ExcelShore® Operational Excellence

Cybage® Delivering Value. Scientifically.

Introduction to Web API

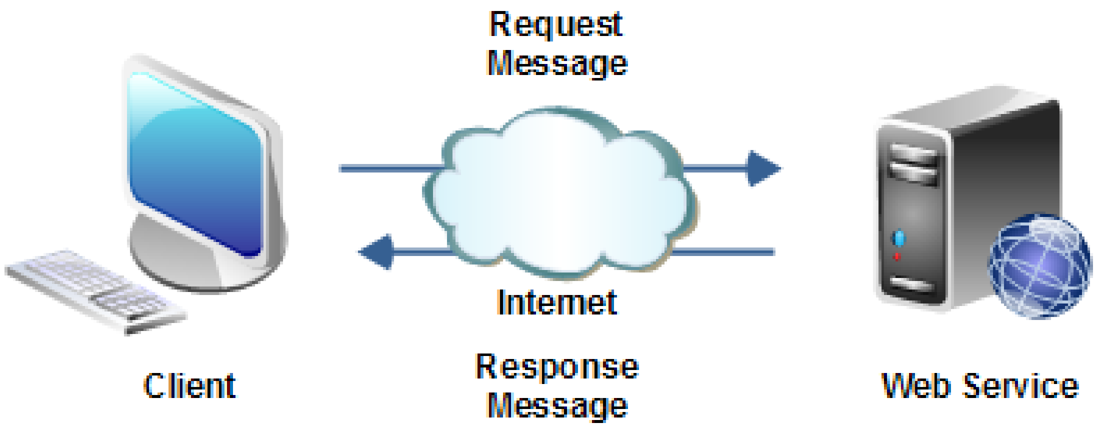
Authored and Presented by: Sushant Banerjee

Agenda

- Introduction
- HTTP Methods
- Routing
- Content Negotiation
- Controlling Response
- .NET and JavaScript Client

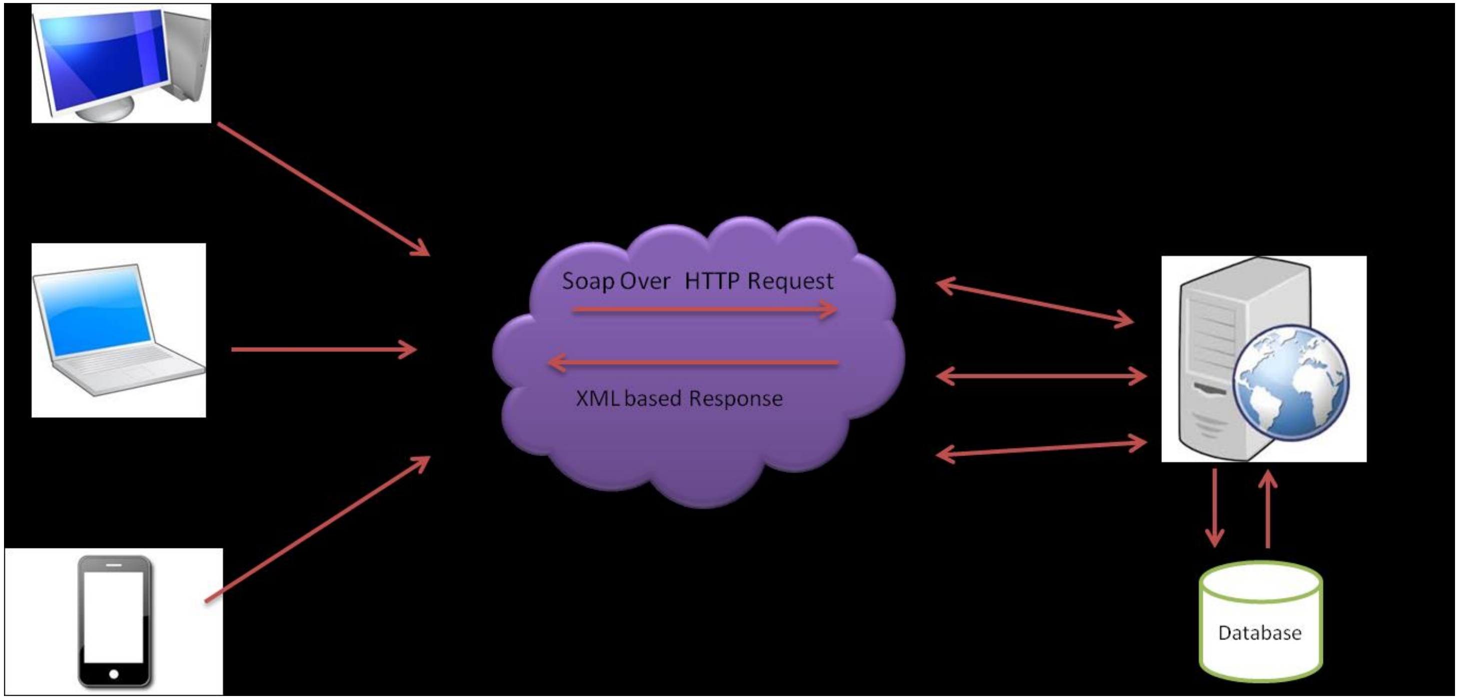
What is a Service

- A service exposes some set of operations
- Accessible through web
- Ideally accessible from any platform.



Why Use Services

- Building reusable component
- Interoperability
- Standard protocols

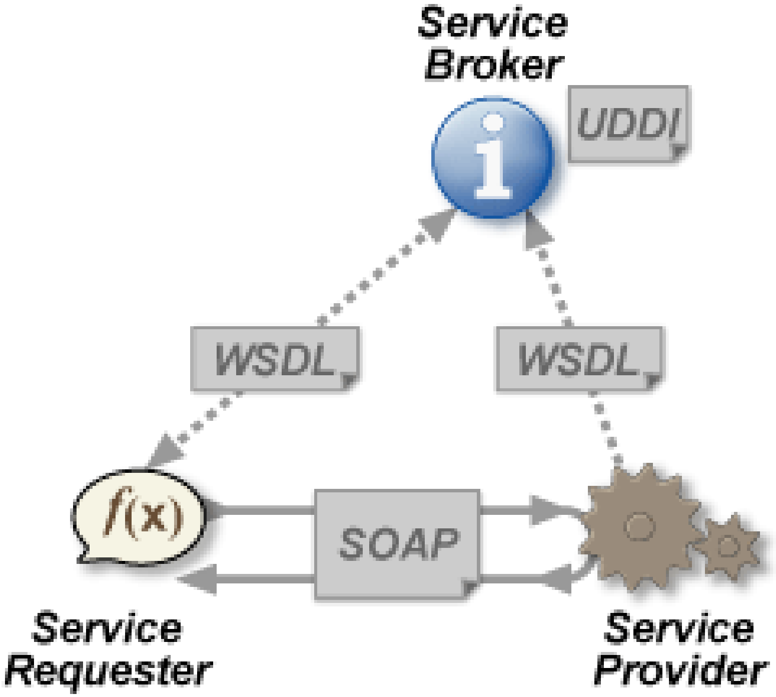


Evolution of Web Service in .NET Framework

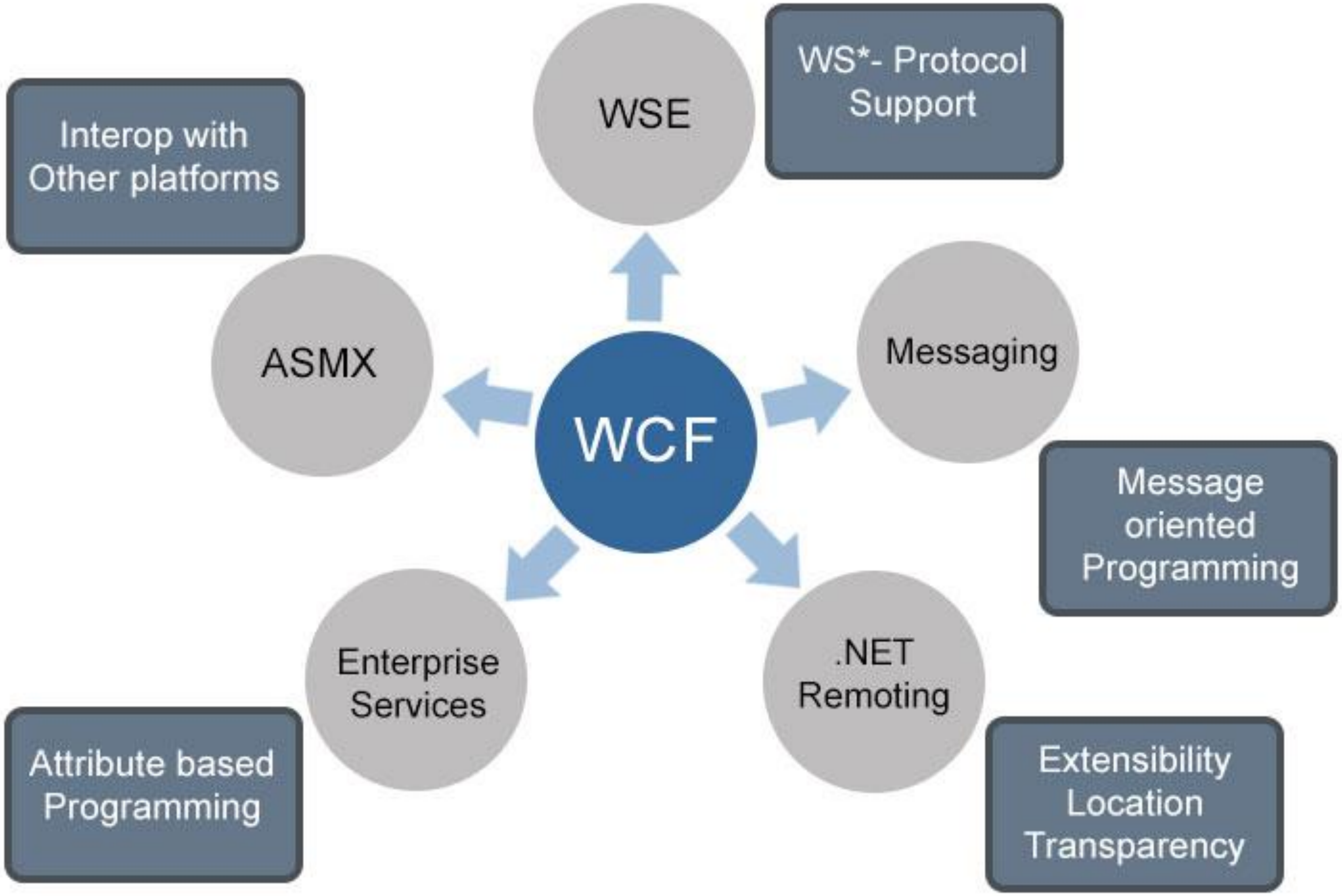
- ASP.NET Web Service(ASMX)
- Windows Communication Foundation(WCF)
- ASP.NET Web API

ASMX Service

- Simple Object Access Protocol(SOAP)
- Web Service Description Language(WSDL)
- Universal Description, Discovery and Integration(UDDI)



Windows Communication Foundation

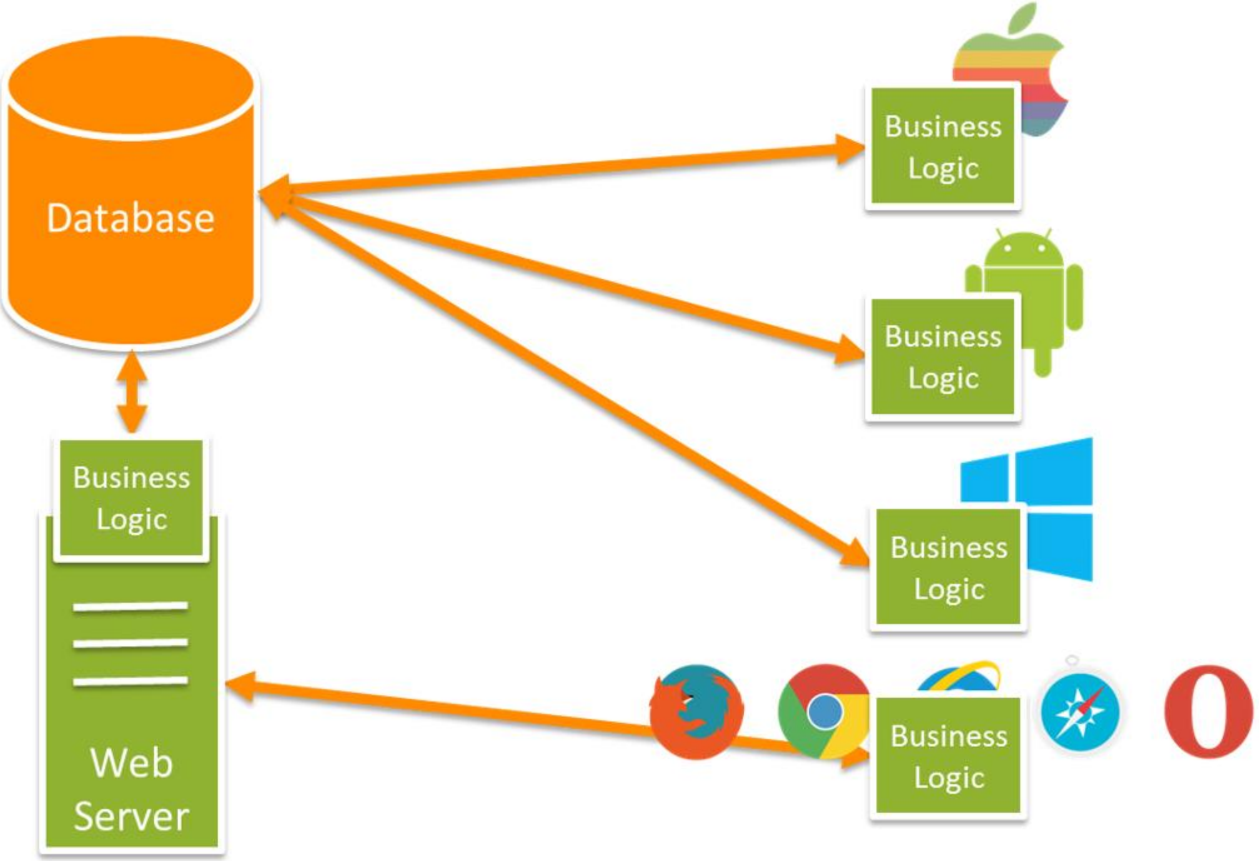

WCF
WSE
WS*- Protocol Support
Messaging
Message oriented Programming
.NET Remoting
Extensibility Location Transparency
Enterprise Services
Attribute based Programming
ASMX
Interop with Other platforms

What is Web API

- A framework to build HTTP based services
- Can reach to a broad range of clients
- Supports features similar to MVC
- Ideal for devices with limited bandwidth
- Can be hosted within application or on IIS
- A light weight and open source architecture

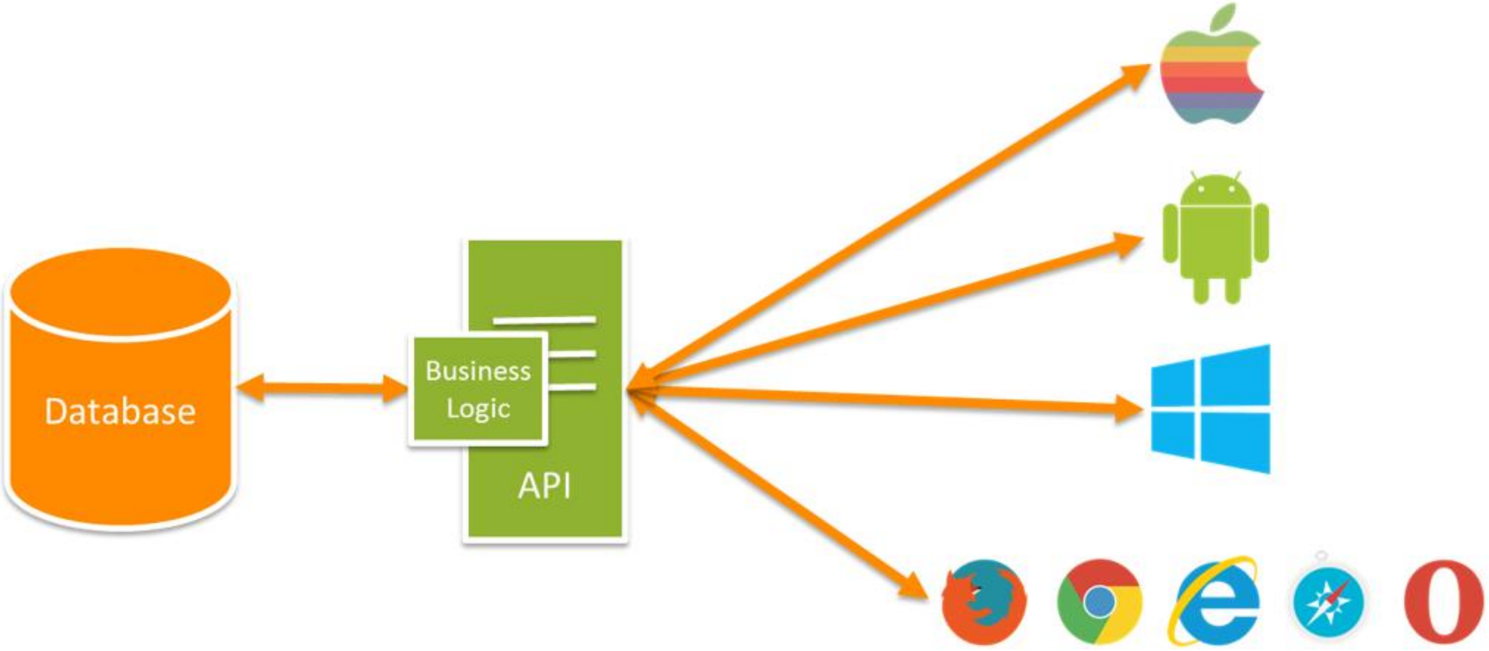
Real World Problem

- Business logic is everywhere
- Maintaining duplicate code is difficult.

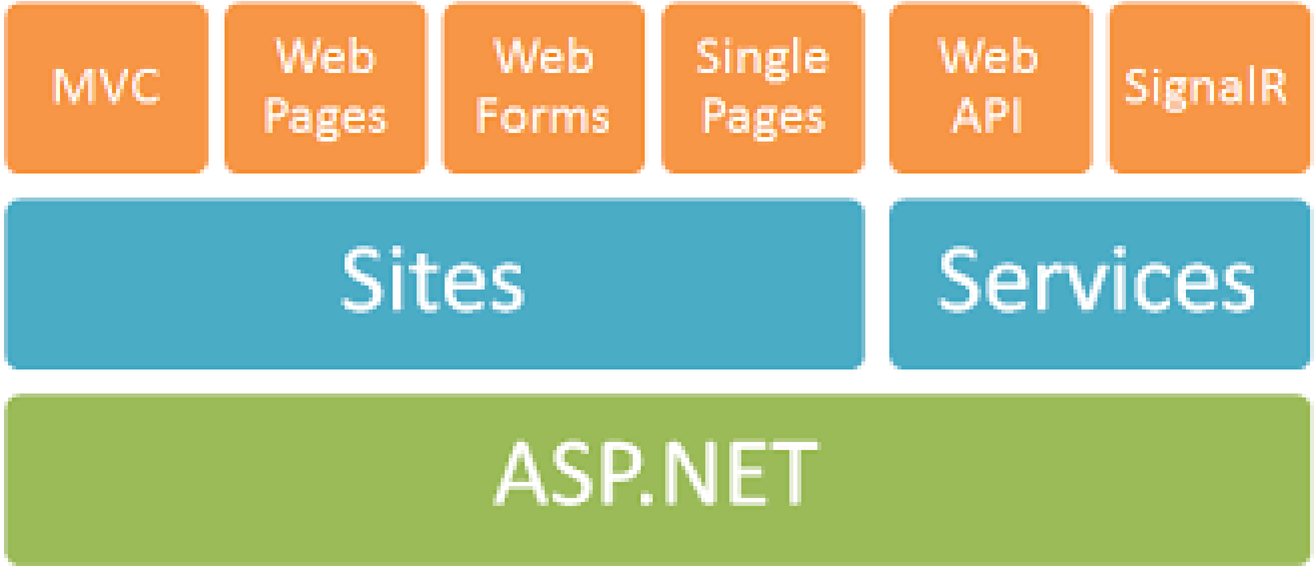


Why Web API – The solution

- Business logic kept centrally in a single location
- If need changes, change in one place.



Web API in ASP.NET Framework



WCF or Web API

- When to use WCF
  - To implement Simplex, MSMQ, Duplex communications
  - To use TCP, Named Pipes, HTTP protocols

- When to use ASP.NET Web API
  - To develop resource oriented services over HTTP
  - To reach broad range of clients – browser, smart phones, tablets.

Demo – Hello Web Api

- Using Developer Tools / F12

What is REST

Representational State Transfer

- A software architectural style
- Defined by Roy Thomas Fielding in 2000
- It describes six constraints
  - Client-server
  - Stateless
  - Cacheable
  - Layered system
  - Code on demand
  - Uniform interface
- A RESTful service should comply with above constraints

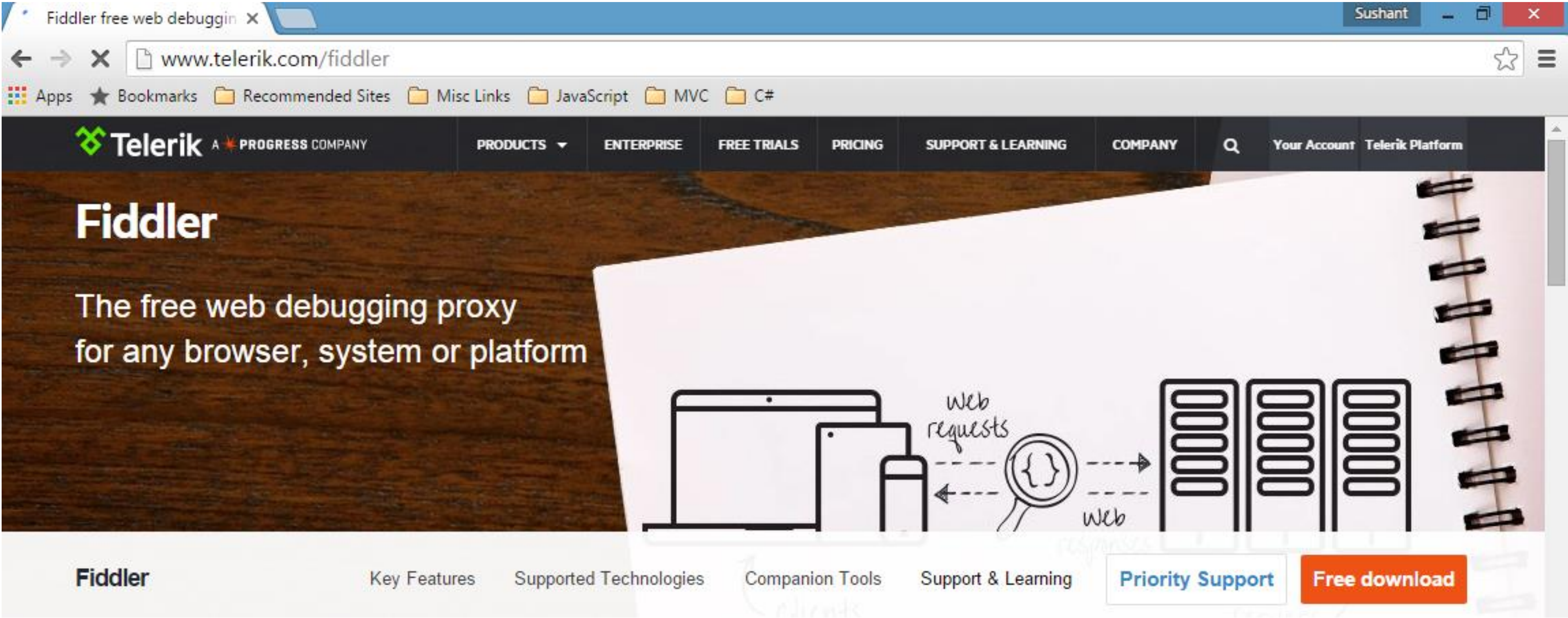
A RESTful Service

- Typically communicate over HTTP
- Considers data and functionalities as resources
- These resources are accessible using URIs
- Communicates using HTTP verbs
  - GET
  - POST
  - PUT
  - DELETE

HTTP Methods

- GET
  - Retrieves the representation of the resource at a specified URI
  - No side effects on the server
- POST
  - Creates a new resource
  - Response message returns with URI of newly created object
- PUT
  - Updates a resource at a specified URI
- DELETE
  - Deletes a resource at a specified URI.

Using Fiddler



Fiddler free web debuggin

www.telerik.com/fiddler

Fiddler

The free web debugging proxy for any browser, system or platform

Demo

Request Processing in Web API

- To find controller Web API framework
  - Adds “Controller” suffix to {controller} variable
  - Looks for a class inherits from ApiController
  - An ApiController implements IHttpController

- To find action method
  - Web API looks for action methods begins with HTTP methods
  - The above convention applies to GET, POST, PUT and DELETE
  - Action parameters are mapped to {id} variable.

MIME

- Stands for Multipurpose Internet Mail Extensions
  - An internet standard for supporting format of email
  - Designed for SMTP but applicable to HTTP also
  - Content Type or Media Type indicates type of data
- A media type includes type/subtype
  - Application/json
  - Application/xml
  - Text/html
  - Image/png etc.
- Serialization depends on media type.

Media Type Formatter

- A media type formatter can
  - Read CLR objects from HTTP message body
  - Write CLR objects into HTTP message body
- Web API supports both
  - XML media type formatters
  - Json media type formatters
- Accept header of HTTP request
  - Client can specify either XML or Json
- A media formatter serializes the resource.

Model Binding and Validation

- Web API parameter binding
  - Simple type binding – gets value from URI
  - Complex type binding – gets value from message body
- FromUri attribute
  - Forces reading type from the URI
- FromBody attribute
  - Forces reading type from the body
- Model validation using Data Annotations – MVC style

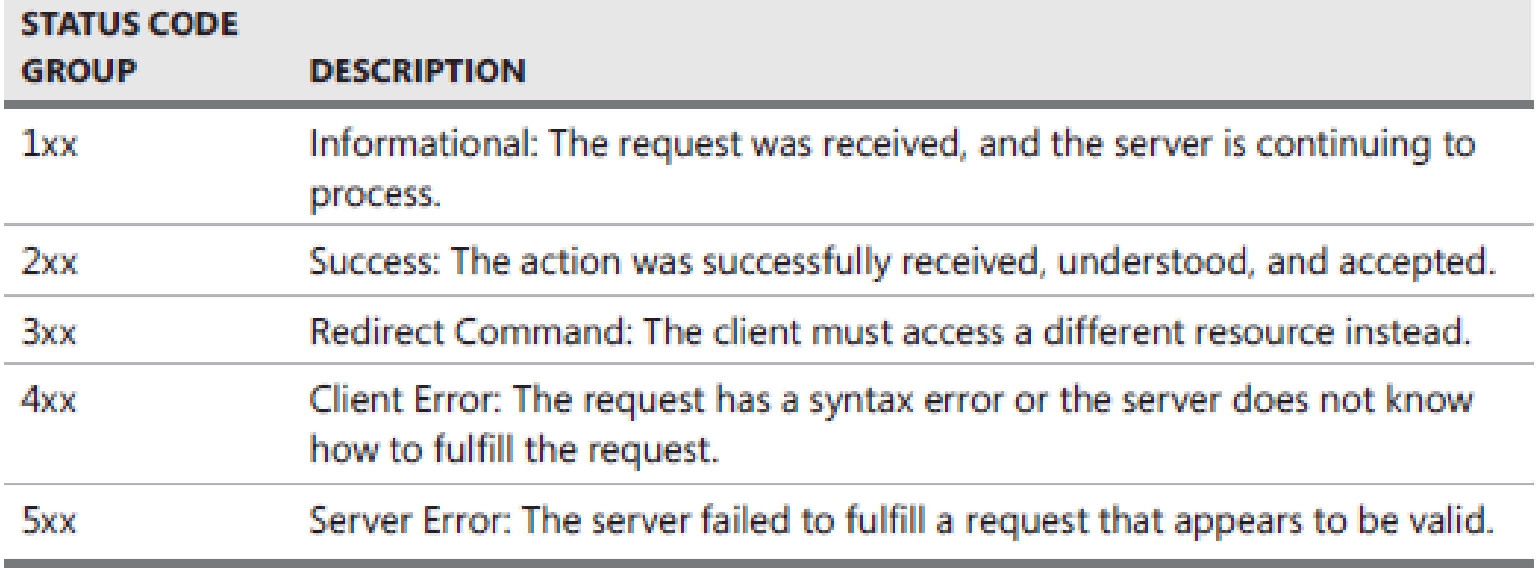
Content Negotiation

- A process of selecting best representation
- Server looks at the HTTP request headers
- Web API framework has built-in support for
  - JSON (JavaScript Object Notation)
  - XML
- You can add your own custom media formatter.

Demo

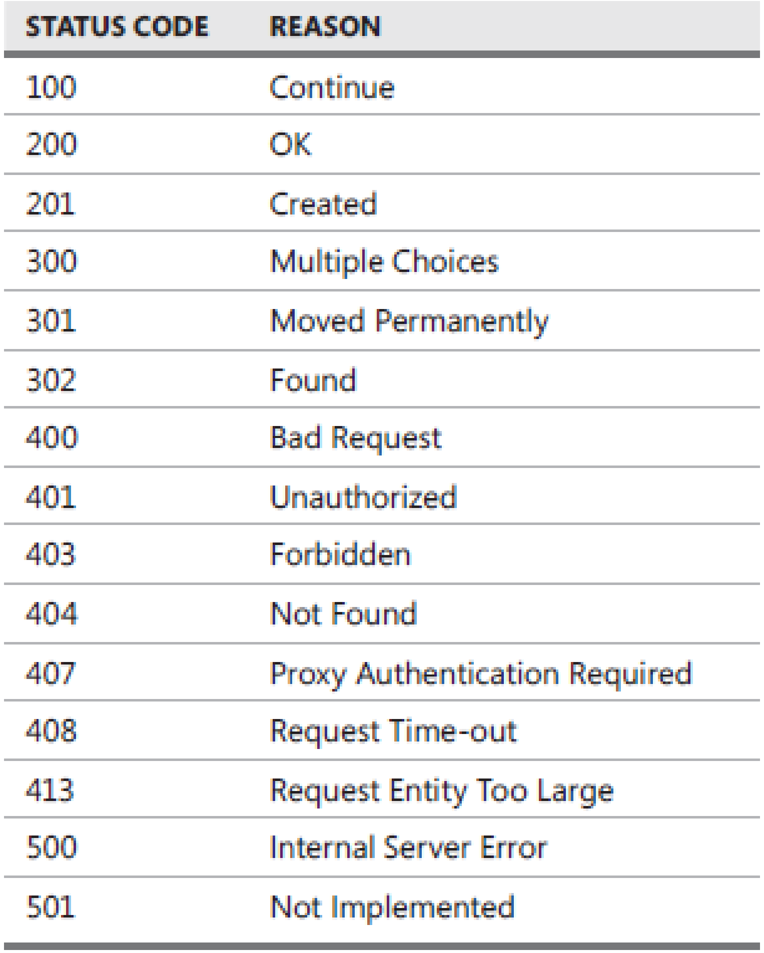
HTTP Status Code Groups

- A status code group refers related status codes.
- An HTTP status code describes HTTP response
- It also describes the reason for the response



STATUS CODE GROUP	DESCRIPTION
1xx	Informational: The request was received, and the server is continuing to process.
2xx	Success: The action was successfully received, understood, and accepted.
3xx	Redirect Command: The client must access a different resource instead.
4xx	Client Error: The request has a syntax error or the server does not know how to fulfill the request.
5xx	Server Error: The server failed to fulfill a request that appears to be valid.

Common HTTP Status Codes



STATUS CODE	REASON
100	Continue
200	OK
201	Created
300	Multiple Choices
301	Moved Permanently
302	Found
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
407	Proxy Authentication Required
408	Request Time-out
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented

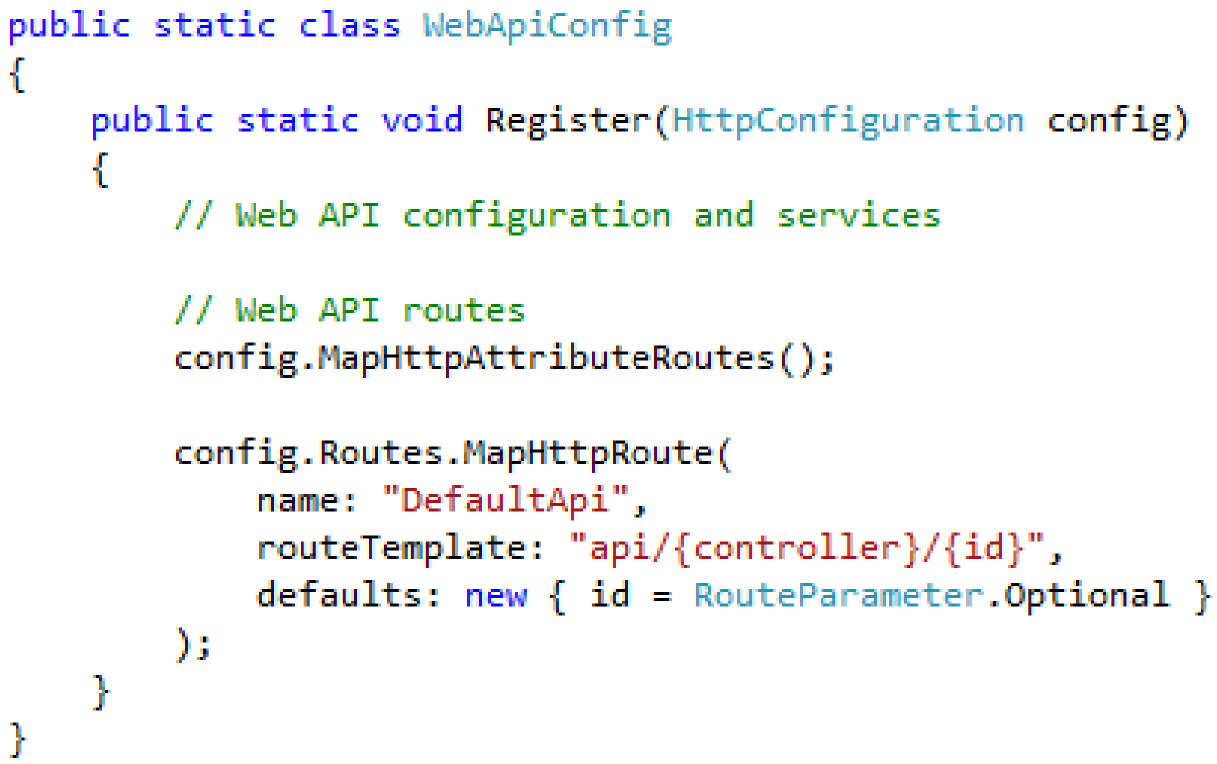
Web API Action’s Return Types

- Void
- HttpResponseMessage
- IHttpActionResult
- Other Types

Demo

Routing in Web API

- MVC uses URI path to select action
- Web API uses HTTP methods to select action
- Can be modified to use MVC style routing



```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        // Web API configuration and services

        // Web API routes
        config.MapHttpAttributeRoutes();

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

Overriding Convention

- By convention Web API maps HTTP verbs to action methods
- Use attributes to override the convention
  - HttpGet
  - HttpPost
  - HttpPut
  - HttpDelete
- Routing by action name – MVC style
- Using ActionName attribute
- Using NonAction attribute.

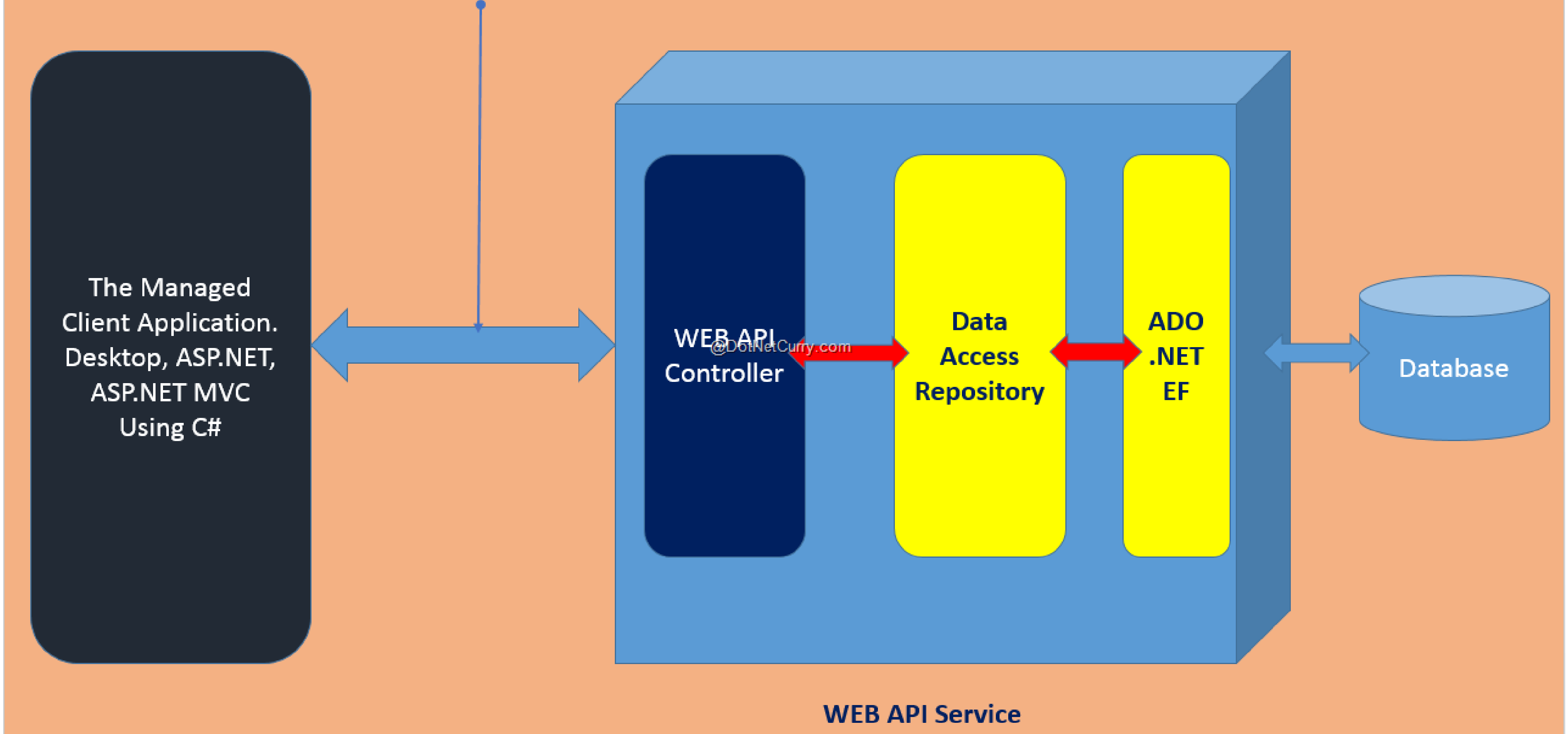
Attribute Routing

- Introduced with Web API 2
- Doesn’t replace convention based routing
- Uses attributes to define routes
- Gives more control over the URIs.

Demo

Web API Client

HttpClient class is used for Making Calls to WEB API and perform Http Operations.



Web API Clients

- For HTTP client add the .dlls
  - System.Net
  - System.Net.Http
  - System.Net.Http.Formatting
- Calling Api From Ajax Client
  - Using JQuery

Demo

Bibliography, Important Links

- http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api
- http://www.asp.net/web-api/overview/older-versions/creating-a-web-api-that-supports-crud-operations
- http://www.asp.net/web-api/overview/advanced/calling-a-web-api-from-a-net-client

Any Questions?

Email : sushantba@cybage.com
Extn : 7221

Thank you!