MODULE 01

INTRODUCTION TO WEB API 2

MODULE TOPICS

HTTP Protocol

History of Web Services

REST Services

ASP.NET MVC and Web API

Future of ASP.NET

HTTP PROTOCOL - REQUEST

1. Request is made with a URL (http://hostname:port/path/file?query=string)

2. Hostname resolves to an IP address via DNS

3. Protocol (http) or port determines port number

4. Connection is made to remote computer typically with TCP

5. Paths, files, query string and any data sent is processed by remote computer

HTTP PROTOCOL - RESPONSE

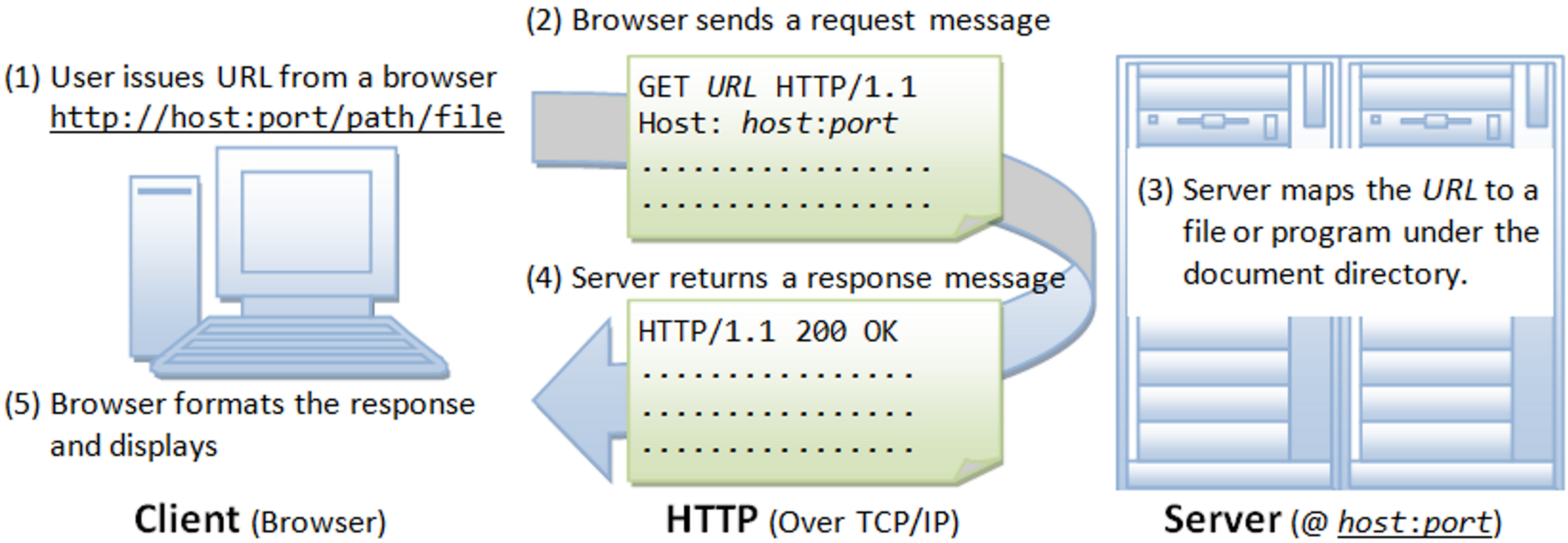
6. Remote computer generates either a response or an error

7. Successful responses will have an HTTP status code of 2xx

8. Other HTTP status codes indicate an error

9. This is repeated for each file (JavaScript, CSS, Image, etc) on the HTML page

HTTP PROTOCOL - REQUEST / RESPONSE



(1) User issues URL from a browser
http://host:port/path/file

(2) Browser sends a request message

```
GET URL HTTP/1.1
Host: host:port
.................
.................
```

(3) Server maps the *URL* to a file or program under the document directory.

(4) Server returns a response message

```
HTTP/1.1 200 OK
................
................
................
```

(5) Browser formats the response and displays

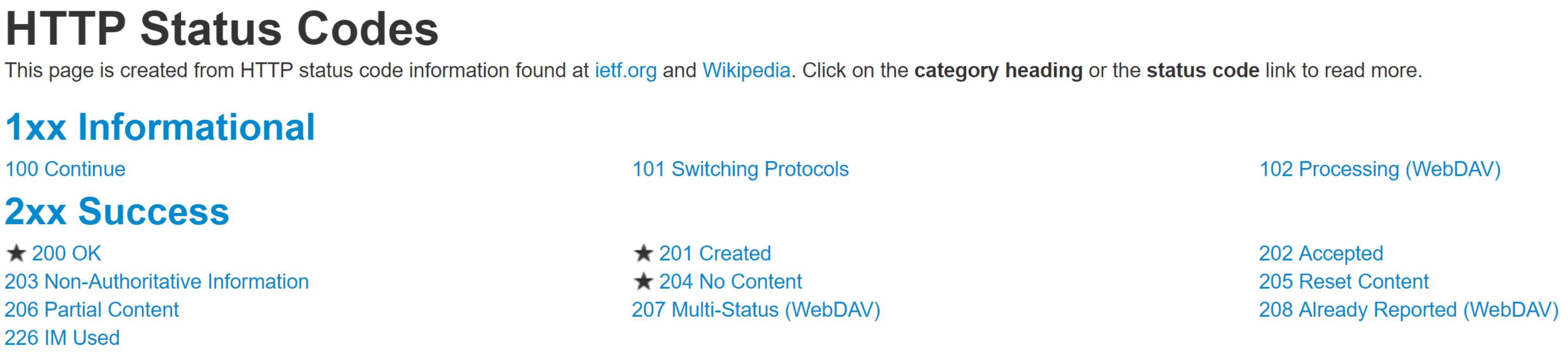
Client (Browser)

HTTP (Over TCP/IP)

Server (@ *host:port*)

HTTP PROTOCOL - STATUS CODES

```
https://www.restapitutorial.com/httpstatuscodes.html
```



HTTP Status Codes

This page is created from HTTP status code information found at ietf.org and Wikipedia. Click on the **category heading** or the **status code** link to read more.

1xx Informational

100 Continue
101 Switching Protocols
102 Processing (WebDAV)

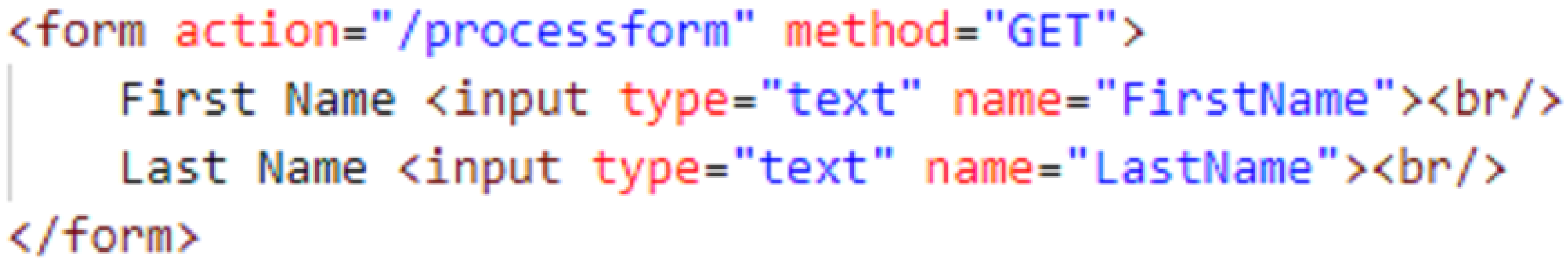
2xx Success

★ 200 OK
★ 201 Created
202 Accepted
203 Non-Authoritative Information
★ 204 No Content
205 Reset Content
206 Partial Content
207 Multi-Status (WebDAV)
208 Already Reported (WebDAV)
226 IM Used

HTTP PROTOCOL - STATELESS

HTTP Doesn't remember one connection to the next
Techniques such as cookies and session IDs are used to get around this

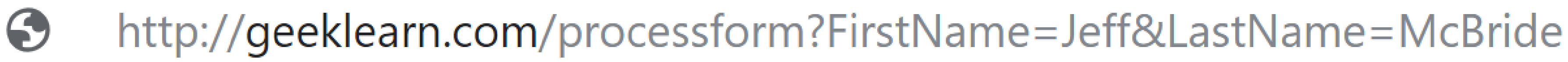
HTTP PROTOCOL - VERBS



```
<form action="/processform" method="GET">
    First Name <input type="text" name="FirstName"><br/>
    Last Name <input type="text" name="LastName"><br/>
</form>
```

HTML Developers are probably familar with setting a form's method to either GET or POST

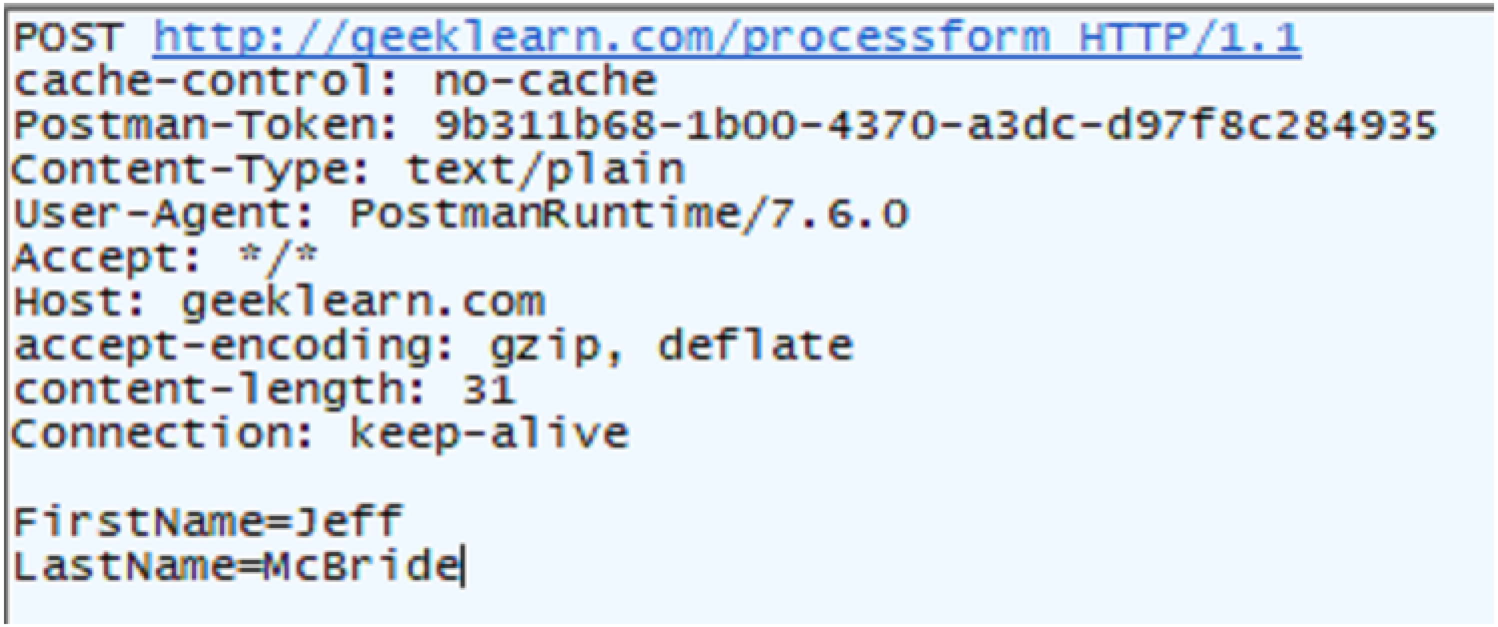
HTTP PROTOCOL - VERBS



http://geeklearn.com/processform?FirstName=Jeff&LastName=McBride

Get will send the form data to the server via the Query String

HTTP PROTOCOL - VERBS



```
POST http://geeklearn.com/processform HTTP/1.1
cache-control: no-cache
Postman-Token: 9b311b68-1b00-4370-a3dc-d97f8c284935
Content-Type: text/plain
User-Agent: PostmanRuntime/7.6.0
Accept: */*
Host: geeklearn.com
accept-encoding: gzip, deflate
content-length: 31
Connection: keep-alive

FirstName=Jeff
LastName=McBride
```

POST will send the form data to the server via a HTTP Host Header

HTTP PROTOCOL - VERBS

REST Services Typically Used Verbs

Verb	Functionality
Get	Read Data
Post	Create / Insert Data
Put	Update / Replace Data
Delete	Delete / Remove Data
Patch	Update / Replace data with changes only

HISTORY OF WEB SERVICES

PROPRIETARY SERVICES

- Use Proprietary transfer protocols such as
  RPC (Remote Procedure Call)
  RMI (Remote Method Invocation)
- Use a variety of different proprietary binary data formats

HISTORY OF WEB SERVICES

XML WEB SERVICES

- **HTTP (Hyper Text Transfer Protocol)**
  Ubiquitous data transfer protocol
- **XML (eXtensible Markup Language)**
  Ubiquitous data format

HISTORY OF WEB SERVICES

XML WEB SERVICES

- **SOAP (Simple Object Access Protocol)**
  A format for object serialize / deserialize
- **WSDL (Web Services Description Language)**
  A service contract for the service
- **UDDI (Universal Description, Discovery, and Integration)**
  A system for locating web services

REST SERVICES

ASP.NET MVC

- Model View Controller design pattern for Web App development
- Models represents the application data
- Views are the User Interface
- Controllers orchestrate the exchange of data between Views & Models

ASP.NET MVC

- Microsoft adopted it as an alternative to Web Forms
- Better control of the outputted HTML / JavaScript
- MVC evolved quickly
  - Version 1 released in 2009 and version 5 in 2013

ASP.NET WEB API

- Web API is Microsoft's REST Service implementation
- Traditionally bundled with MVC and shares most of its code base
- It is basically MVC without the Views
- WCF can also be used to develop services, but more complexity

FUTURE OF ASP.NET

FUTURE OF ASP.NET

- ASP.NET Core is the current evolution of ASP.NET
- Allows .NET web apps to be developed and deployed on Windows, Linux, and Mac
- This allows ASP.NET to work well in a DevOps development environment
- Container environments such as Docker and Kubernetes also work well with Core

ANY QUESTIONS?