Introduction to ASP.NET Web API Services and OData

Brian Noyes
CTO, Solliance (www.solliance.net)
@briannoyes

pluralsight
hardcore developer training

Outline

- ASP.NET Web API overview
- ASP.NET Web API pipeline and configuration
- REST fundamentals
- Defining CRUD services with Web API
- OData protocol overview

ASP.NET Web API Overview

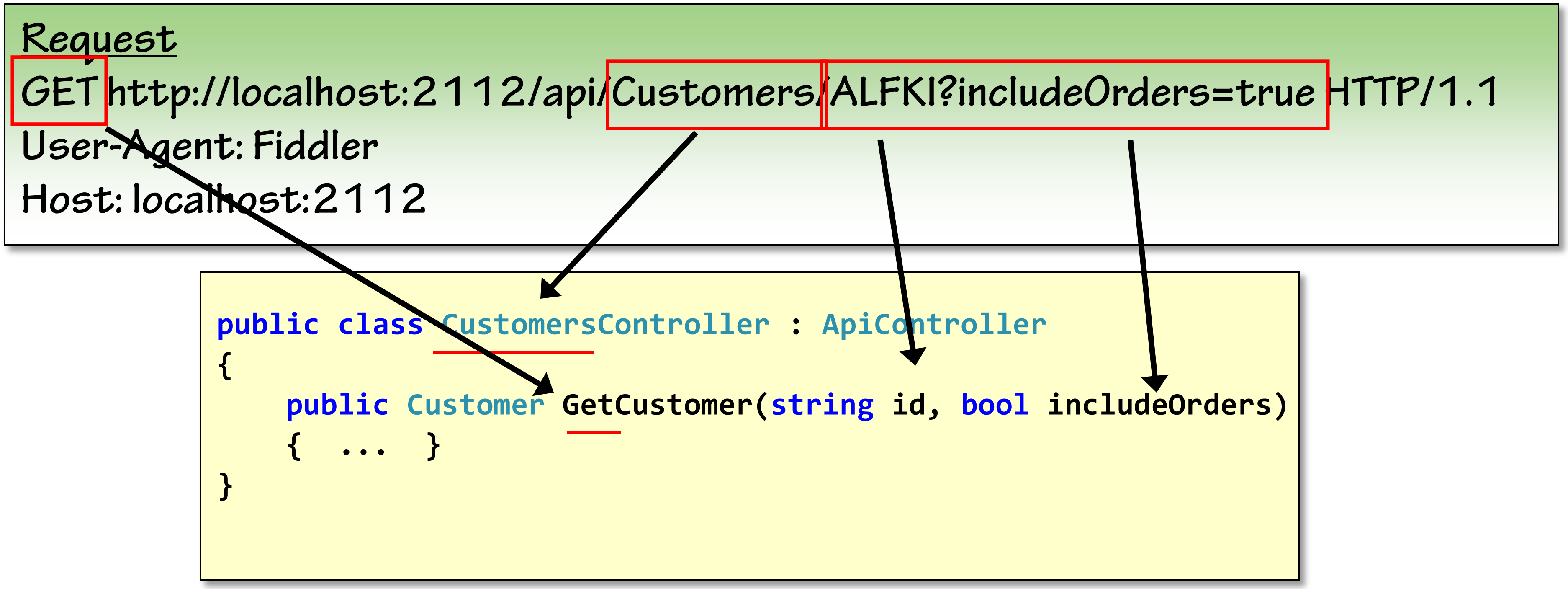
- **New platform for building HTTP web services (Web APIs)**
- **Built on top of ASP.NET MVC 4 framework**
  - Released with .NET 4.5
  - Compatible with .NET 4.0
- **Makes it easy to build services for consumption from multi-platform clients**
  - Simple RPC services
  - CRUD services
  - REST services
  - OData services

ASP.NET Web API Overview

- **Services are Controllers**
  - ApiController class
- **Leverages MVC features**
  - Routing
  - Model binding
  - Action filters

ASP.NET Web API Overview

- **Convention over configuration**
  - Maps URIs to controllers
  - Maps HTTP verbs to methods / actions
  - Maps URI / query string parameters to method parameters



```
Request
GET http://localhost:2112/api/Customers/ALFKI?includeOrders=true HTTP/1.1
User-Agent: Fiddler
Host: localhost:2112
```

```
public class CustomersController : ApiController
{
    public Customer GetCustomer(string id, bool includeOrders)
    {  ...  }
}
```

ASP.NET Web API Overview

- **Content negotiation**
  - Based off HTTP Accept / Content-Type headers
  - JSON / XML formatters out of the box
  - OData formatter through NuGet
  - Can plug in custom formatters

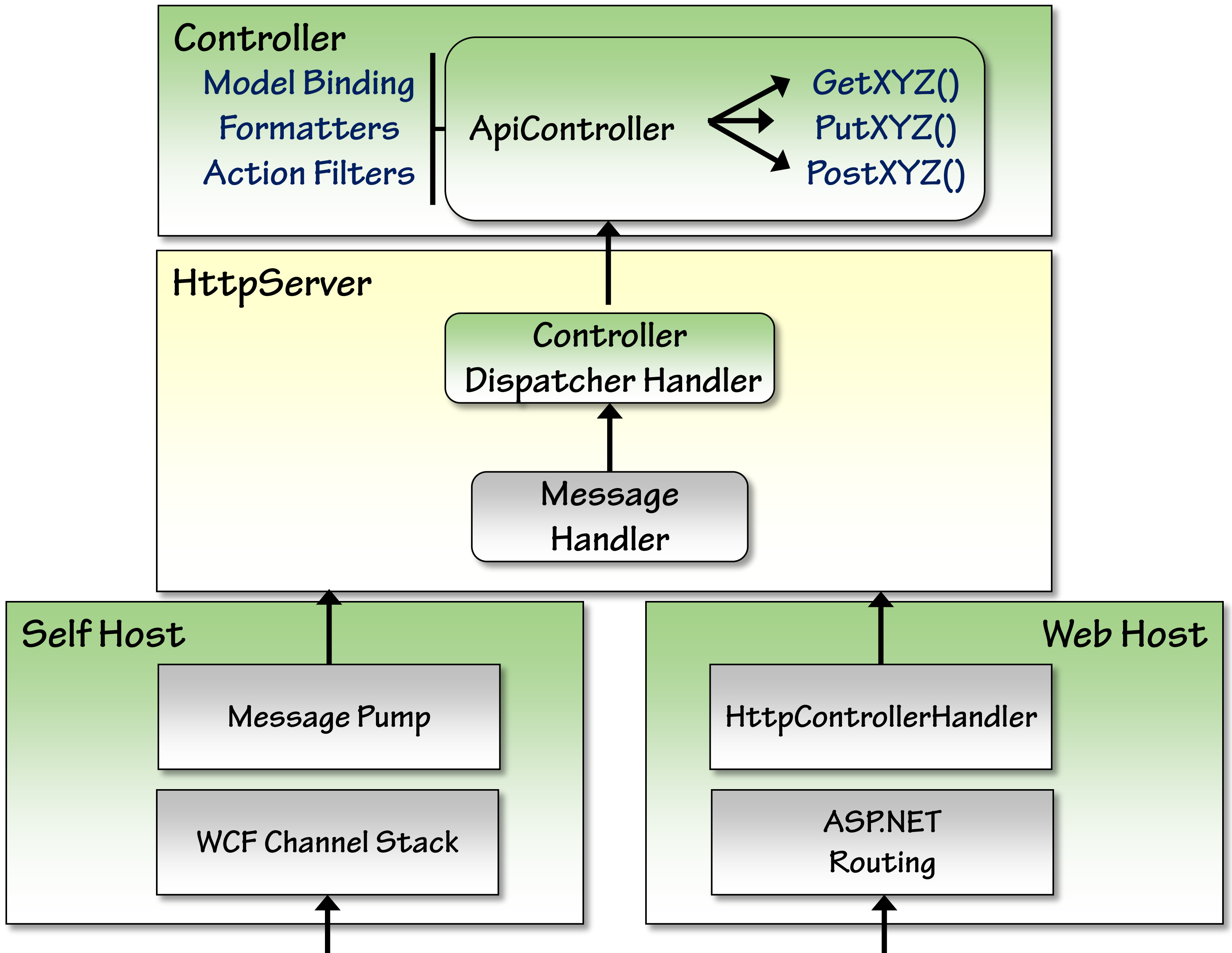
<u>Request</u>

```
GET http://localhost:2112/api/Customers/ALFKI HTTP/1.1
User-Agent: Fiddler
Host: localhost:2112
Accept: application/json
```

<u>Response</u>

```
HTTP/1.1 200 OK ...
Content-Type: application/json; charset=utf-8
Content-Length: 206
{"CustomerID":"ALFKI","CompanyName":"Alfreds Futterkiste"}
```

ASP.NET Web API Pipeline



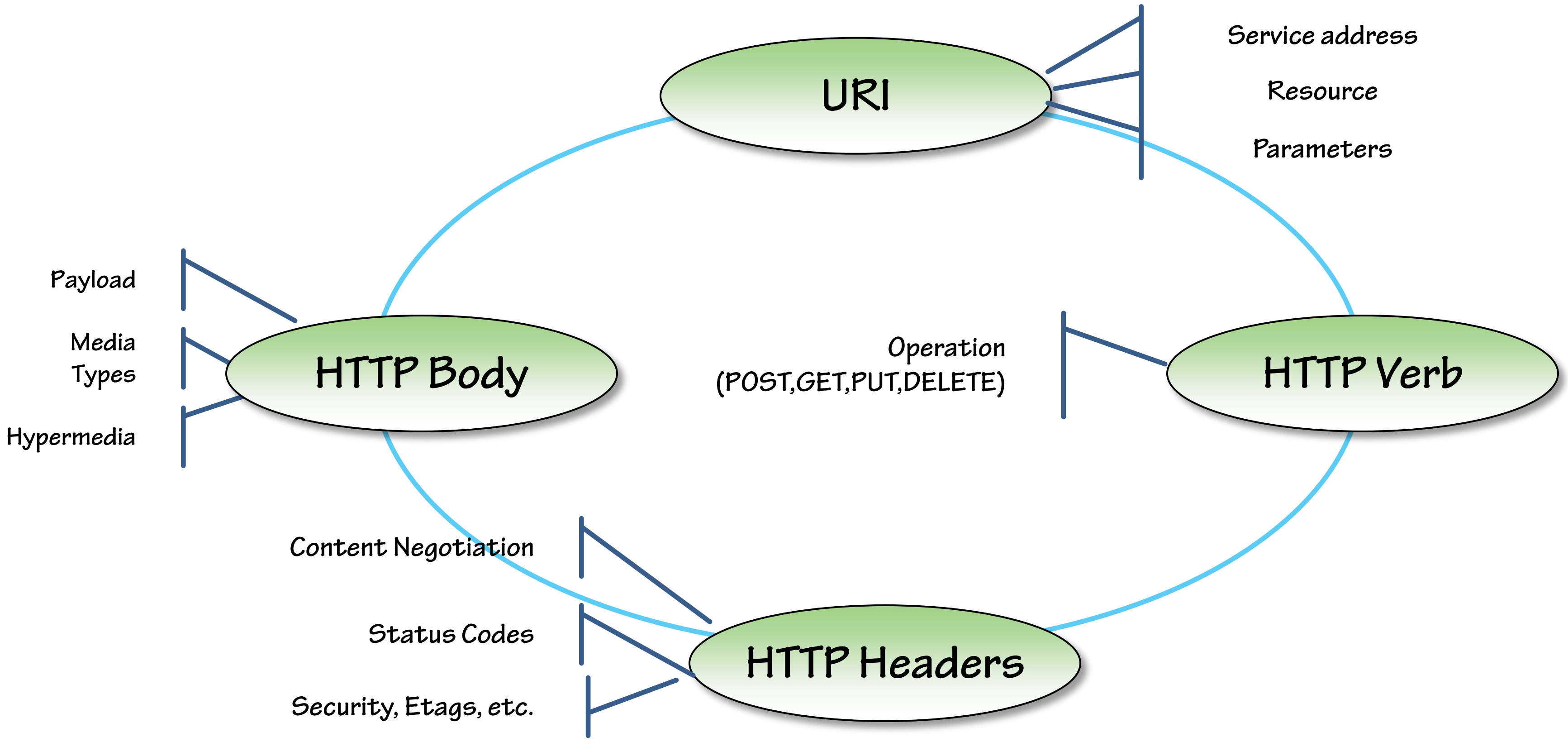
ASP.NET Web API Configuration

- No config file settings needed
- HttpConfiguration class
  - Associated with the ASP.NET web application instance
  - Accessible from Global.asax code behind
    - Calls WebApiConfig.Register
  - Defaults are good enough for basic Web APIs
  - Can plug in formatters, filters, message handlers and other custom extensibility objects through this class

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

REST

- REpresentational State Transfer
- REST is an architectural style, SOAP is a protocol
  - Based on Ph.D. thesis: Roy Fielding
- REST fully embraces HTTP



Simple CRUD Services with Web API

- Data collections are resources
- Expose a data collection as a Web API controller
- Map HTTP verbs onto CRUD Actions and controller methods

HTTP Verb	CRUD Action	Controller Method
POST	Create	PostCustomer
GET	Retrieve	GetCustomers, GetCustomer(id)
PUT	Update	PutCustomer(id)
DELETE	Delete	DeleteCustomer(id)

OData

- **Industry open standard protocol (www.odata.org)**
  - Led by Microsoft
- **Expose data over HTTP services for query or update**
- **REST-based services**
  - Hypermedia links for related resources and actions
  - Includes metadata for client code generation
- **Current version: 3.0**
- **OData Query Syntax: URL syntax for expressing queries**
  - Data service URI
  - Entity set name
  - Navigation property
  - Operators and functions
- **OData Formatting: ATOM Publishing Protocol or JSON formatting**
  - application/atom+xml or application/json
  - JSON Verbose vs JSON Light
- **HTTP Verbs to express operation**
  - GET, POST, PUT, DELETE, PATCH, MERGE

Summary

- ASP.NET Web API is a new platform for HTTP services
- ASP.NET Web API emphasizes convention over configuration
- The Web API pipeline exposes extensibility points with message handlers, action filters, and formatters
- REST is an architectural style that fully leverages the HTTP protocol and emphasizes resources and representations instead of remote procedure calls
- Defining a CRUD service involves exposing GET, POST, PUT, and DELETE methods from a data collection-centric resource controller
- The OData protocol defines a standardized way to expose CRUD oriented services in a RESTful way