

10 Ways to Build Web Services in .NET FubuMVC

Chad McCallum
@ChadEmm



Module Outline

- **Introduction to FubuMVC**
- **Installing and Configuring FubuMVC**
- **Creating our first Endpoint**
- **An Endpoint with a route parameter**
- **Endpoints using JSON requests**
- **Endpoint with Delete Verb**
- **Creating and using our own Behavior**
- **Receiving data using the Spring.NET REST Client**
- **Sending data using the Spring.NET REST Client**
- **Review**

FubuMVC

- **A MVC framework built on top of the ASP.NET library**
- **A strong focus on “pluggability” and conventional configuration**
- **Can easily replace, override, extend, and customize most components of FubuMVC**
- **Uses user-defined convention over configuration to avoid unnecessary configuration**

FubuMVC

- **Uses “behaviors” to process input and return output**
 - Every step in the process is a behavior that can be reused between requests
 - Route -> Action -> Output
 - Behavior chain can be modified by developers
- **One Model In, One Model Out**
 - Controller code is only concerned with processing the data, not retrieving it or rendering the output

Review

- **Introduction to FubuMVC**
- **Installing and Configuring FubuMVC**
- **Creating our first Endpoint**
- **An Endpoint with a route parameter**
- **Endpoints using JSON requests**
- **Endpoint with Delete Verb**
- **Creating and using our own Behavior**
- **Receiving data using the Spring.NET REST Client**
- **Sending data using the Spring.NET REST Client**

FubuMVC – Convention over Configuration

- **Controllers are automatically registered based on Class name (*Endpoints)**
- **Routes are automatically mapped based on convention**
 - Verb_Route_Route_Variable(InputModel model)
 - Get_Puns_PunID(PunIDRequest request) results in
GET /Puns/{PunID}

FubuMVC – Behaviors

- **Behaviors are reusable components of a request pipeline**
 - Routing, Actions, Content Negotiation & Output
- **Each request is served by a Behavior Chain**
 - Routing -> Endpoints Method -> Content Negotiation & Output
- **Each Behavior can continue or stop the chain**
- **New Behaviors can be created and inserted at any point in the Behavior Graph**

Spring.NET REST Client

- **RestTemplate wraps the root URL and functionality of the client**
- **MessageConverters handle serialization and deserialization of objects to and from the request and response**
- **.GetForObject<Type>("/route") automatically deserializes the response data into the specified type**
- **Can use route parameters by specifying a dictionary or ordered list of parameters**
- **Passing an object into Put or Post will serialize and send it with the request**