Module14

Implementing Web APIs in ASP.NET MVC Web Applications

Module Overview

- Developing a Web API
- Calling a Web API from Mobile and Web Applications

Lesson 1: Developing a Web API

- What Is a Web API?
- Routing in Web API
- Creating a Web API for an MVC 4 Web Application
- RESTful Services
- Data Return Formats
- Using Routes and Controllers in Web APIs
- Demonstration: How to Explore a Web API by Using Internet Explorer

What Is a Web API?

Web API:

- Helps create REST-style APIs
- Enables external systems to use the business logic implemented in your application
- Uses URLs in requests and helps obtain results in the JSON format
- Is ideal for mobile application integration

Routing in Web API

Characteristics of routing in Web API:

- You can use API controller names and a naming convention for actions to route Web API requests
- Alternatively, you can use the following attributes to control the mapping of HTTP requests (HTTP verb+URL) to actions in the controller:
  - The **HttpGet**, **HttpPut**, **HttpPost**, or **HttpDelete** attributes
  - The **AcceptVerbs** attribute
  - The **ActionName** attribute

Creating a Web API for an MVC Web Application

itucation

To create a Web API for an MVC application:

1. Implement a Web API template in your project:
    1. In the **New Project** dialog box, click **ASP.NET MVC Web Application**
    2. In the **Select a Template** box of the **New ASP.NET MVC Project** dialog box, click **Web API**

2. Add an MVC API controller class to the project:
    - Hosts application code for handling requests
    - Derives from the ApiController base class

3. Add action methods to the controller class

RESTful Services

ITucation

Characteristics of a REST Service:

- Can be called to retrieve business information from the server
- Can create, update, and delete information in a database through HTTP operations
- Uses URLs to uniquely identify the entity that it operates on
- Uses HTTP verbs to identify the operation that the application needs to perform. The HTTP verbs include:
  - **GET**
  - **POST**
  - **PUT**
  - **DELETE**

Data Return Formats

- Web API can return data in JSON or XML formats
- Web API uses the media formatter to:
  - Format or serialize the information that a Web API REST service returns
  - Control the media type in the HTTP header
  - Format all content that the server renders to client systems
- Media formatter classes inherit from the **MediaTypeFormatter** class and the **BufferedMediaTypeFormatter** class

Using Routes and Controllers in Web APIs

Routing in ASP.NET MVC applications involves the following:

- ASP.NET adds a default route to:
  - Map a URL and a controller
  - Support the operations of the REST-style Web APIs
- You can modify the default route to include multiple actions in the same HTTP method
- You can use the **WebApiConfig** class to:
  - Modify the routing
  - Enable multiple versions of API to coexist in the same project

Demonstration: How to Explore a Web API by Using Internet Explorer

- In this demonstration, you will see how to:
  - Add a new Web API controller to an MVC web application
  - Create actions in a Web API controller
  - Call Web API actions from Internet Explorer
  - View JSON code returned by an MVC Web API

Lesson 2: Calling a Web API from Mobile and Web Applications

- Calling Web APIs by Using Server-Side Code
- Calling Web APIs by Using jQuery Code
- Calling Web APIs Using Windows Phone Applications

Calling Web APIs by Using Server-Side Code

To call Web APIs by using server-side code:

- Install the **Microsoft.AspNet.WebApi.Client** NuGet package
- Add code to initialize the **HttpClient** class
- Add code to create requests by using **GetAsync** and **ReadAsAsync**

Calling Web APIs by Using jQuery Code

Using jQuery to call Web API services provides you the following options:

- You can use the jQuery **ajax** function to call Web API services
- You can set the **dataType** parameter of the **ajax** function to **json**
- You can use **JSON.stringify()** in the **data** parameter of the **ajax** function to serialize the JavaScript objects into JSON objects

Calling Web APIs Using Windows Phone Applications

itucation

To call Web APIs by using the JSON.NET library:

- Download the JSON.NET library from http://json.codeplex.com
- Add code to create requests by using the **WebClient** class
- Add the **DownloadStringCompleted** event handler to deserialize the results into .NET objects

Lab: Implementing APIs in ASP.NET MVC 4 Web Applications

- Exercise 1: Adding a Web API to the Photo Sharing Application
- Exercise 2: Using the Web API for a Bing Maps Display

Estimated Time: 60 minutes

Lab Scenario

Your manager wants to ensure that the photos and information stored in the Photo Sharing application can be integrated with other data in web mash-ups, mobile applications, and other locations. To re-use such data, while maintaining security, you need to implement a RESTful Web API for the application. You will use this Web API to display the locations of photos on a Bing Maps page.

Lab Review

- How do the API actions you added to the PhotoApiController controller in Exercise 1 differ from other actions in MVC controllers?

Module Review and Takeaways

- Real-world Issues and Scenarios
- Review Question(s)