp Project**.
- ☐ 6. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 7. In Solution Explorer, right-click **Client**, and then click **Set as StartUp Project**.
- ☐ 8. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

**Note:** The browser displays the restaurant branches.

- ☐ 9. In Microsoft Edge, click **Close**.


### Task 3: Calling a Web API Post method

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, right-click **Controllers**, point to **Add**, and then click **Controller...**
- ☐ 2. In the **Add New Scaffolded Item** dialog box, click **MVC Controller - Empty**, and then click **Add**.


## go deploy




- ☐ 4. In the **ReservationController.cs** code window, locate the following code:

```
 using Microsoft.AspNetCore.Mvc;
```


- ☐ 5. Ensure that the cursor is at the end of the **Microsoft.AspNetCore.Mvc** namespace, press Enter, and then type the following code:

```
 using Client.Models;
```


- ☐ 6. In the **ReservationController.cs** code window, select the following code:

```
 public IActionResult Index()  
{  
    return View();  
}
```

- ☐ 7. Replace the selected code with the following code:

```
 private IHttpClientFactory _httpClientFactory;  
  
public ReservationController(IHttpClientFactory httpClientFactory)  
{  
    _httpClientFactory = httpClientFactory;  
}
```

- ☐ 8. Ensure that the cursor is at the end of the **ReservationController** constructor code block, press Enter twice, and then type the following code:

```
 [HttpGet]  
public async Task<IActionResult> Create()  
{  
    await PopulateRestaurantBranchesDropDownListAsync();  
    return View();  
}
```

- ☐ 9. in the **ReservationController.cs** code window, ensure that the cursor is at the end of the **Create** action code block, press Enter twice, and then type the following code:



## go deploy



```

{
    HttpClient httpClient = _httpClientFactory.CreateClient();
    HttpResponseMessage response = await httpClient.PostAsJsonAsync("http://localho
    if (response.IsSuccessStatusCode)
    {
        OrderTable order = await response.Content.ReadAsAsync<OrderTable>();
        return RedirectToAction("ThankYou", new { orderId = order.Id});
    }
    else
    {
        return View("Error");
    }
}

```

- ☐ 10. In the **ReservationController.cs** code window, ensure that the cursor is at the end of the **CreatePostAsync** action code block, press Enter twice, and then type the following code:

```

private async Task PopulateRestaurantBranchesDropDownListAsync()
{
    HttpClient httpClient = _httpClientFactory.CreateClient();
    httpClient.BaseAddress = new Uri("http://localhost:54517");
    HttpResponseMessage response = await httpClient.GetAsync("api/RestaurantBranche
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<RestaurantBranch> restaurantBranches = await response.Content.R
        ViewBag.RestaurantBranches = new SelectList(restaurantBranches, "Id", "City
    }
}

```

- ☐ 11. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **Views**, expand **Reservation**, and then click **Create.cshtml**.

**Note:** Examine the file content.

## Task 4: Calling a Web API Get method with parameter

- ☐ 1. In the **ReservationController.cs** code window, ensure that the cursor is at the end of the **PopulateRestaurantBranchesDropDownListAsync** action code block, press Enter twice, and then type the following code:

```

public async Task<IActionResult> ThankYou(int orderId)
{
}

```

- ☐ 2. In the **ThankYou** action code block, type the following code:

## go deploy



```
HttpResponseMessage response = await httpClient.GetAsync("api/Reservation/" + order
if (response.IsSuccessStatusCode)
{
    OrderTable orderResult = await response.Content.ReadAsAsync<OrderTable>();
    return View(orderResult);
}
else
{
    return View("Error");
}
```

- ☐ 3. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **Views**, under **Reservation**, click **ThankYou.cshtml**.

 **Note:** Examine the file content.

## Task 5: Run the application

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In Solution Explorer, right-click **Server**, and then click **Set as StartUp Project**.
- ☐ 3. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 4. In Solution Explorer, right-click **Client**, and then click **Set as StartUp Project**.
- ☐ 5. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 6. In the **Microsoft Edge** window, in the menu bar, click **Reservation**.
- ☐ 7. On **Reservation**, in the **Restaurant Branch** drop-down box, select any restaurant.
- ☐ 8. On **Reservation**, in the **First Name** box, type your first name.
- ☐ 9. On **Reservation**, in the **Last Name** box, type your second name.
- ☐ 10. On **Reservation**, in the **Phone Number** box, type some random digits.
- ☐ 11. On **Reservation**, in the **Reservation Time** box, use the picker to choose a date and time.
- ☐ 12. On **Reservation**, in the **Dinner Guests** box, type a number between 1 and 20, and then click **Make a Reservation**.
- ☐ 13. In Microsoft Edge, click **Close**.

## Exercise 3: Calling a Web API by Using jQuery


### ? Scenario

In this exercise, you will call a Web API by using jQuery. You will first create an MVC controller and use jQuery to call a Get action in the Web API. After that, you will create another MVC controller and use jQuery to call a Post action in the Web API.

The main tasks for this exercise are as follows:

- Calling a Web API Get method by using jQuery
- Run the application
- Calling a Web API Get method by using HttpClient
- Calling a Web API Post method by using jQuery
- Run the application

### Task 1: Calling a Web API Get method by using jQuery

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, expand **wwwroot**, right-click **js**, point to **Add**, and then click **New Item....**
- ☐ 2. In the **Add New Item - Client** dialog box, in the templates pane, find and click **JavaScript File**.
- ☐ 3. In the **Add New Item -- Client** dialog box, in the **Name:** box, type  wanted-ad-get, and then click **Add**.
- ☐ 4. In the **wanted-ad-get.js** code window, type the following code:

```
 $(function () {  
    $.ajax({  
        method: "GET",  
        url: "http://localhost:54517/api/RestaurantWantedAd",  
        contentType: "application/json; charset=utf-8",  
        dataType: "json"  
    }).done(function (data) {  
    }).fail(function () {  
        alert('An error has occurred');  
    });  
});
```

- ☐ 5. In the **wanted-ad-get.js** code window, locate the following code:

```
 }).done(function (data) {
```

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

☐ 7. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **Views**, expand **WantedAd**, and then click **Index.cshtml**.

☐ 8. In the **Index.cshtml** code window, locate the following code:

☐ 9. Place the cursor at the end of the located code, press Enter twice, and then type the following code:

## Task 2: Run the application

- 12 of 16


## go deploy




 **Note:** The browser displays the jobs that required in a restaurant.


- ☐ 7. In Microsoft Edge, click **Close**.

### Task 3: Calling a Web API Get method by using HttpClient


- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, right-click **Controllers**, point to **Add**, and then click **Controller...**
- ☐ 2. In the **Add New Scaffolded Item** dialog box, click **MVC Controller - Empty**, and then click **Add**.
- ☐ 3. In the **Add New Item - Client** dialog box, in the **Name:** textbox, type  JobApplicationController, and then click **Add**.
- ☐ 4. In the **JobApplicationController.cs** code window, locate the following code:

```
 using Microsoft.AspNetCore.Mvc;
```


- ☐ 5. Ensure that the cursor is at the end of the **Microsoft.AspNetCore.Mvc** namespace, press Enter, and then type the following code:

```
 using Client.Models;
```

- ☐ 6. In the **JobApplicationController.cs** code window, select the following code:

```
 public IActionResult Index()  
{  
    return View();  
}
```

- ☐ 7. Replace the selected code with the following code:

```
 private IHttpClientFactory _httpClientFactory;  
  
public JobApplicationController(IHttpClientFactory httpClientFactory)  
{  
    _httpClientFactory = httpClientFactory;  
}
```


- ☐ 8. Ensure that the cursor is at the end of the **JobApplicationController** constructor code block, press Enter twice, and then type the following code:

## go deploy



```
{  
    await PopulateEmployeeRequirementsDropDownListAsync();  
    return View();  
}
```


- ☐ 9. At the end of the **Create** action code block, press Enter, and then type the following code:

```
 private async Task PopulateEmployeeRequirementsDropDownListAsync()  
{  
    HttpClient httpClient = _httpClientFactory.CreateClient();  
    httpClient.BaseAddress = new Uri("http://localhost:54517");  
    HttpResponseMessage response = await httpClient.GetAsync("api/RestaurantWantedA  
    if (response.IsSuccessStatusCode)  
    {  
        IEnumerable<EmployeeRequirements> employeeRequirements = await response.Con  
        ViewBag.EmployeeRequirements = new SelectList(employeeRequirements, "Id", "  
    }  
}
```

- ☐ 10. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **Views**, expand **JobApplication**, and then click **Create.cshtml**.

 **Note:** Examine the file content.

- ☐ 11. In the **JobApplicationController.cs** code window, ensure that the cursor is at the end of the **PopulateEmployeeRequirementsDropDownListAsync** action code block, press Enter, and then type the following code:

```
 public IActionResult ThankYou()  
{  
    return View();  
}
```

- ☐ 12. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **Views**, expand **JobApplication**, and then click **ThankYou.cshtml**.

 **Note:** Examine the file content.

## Task 4: Calling a Web API Post method by using jQuery

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **wwwroot**, right-click **js**, point to **Add**, and then click **New Item....**
- ☐ 2. In the **Add New Item - Client** dialog box, in the templates pane, find and click **JavaScript File**.

## go deploy




- ☐ 4. In the **wanted-ad-post.js** code window, type the following code:

```
 $(function() {  
    $('#btn-post').click(function (e) {  
        if ($('#submit-form').valid()) {  
            var formData = {};  
            $('#submit-form').serializeArray().map(function (item) {  
                item.name = item.name[0].toLowerCase() + item.name.slice(1);  
                if (formData[item.name]) {  
                    if (formData[item.name] === "string") {  
                        formData[item.name] = [formData[item.name]];  
                    }  
                    formData[item.name].push(item.value);  
                } else {  
                    formData[item.name] = item.value;  
                }  
            });  
            e.preventDefault();  
            $.ajax({  
                method: "POST",  
                url: "http://localhost:54517/api/job",  
                data: JSON.stringify(formData),  
                contentType: "application/json; charset=utf-8"  
            }).done(function () {  
                location.href = 'http://localhost:54508/JobApplication/ThankYou';  
            }).fail(function () {  
                alert('An error has occurred');  
            });  
        }  
    });  
});
```

- ☐ 5. In the **Restaurant - Microsoft Visual Studio** window, in Solution Explorer, under **Client**, under **Views**, under **JobApplication**, click **Create.cshtml**.

- ☐ 6. In the **Create.cshtml** code window, locate the following code:

```
         </form>  
    </div>  
</div>
```

- ☐ 7. Place the cursor at the end of the located code, press Enter twice, and then type the following code:

go deploy



```
    <script src=~\lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.mi  
    <script src=~\js/wanted-ad-post.js"></script>  
  }
```

### Task 5: Run the application

- ☐ 1. In the **Restaurant - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In Solution Explorer, right-click **Server**, and then click **Set as StartUp Project**.
- ☐ 3. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 4. In Solution Explorer, right-click **Client**, and then click **Set as StartUp Project**.
- ☐ 5. In the **Restaurant - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 6. In the first **Microsoft Edge** window, in the menu bar, click **Submit Job Application**.
- ☐ 7. On **Submit Job Application**, in the **Job title** box, select **Manager**.
- ☐ 8. On **Submit Job Application**, in the **First name** box, type your first name.
- ☐ 9. On **Submit Job Application**, in the **Last name** box, type your last name.
- ☐ 10. On **Submit Job Application**, in the **Phone number** box, type some random digits.
- ☐ 11. On **Submit Job Application**, in the **Email** box, type a plausible email address
- ☐ 12. On **Submit Job Application**, in the **Address** box, type some random text, and then click **Apply For a Job**.
- ☐ 13. In Microsoft Edge, click **Close**.
- ☐ 14. In the **Restaurant - Microsoft Visual Studio** window, on the **File** menu, click **Exit**.

✓ **Results:** In this exercise, you enabled users to see the wanted ads to find a new job, and enabled them to apply for a job.