

10 Ways to Build Web Services in .NET ASP.NET MVC3

Chad McCallum
@ChadEmm



Module Outline

- **Introduction to ASP.NET MVC3**
- **Review ASP.NET MVC3 Configuration**
- **Create our first controller and action method**
- **Create an action method that takes a route parameter**
- **Create action methods that use a POST verb, a PUT verb, and request content**
- **Create an action method that uses a DELETE verb**
- **Change our response data format based on the Accept request header**
- **Use HttpClient to download data from the web service**
- **Use HttpClient to send data to the web service**
- **Review**

ASP.NET MVC3

- **Model – View – Controller**

- Model: Data and the operations
- Controller: Handles the interaction between the client and the server
- View: A representation of data (HTML, JSON, XML, etc)

- **Route**

- Maps between request and appropriate controller & action
- Uses Uri Templates
 - `/ {controller} / {action} / {param}`

ASP.NET MVC3

■ **Convention over Configuration**

- Relies on naming conventions and other standards to avoid configuration steps
- Controllers : /Controllers folder, extend Controller
- Views: /{Controller}/{View}.cshtml
- Routes: /{Controller}/{Method}

Review

- **Introduction to ASP.NET MVC3**
- **Review ASP.NET MVC3 Configuration**
- **Create our first controller and action method**
- **Create an action method that takes a route parameter**
- **Create action methods that use a POST verb, a PUT verb, and request content**
- **Create an action method that uses a DELETE verb**
- **Change our response data format based on the Accept request header**
- **Use HttpClient to download data from the web service**
- **Use HttpClient to send data to the web service**

ASP.NET MVC3 – Convention over Configuration

- **MVC3 relies on convention to replace configuration files**
 - Uses reflection to automatically include code that extends certain interfaces or base classes (Controller, IValueProvider, etc.)
 - Routes define how URIs are mapped to Controller and Action methods
 - Views require a certain folder and naming structure to match Controllers

ASP.NET MVC3 - ValueProviders

- **MVC3 comes with a few baked-in value providers**
 - Route, Form (request body), File, Json, Query String
- **Value Providers automatically deserialize data from a request and pass it into a Controller's method if present**
 - For GetPun(int id), the ID could come from a route parameter, the Query String, or Form Data

ASP.NET MVC3 - ActionResult

- **Controller methods return an ActionResult object**
 - Consists of a single ExecuteResult method, which is responsible for sending data to the client
 - Usually in MVC3, renders an HTML template using data retrieved by a Model and/or Controller
 - In our case, returns formatted data
 - JsonResult
 - ContentResult
 - our own custom NegotiatedResult

ASP.NET MVC3 - HttpClient

- **HttpClient is part of .NET 4.5, and provides a cleaner interface to make HTTP calls compared to WebClient or WebHttpRequest**
 - Uses Async methods that asynchronously make network calls
 - Use the new async/await keywords in .NET
 - Allows easy, one-line access to GET, POST, PUT, and DELETE requests
- **Json.NET is an open source library that provides one-line access to JSON serialization and deserialization methods**
 - Open source library (MIT license) that now ships with newer versions of ASP.NET
 - Installed via NuGet