WEB API

Week 11 Lecture 01

What’s Web API

- ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices.
- ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework.

HTTP Verbs

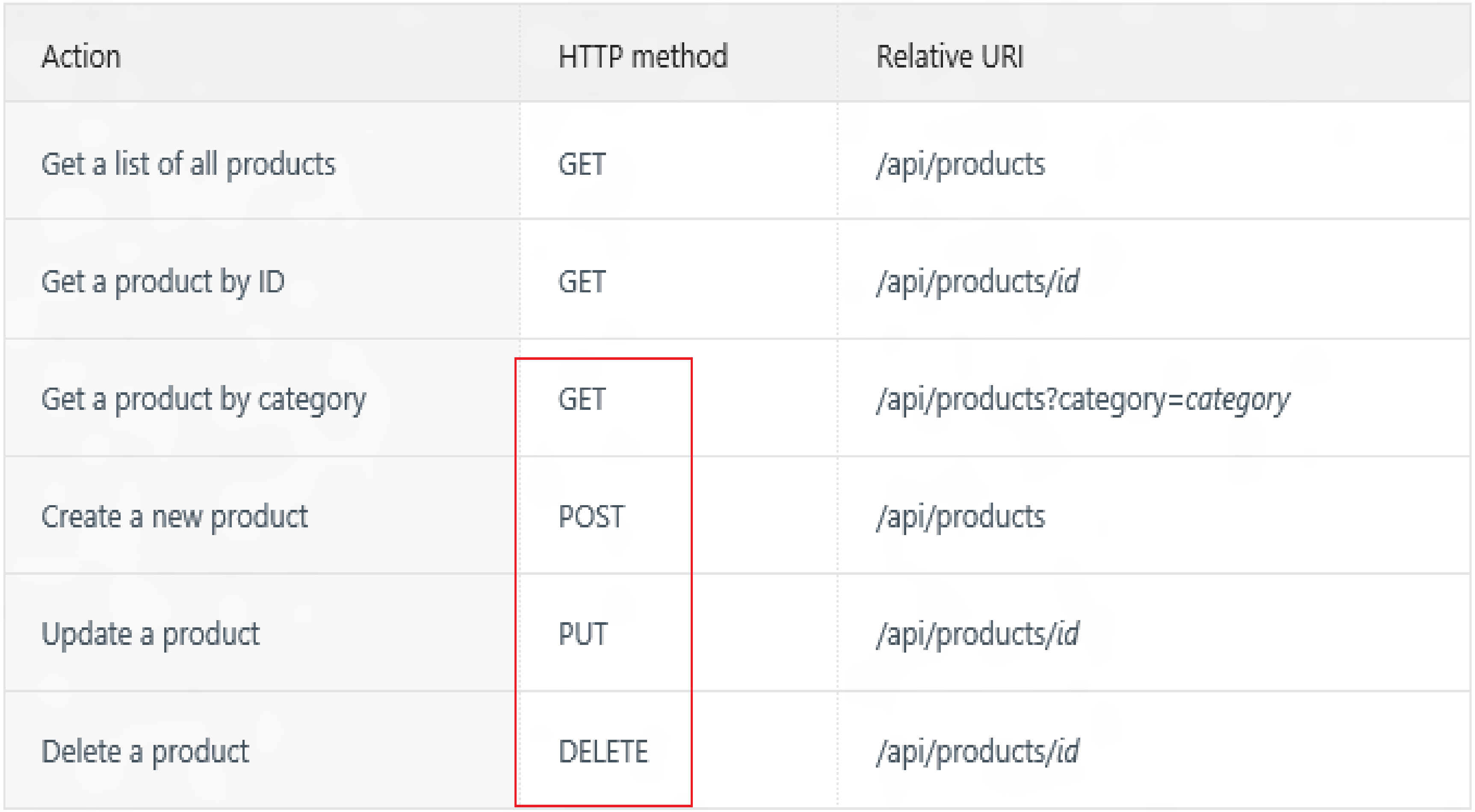
- GET
  - The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.
  - Example : GET http://www.example.com/customers/12345

- POST
  - The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server
  - Example : POST http://www.example.com/customers

- PUT
  - The PUT method replaces all current representations of the target resource with the request payload.
  - Example: PUT http://www.example.com/customers/12345

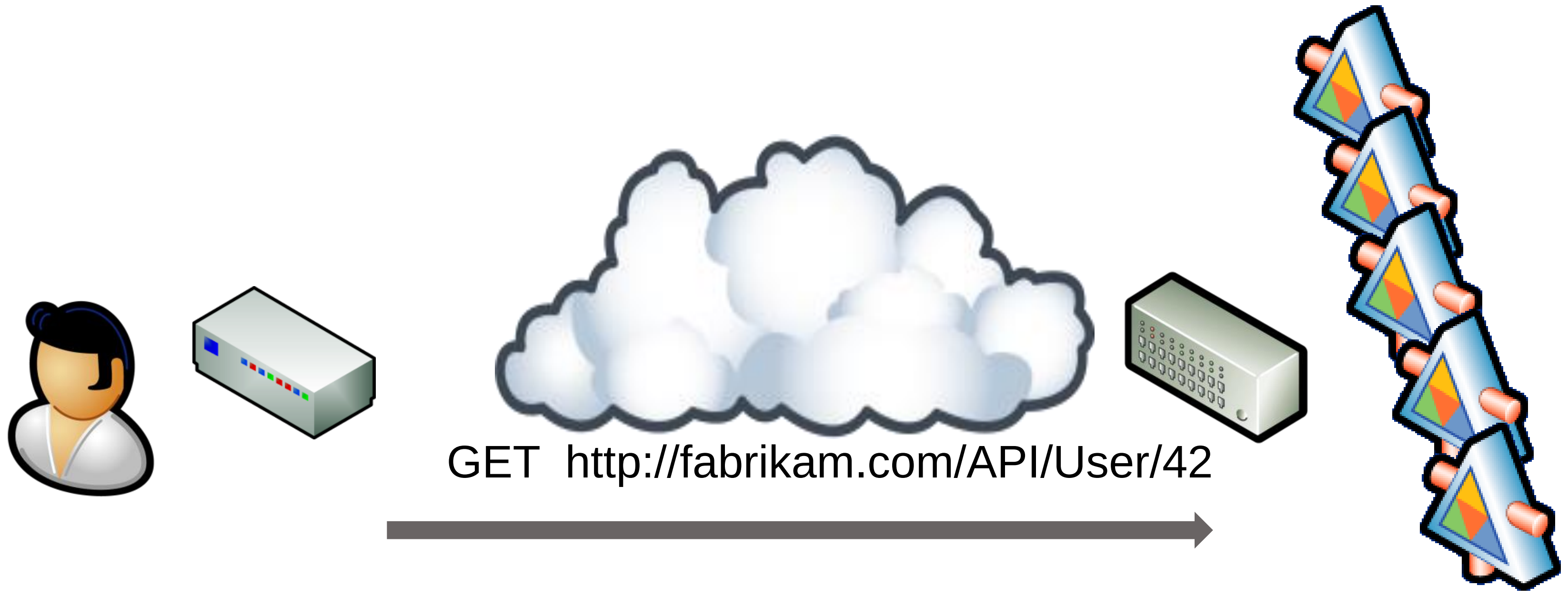
- DELETE
  - The DELETE method deletes the specified resource.
  - Example: DELETE http://www.example.com/customers/12345

HTTP method



Action	HTTP method	Relative URI
Get a list of all products	GET	/api/products
Get a product by ID	GET	/api/products/*id*
Get a product by category	GET	/api/products?category=*category*
Create a new product	POST	/api/products
Update a product	PUT	/api/products/*id*
Delete a product	DELETE	/api/products/*id*

Scalability



Media types & Media Formatters

- Built-in support for:
  - XML
  - JSON
  - form-urlencoded data
- Can be extended with custom Media Formatters

Data Return Formats

- Web API can return data in JSON or XML formats
- Web API uses the media formatter to:
  - Format or serialize the information that a Web API REST service returns
  - Control the media type in the HTTP header
  - Format all content that the server renders to client systems
- Media formatter classes inherit from the **MediaTypeFormatter** class and the **BufferedMediaTypeFormatter** class

Routing in Web API

Characteristics of routing in Web API:

- You can use API controller names and a naming convention for actions to route Web API requests
- Alternatively you can use the following attributes to control the mapping of HTTP requests (HTTP Verb+URL) to actions in the controller:
  - The **HttpGet**, **HttpPut**, **HttpPost**, or **HttpDelete** attributes
  - The **AcceptVerbs** attribute
  - The **ActionName** attribute

Should I use WCF or ASP.NET Web API

- Use WCF
- ❑ If you are limited to .Net 3.5
- ❑ If you are exposing SOAP based services
- ❑ If you need to support multiple protocols
- ❑ If you need to support WS-* transaction
- ❑ If you need to achieve message level security

Use ASP.Net Web API

- ❑ If you need to reach wider and diverse cross platform clients / devices
- ❑ If you need to leverage the benefits of Http

Comparison

FEATURE	WCF	WEB API
Transport	HTTP/S, TCP, UDP, MSMQ, named pipes, custom	HTTP/S
Protocols	WS*	HTTP
Content Format	SOAP+XML	Any media type, format
Types	Data contracts (opt in)	CLR Types (opt out)
Service Interface	Service contracts	URL patterns, HTTP methods
State Management	Stateless with Per Call	Stateless
Caching	Handled by application	Built-in to HTTP Prefer application control
Hosting	IIS or self-host	IIS or self-host
Error Handling	Faults. behaviors	Exceptions, HTTP status codes filters
Security	Windows, Basic, Certificate WS*, Authorization header	Windows, Basic, Certificate Authorization header
Client	Proxy generation Shared libraries	IApiExplorer discovery Shared libraries