



MODULE 02

WEB API 2

MODULE TOPICS

ApiController
Configuration and Routing
Formatters
Action Methods
Model Binding

APICONTROLLER

- Controller classes inherit from ApiController
- Web API maps HTTP requests to controller methods
 - These are called Action Methods
- Action methods can return data which is made into an HTTP response
- The HTTP response is unformatted data as compared to MVC for Web Forms
- Supports automatic content negotiation via the request Accept header such as XML or JSON
- Asynchronous by design

WALKTHRU - APICONTROLLER

CONFIGURATION

```
public static void Register(HttpConfiguration config)
{
    System.Diagnostics.Debug.WriteLine(GlobalConfiguration.Configuration.Routes.Count);
    System.Diagnostics.Debug.WriteLine(config.Routes.Count);
}
```

- Web API configuration values are stored in an instance of `HttpConfiguration`
- `GlobalConfiguration.Configuration` and `WebApiConfig.Register()` gives access to it
- Used for routing, filters, formatters, etc

ROUTING

```
// Web API routes

// Enable Attribute Routing
config.MapHttpAttributeRoutes();

// Convention Based Routing - Default Route
config.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{controller}/{id}",
    defaults: new { id = RouteParameter.Optional }
);
```

ROUTING

- Routing can be done using `HttpConfiguration.Routes` for global routes
- Or Attribute Routing such as `[Route("api/v2/HelloWorld")]`
- Routes can use filters, Regular Expressions, and other features to configure routing logic
- Routing should be kept simple to avoid confusion and loop holes

FORMATTERS

- Formatters consume and produce body content
- 3 Built-In Formatters (JSON, XML, Form Data)
- Other formatters can be created / added as needed
- Query String Mappings can be used to select a formatter

WALKTHRU - FORMATTERS

ACTION METHODS

```
public class HelloWorldController : ApiController
{
    // GET: api/HelloWorld
    public string Get()
    {
        return "Get: Hello World";
    }
}
```

ACTION METHOD RETURN TYPES

Return Type	Response Created
void	204 No Content
HttpResponseMessage	Converted directly to an HTTP response message
IHttpActionResult	ExecuteAsync() creates an HttpResponseMessage
Other type	Serialize return value into the response body

ERRORS AND EXCEPTIONS

- When a problem happens processing a request, an error or exception can be used
- `HttpResponseException` is used to send back a HTTP Status Code
- `Request.CreateErrorResponse()` can be used to return Status Codes and Messages
- Helper methods can also return error Status Codes `NotFound()` for example

WALKTHRU - ACTION METHODS RETURN TYPES & EXCEPTIONS

ACTION METHOD PARAMETERS

WEB API ASSUMES

- Simple types are taken from non-body values, aka query string
 - Simple types are primitives (int, bool, double, etc) TimeSpan, DateTime, Guid, decimal, string
 - Attribute [FromBody] is used to force Web API to read a simple type from the body
- Complex types are taken from the body
 - Attribute [FromUri] is used to force Web API to read a complex type from the URL

ACTION METHOD PARAMETERS

- Only a single object can come from the body based on how it reads the request
- Request bodies are handled by media formatters
- Query string URL data is handled by Type Converters and Model Binders
- Web API will automatically call the appropriate Action Method based on the data / parameters
- This allows Action Methods to be overridden by Verb

WALKTHRU - ACTION METHODS PARAMETER BINDING

ANY QUESTIONS?