

# HTTP Protocol

Handling Requests,  
Constructing Responses, HTTP/2

# HTTP

SoftUni Team  
Technical Trainers



SoftUni  
Foundation



Software University  
<http://softuni.bg>

# Table of Contents

1. Http Basics
2. URL
3. Http Request
4. Http Response
5. MIME and Media types
6. Web Server
7. Html Forms



Questions?



**sli.do**

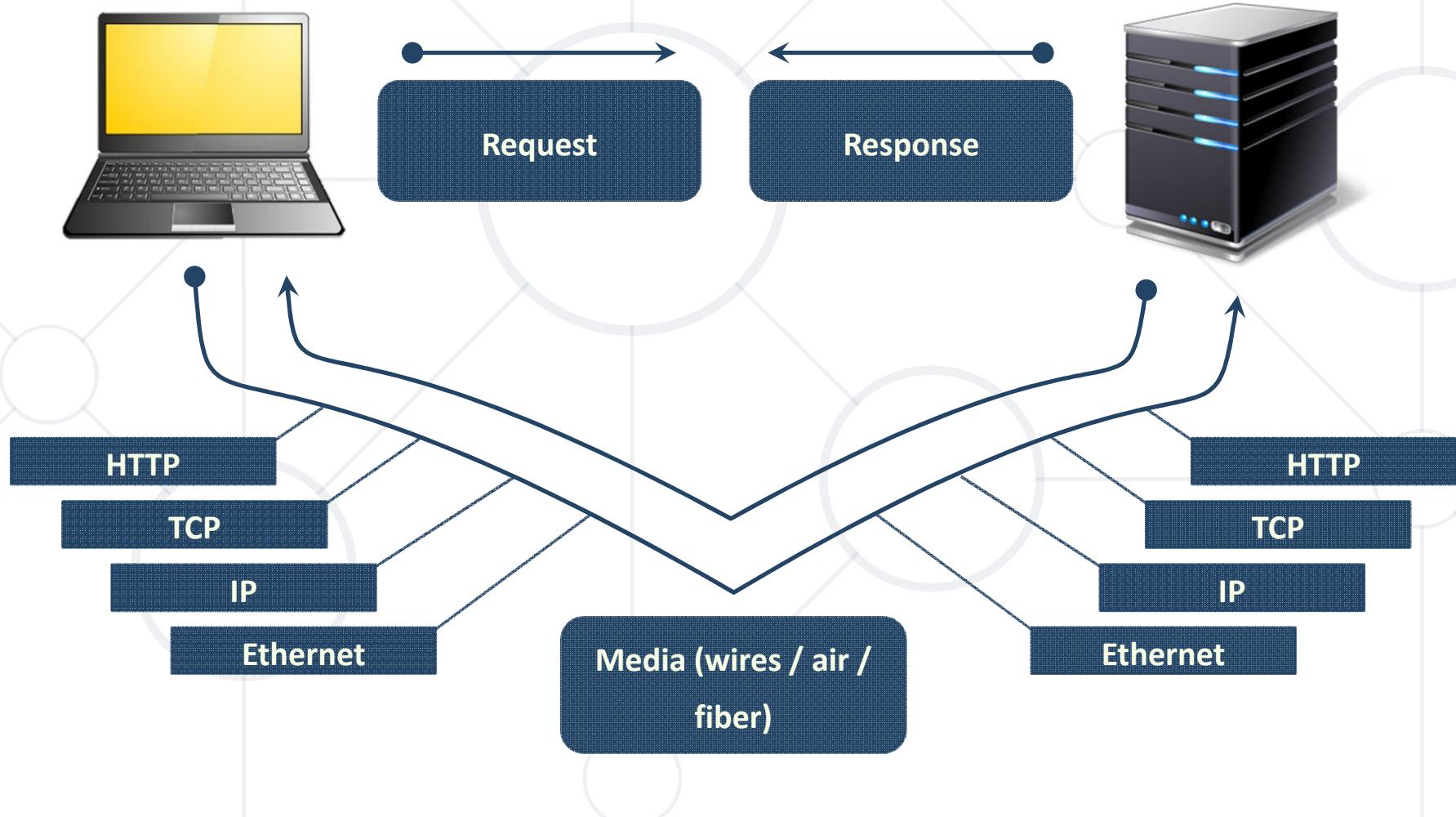
**#Java-Web**



# HTTP Basics

## Request and Responses

# Hyper Text Transfer Protocol



# HTTP Request Methods

- HTTP defines **methods** to indicate the desired action to be performed on the identified resource

Method	Description
GET	Retrieve / load a resource
POST	Create / store a resource
PUT	Update a resource
DELETE	Remove a resource

# HTTP Conversation: Example

- HTTP request:
- HTTP response:

```
GET /courses/javascript HTTP/1.1
Host: www.softuni.bg
User-Agent: Mozilla/5.0
<CRLF>
```

The empty line denotes the end of the request header

```
HTTP/1.1 200 OK
Date: Mon, 5 Jul 2010 13:09:03 GMT
Server: Microsoft-HTTPAPI/2.0
Last-Modified: Mon, 12 Jul 2014 15:33:23 GMT
Content-Length: 54
<CRLF>
<html><title>Hello</title>
Welcome to our site</html>
```

The empty line denotes the end of the response header



**URL**  
**Uniform Resource Locator**

# Uniform Resource Locator (URL)

`http://mysite.com:8080/demo/index.php?id=27&lang=en#lectures`



- URL is a formatted string, consisting of:
  - Protocol for communicating (**http, ftp, https...**) – HTTP in most cases
  - Host or IP address (**www.softuni.bg, gmail.com, 127.0.0.1, web**)
  - Port (the default port is **80**) – a number in range [0...65535]
  - Path (**/forum, /path/index.php**)
  - Query string (**?id=27&lang=en**)
  - Fragment (**#lectures**) – used on the client to navigate to some section

# URL Encoding

- URLs are encoded according RFC 1738:

- Safe URL characters: [0-9a-zA-Z], \$, -, \_, ., +, \*, ', (, ), , !

- All other characters are escaped by:

%[character hex code]

- Space is encoded as "+" or "%20"

- Example: Наков-爱-SoftUni

- URL-encoded string:

%D0%9D%D0%B0%D0%BA%D0%BE%D0%B2-%E7%88%B1-SoftUni

Char	URL Encoding
space	%20
Щ	%D1%89
"	%22
#	%23
\$	%24
%	%25
&	%26

# Valid and Invalid URLs – Examples

- Some valid URLs:

```
http://www.google.bg/search?sourceid=navclient&ie=UTF-8&rlz=1T4GGLL_enBG369BG369&q=http+get+vs+post
```

```
http://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80%D0%BD%D0%B0%D0%B0%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D0%B8%D1%8F
```

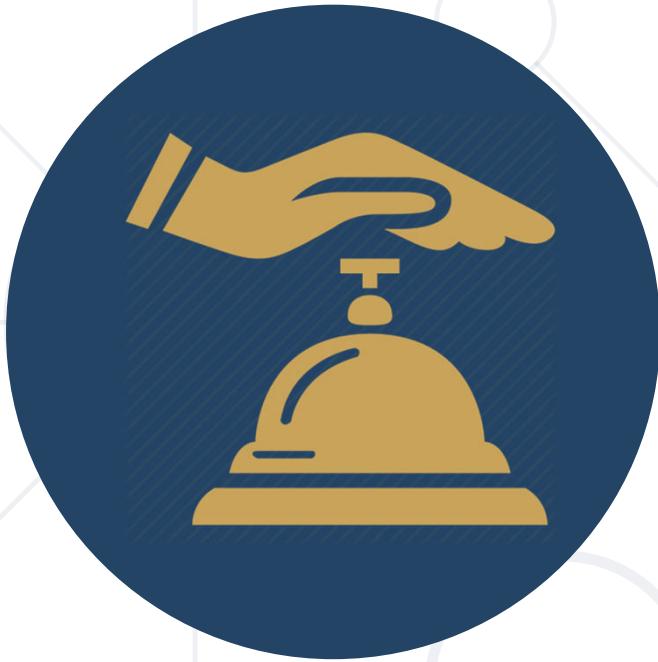
- Some invalid URLs:

```
http://www.google.bg/search?&q=C# .NET 4.0
```

Should be:  
`?q=C%23+.NET+4.0`

```
http://www.google.bg/search?&q=бира
```

Should be: `?q=%D0%B1%D0%B8%D1%80%D0%B0`



# HTTP Request

## What is a HTTP Request?

# HTTP Request Message

Request message sent by a client consists of:

- HTTP **request line**
  - Request method (**GET / POST / PUT / DELETE / ...**)
  - Resource URI (**URL**)
  - Protocol version
- HTTP **request headers**
  - Additional parameters
- HTTP **request body** – optional data, e.g. posted form fields

```
<method> <resource> HTTP/<version>
<headers>
(empty Line)
<body>
```

# HTTP GET Request – Example

- Example of HTTP **GET** request:

```
GET /index.html HTTP/1.1
Host: localhost
<CRLF>
```

HTTP request headers

HTTP request line

The request body is empty

# HTTP POST Request – Example

- Example of HTTP **POST** request:

```
POST /login.html HTTP/1.1
Host: localhost
Content-Length: 59
<CRLF>
username=Mente&password=top*secret
<CRLF>
```

HTTP request line

HTTP request headers

The request body holds  
the submitted form data



# HTTP Response

## What is a HTTP Response?

# HTTP Response Message

- The **response message** sent by the HTTP server consists of:
  - HTTP response **status line**
    - Protocol version
    - Status code
    - Status phrase
  - Response **headers**
    - Provide meta data about the returned resource
  - Response **body**
    - The content of the HTTP response (data)

```
HTTP/<version> <status code> <status text>
<headers>
(empty Line)
<response body - the requested
resource>
```

# HTTP Response – Example

- Example of HTTP **response** from the Web server:

HTTP/1.1 200 OK

HTTP response status line

Date: Fri, 17 Jul 2010 16:09:18 GMT+2

Server: Apache/2.2.14 (Linux)

Accept-Ranges: bytes

Content-Length: 84

Content-Type: text/html

<CRLF>

<html>

<head><title>Test</title></head>

<body>Test HTML page.</body>

</html>

HTTP response  
headers

HTTP response  
body

# HTTP Response Codes

- HTTP response code classes
  - **1xx**: informational (e.g., "**100 Continue**")
  - **2xx**: successful (e.g., "**200 OK**", "**201 Created**")
  - **3xx**: redirection (e.g., "**304 Not Modified**", "**301 Moved Permanently**", "**302 Found**")
  - **4xx**: client error (e.g., "**400 Bad Request**", "**404 Not Found**", "**401 Unauthorized**", "**409 Conflict**")
  - **5xx**: server error (e.g., "**500 Internal Server Error**", "**503 Service Unavailable**")

# HTTP Response – Example

- Example of **HTTP response** with error result:

```
HTTP/1.1 404 Not Found
```

HTTP response status line

```
Date: Fri, 17 Nov 2014 16:09:18 GMT+2
```

```
Server: Apache/2.2.14 (Linux)
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<CRLF>
```

```
<html><head><title>404 Not Found</title></head>
```

```
<body>
```

```
<h1>Not Found</h1>
```

```
<p>The requested URL /img/logo.gif was not found on this server.</p>
```

```
<hr><address>Apache/2.2.14 Server at Port 80</address>
```

```
</body></html>
```

HTTP response headers

The HTTP response body

# Browser Redirection

- HTTP **GET** requesting a moved URL:

```
GET / HTTP/1.1
Host: http://softuni.org
User-Agent: Gecko/20100115 Firefox/3.6
<CRLF>
```

- The following HTTP response (**301 Moved Permanently**) tells the browser to request another URL:

```
HTTP/1.1 301 Moved Permanently
Location: http://softuni.bg
...
```

# Content-Type and Disposition

- The **Content-Type** response header the server specifies how the output should be processed
- Examples:

UTF-8 encoded HTML page.  
Will be shown in the browser.

**Content-Type: text/html; charset=utf-8**

**Content-Type: application/pdf**  
**Content-Disposition: attachment; filename="Report-April-2016.pdf"**

This will download a PDF file named  
**Report-April-2016.pdf**



# MIME and Media Types

## Multi-Purpose Internet Mail Extensions

# What is MIME?

- **MIME** == Multi-Purpose Internet Mail Extensions
  - Internet standard for encoding resources
  - Originally developed for email attachments
  - Used in many Internet protocols like HTTP and SMTP
- MIME defines several concepts
  - Content-Type, e.g. `text/html`, `image/gif`, `application/pdf`
    - Content charset, e.g. `utf-8`, `ascii`, `windows-1251`
  - Content-Disposition, e.g. `attachment; filename=logo.jpg`
  - Multipart messages (multiple resources in a single document)

# Common MIME Media Types

MIME Type / Subtype	Description
application/json	JSON data
image/png	PNG image
image/gif	GIF image
text/html	HTML
text/plain	Text
text/xml	XML
video/mp4	MP4 video
application/pdf	PDF document



# HTTP Tools for Developers



# HTTP Tools for Developers - Browser (2)

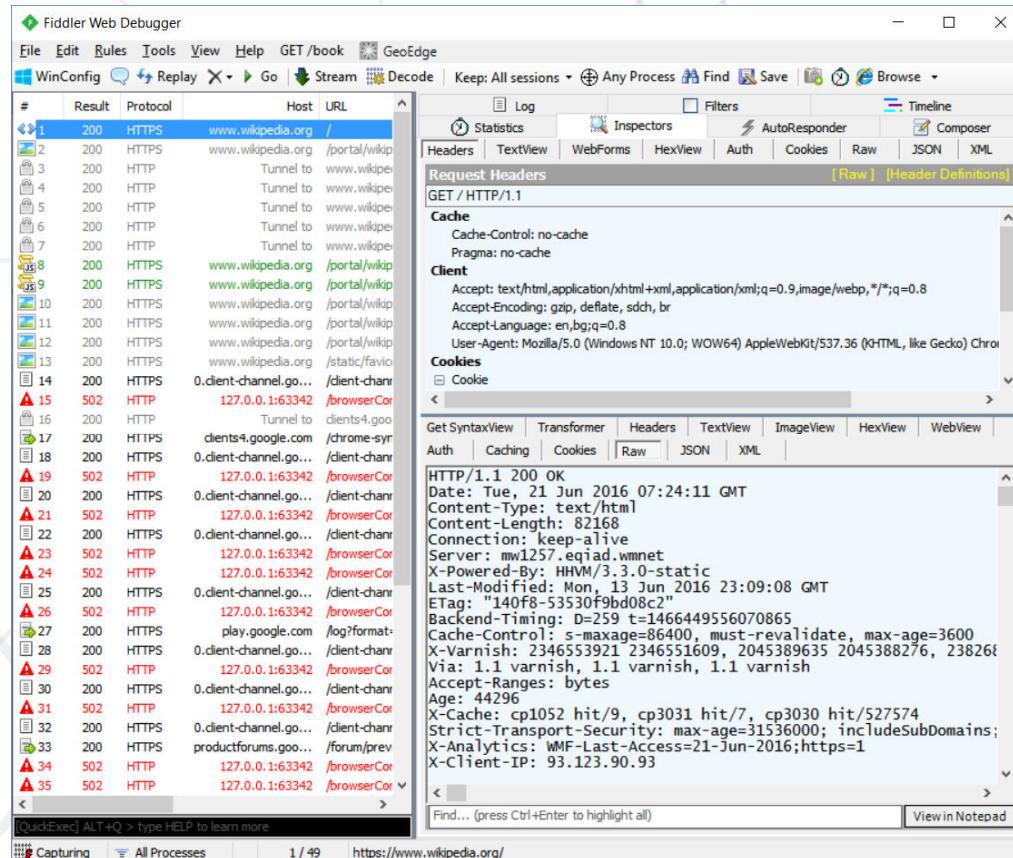


[Postman](#) - Chrome



[RESTClient](#) - Firefox

# HTTP Tools for Developers - Desktop



Fiddler



# Web Server

# What is a Web Server?

- Computer system that processes requests via **HTTP**, the basic network protocol

Web Client

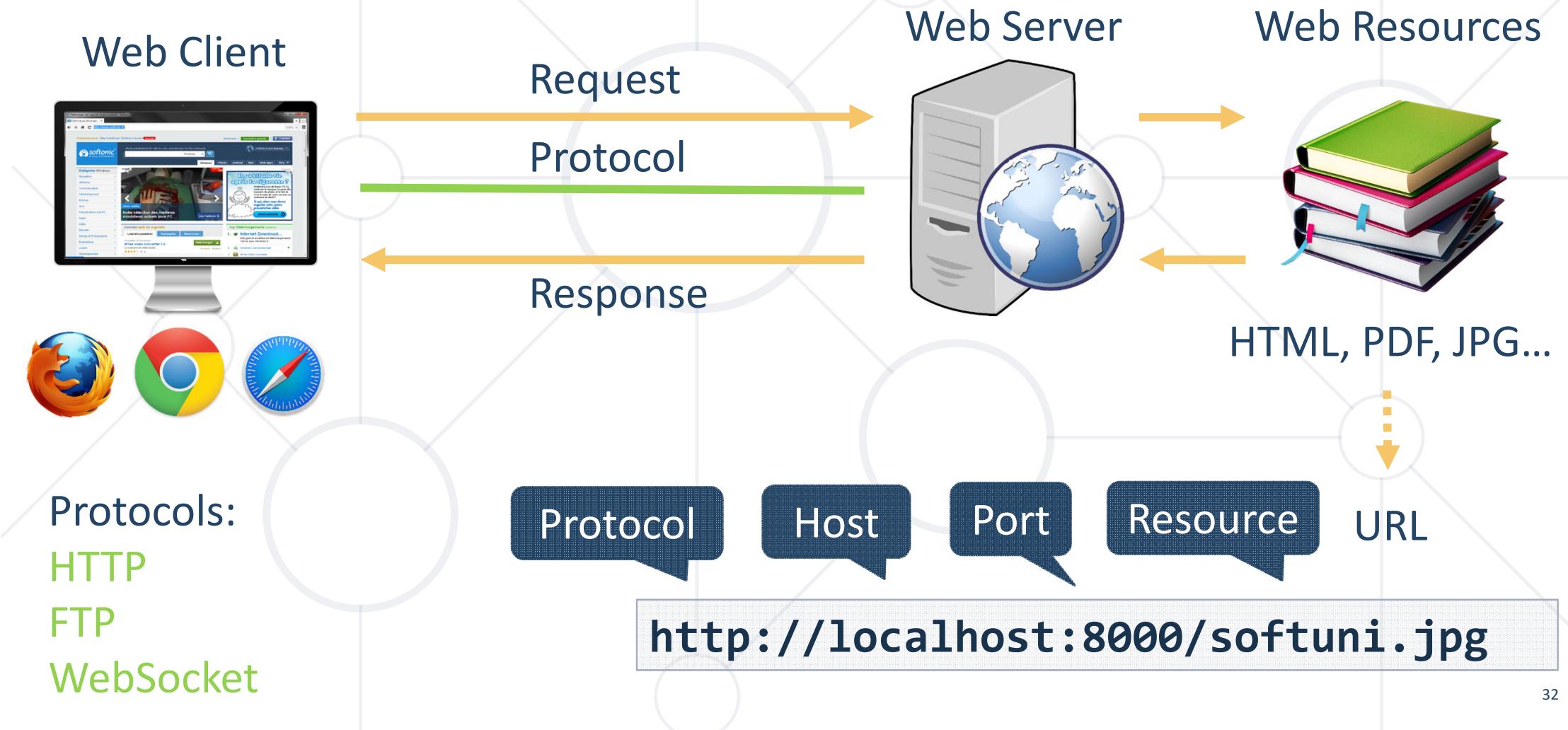


Web Server

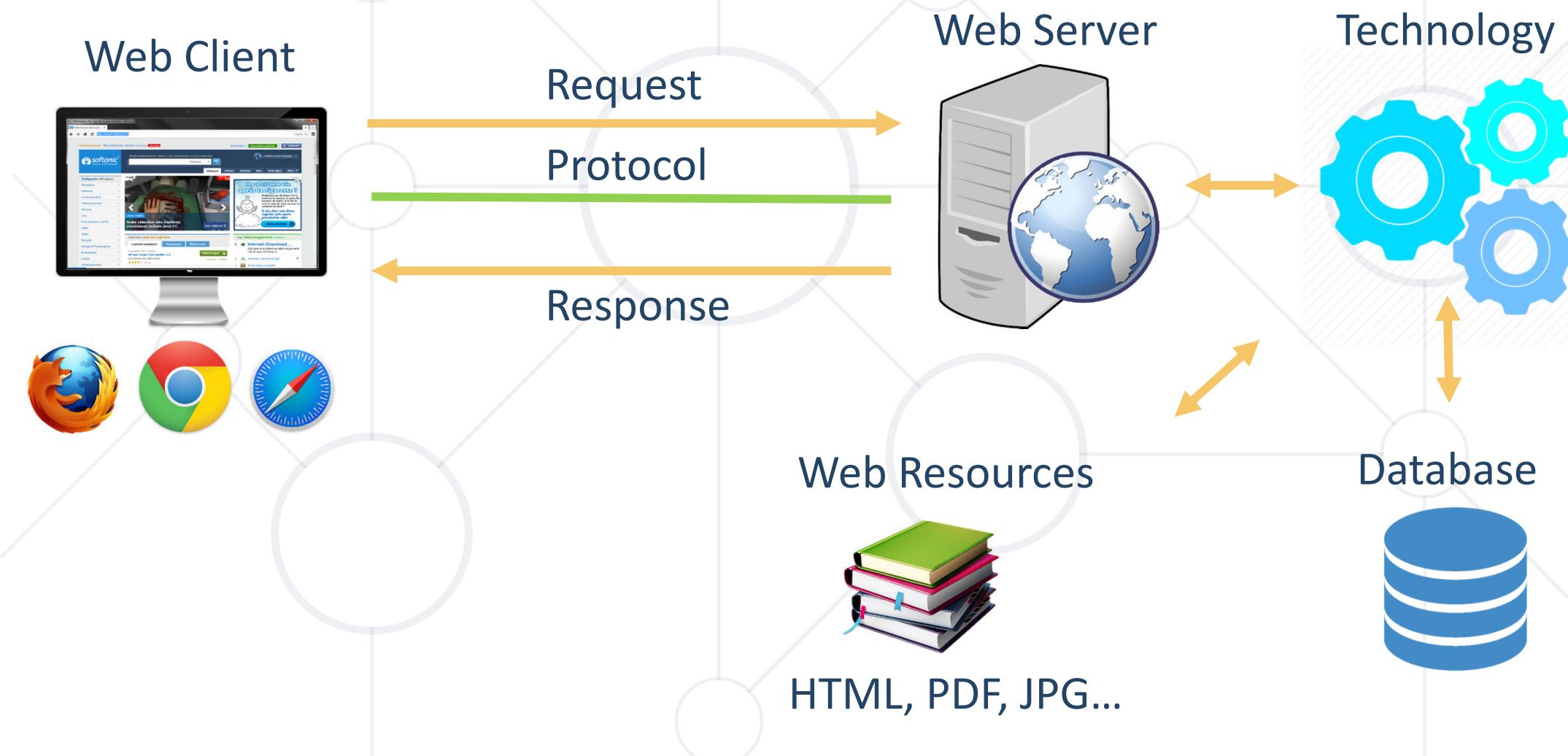


Communication

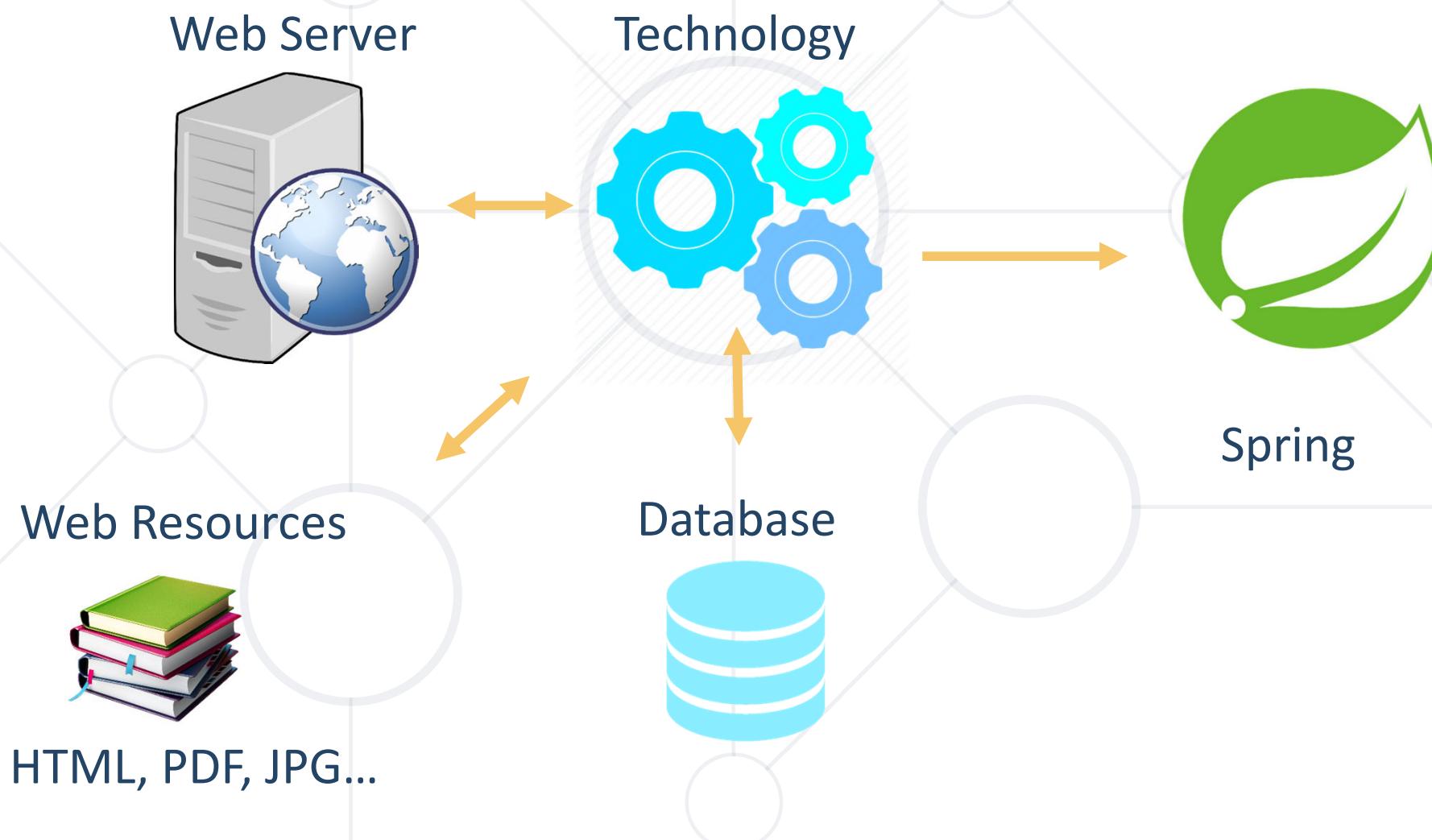
# Web Server Work Model



# Web Server Work Model (2)



# Web Server Work Model (3)



# Most Popular Web Servers (W3Techs)

Product	Vendor	Percent
Apache	Apache	48.5%
nginx	NGINX, Inc.	35.4%
IIS	Microsoft	10.8%
LiteSpeed Web Server	LiteSpeed Technologies	2.9%
GWS	Google	1.1%

OCTOBER 2017



July 2018 [edit]

Product	Vendor	Percent
Apache	Apache	45.9%
nginx	NGINX, Inc.	39.0%
IIS	Microsoft	9.5%
LiteSpeed Web Server	LiteSpeed Technologies	3.4%
GWS	Google	1.0%

# Most Popular Web Servers (NetCraft)

Product	Vendor	January 2016	Percent	February 2016	Percent	Change
Apache	Apache	304,271,061	33.56%	306,292,557	32.80%	0.76
IIS	Microsoft	262,471,886	28.95%	278,593,041	29.83%	0.88
nginx	NGINX, Inc.	141,443,630	15.60%	137,459,391	16.61%	-0.88
GWS	Google	20,799,087	2.29%	20,640,058	2.21%	-0.08

Product	Vendor	January 2017	Percent	February 2017	Percent	Change	Chart color
IIS	Microsoft	821,905,283	45.66%	773,552,454	43.16%	-2.50	red
Apache	Apache	387,211,503	21.51%	374,297,080	20.89%	-0.63	black
nginx	NGINX, Inc.	317,398,317	17.63%	348,025,788	19.42%	1.79	green
GWS	Google	17,933,762	1.00%	18,438,702	1.03%	0.03	blue

Developer	July 2018	Percent	August 2018	Percent	Change
Microsoft	670,209,364	40.28%	675,366,733	40.65%	0.36
Apache	366,023,119	22.00%	336,937,568	20.28%	-1.72
nginx	287,045,450	17.25%	300,644,222	18.10%	0.84
Google	23,347,290	1.40%	23,143,610	1.39%	-0.01



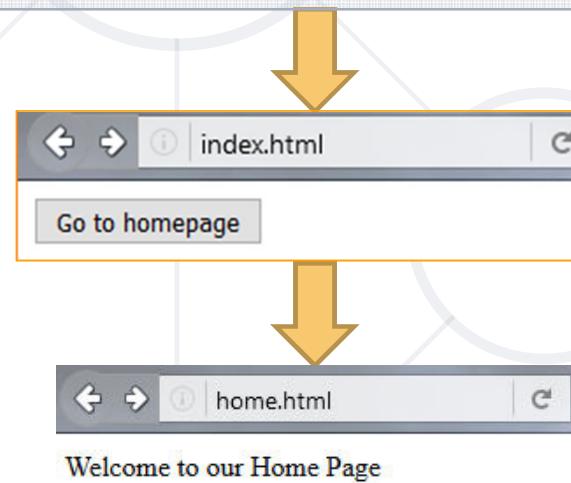
# HTML Forms

## Form Method and Action

# HTML Forms - Action Attribute

- Defines where to submit the form data:

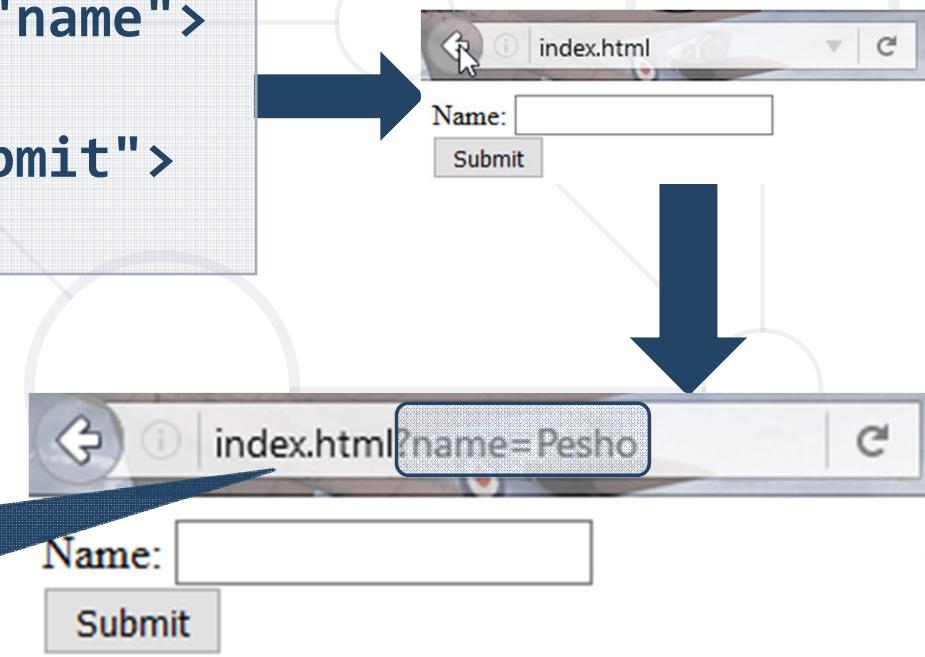
```
<form action="home.html">  
  <input type="submit" value="Go to homepage"/>  
</form>
```



# HTML Forms – Method Attribute

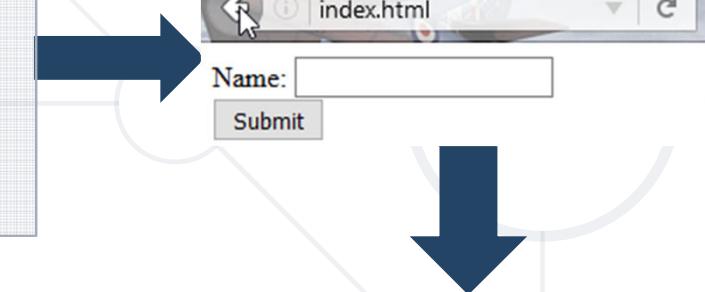
- Specifies the HTTP method to use when sending form-data

```
<form method="get">  
    Name: <input type="text" name="name">  
    <br>  
    <input type="submit" value="Submit">  
</form>
```



# HTML Forms – Method Attribute (2)

```
<form method="post">  
    Name: <input type="text" name="name">  
    <br>  
    <input type="submit" value="Submit">  
</form>
```



POST http://localhost/index.html HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

Content-Length: 10

name=Pesho

HTTP request body  
holds the form data

# URL Encoded Form Data – Example

```
<form method="post">  
    Name: <input type="text" name="name"/> <br/>  
    Age: <input type="text" name="age"/> <br/>  
    <input type="submit" />  
</form>
```

index.cgi

Name:

Age:



POST http://localhost/cgi-bin/index.cgi HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

Content-Length: 23

name=Maria+Smith&age=19

File uploads are not  
supported



# HTTP/2

**What's new? What's better? What's HTTP/2**

# What's HTTP/2

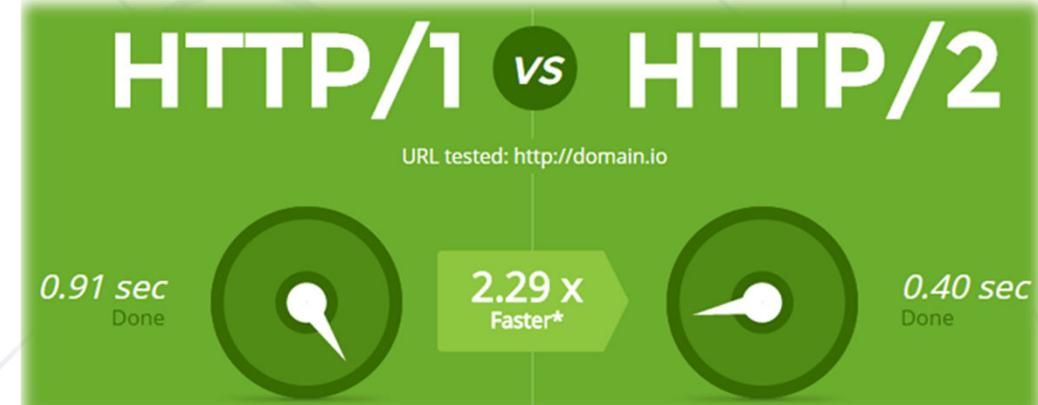
- **HTTP/2** (originally named **HTTP/2.0**) major revision of the **HTTP** network protocol used by the **World Wide Web**.
  - Supported by most of the popular web browsers (Chrome, Mozilla, Opera...)
- Fast & Optimized. Meets modern web usage requirements.
- Completely Backwards-Compatible.
- As of January 2019, **32.5%** of the top 10 million websites support **HTTP/2** (W3Techs statistics).



# What's new?

- **HTTP/2** is meant to erase the need of maintaining complex server infrastructures in order to perform well.
- **HTTP/2** communicates in binary data frames.
- **HTTP/2** introduces several new important elements:
  - HTTP/2 Multiplexing
  - HTTP/2 Header Compression
  - HTTP/2 Server Push

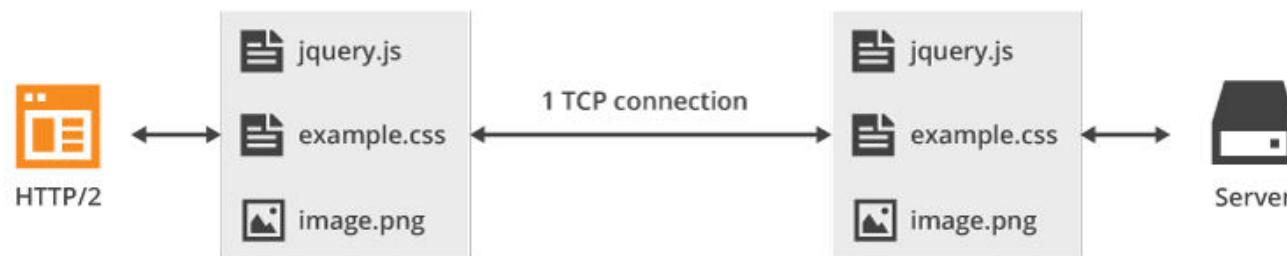
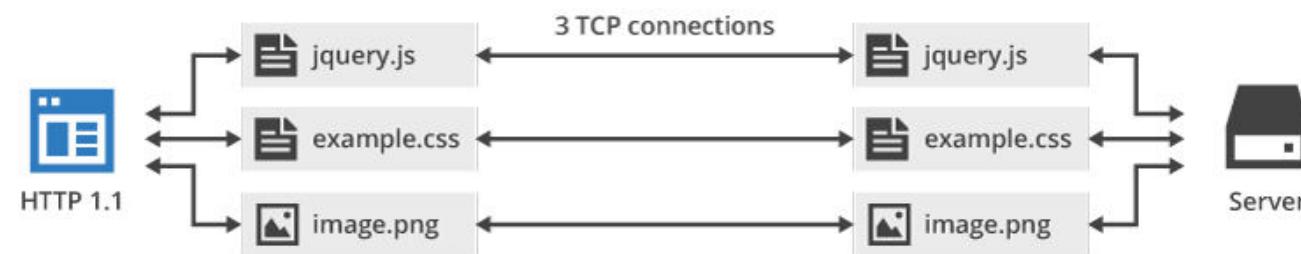
HTTP / 2



# HTTP/2 Multiplexing

- The art of handling multiple streams over a **single** TCP connection.

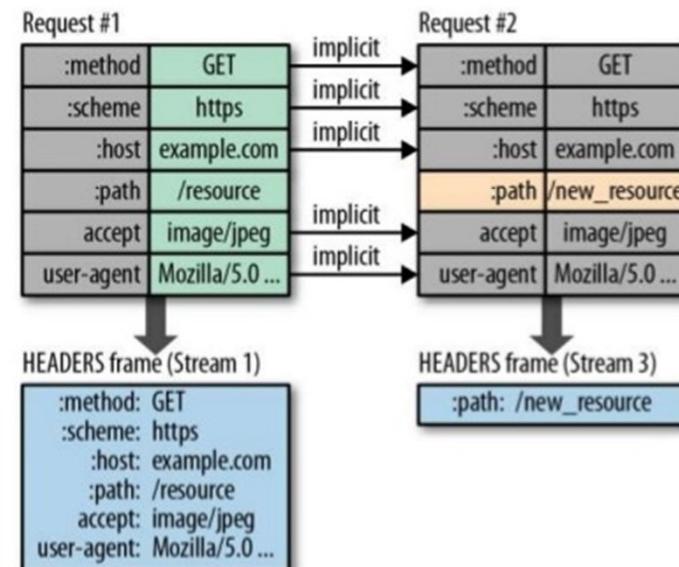
Multiplexing



# HTTP/2 Header Compression

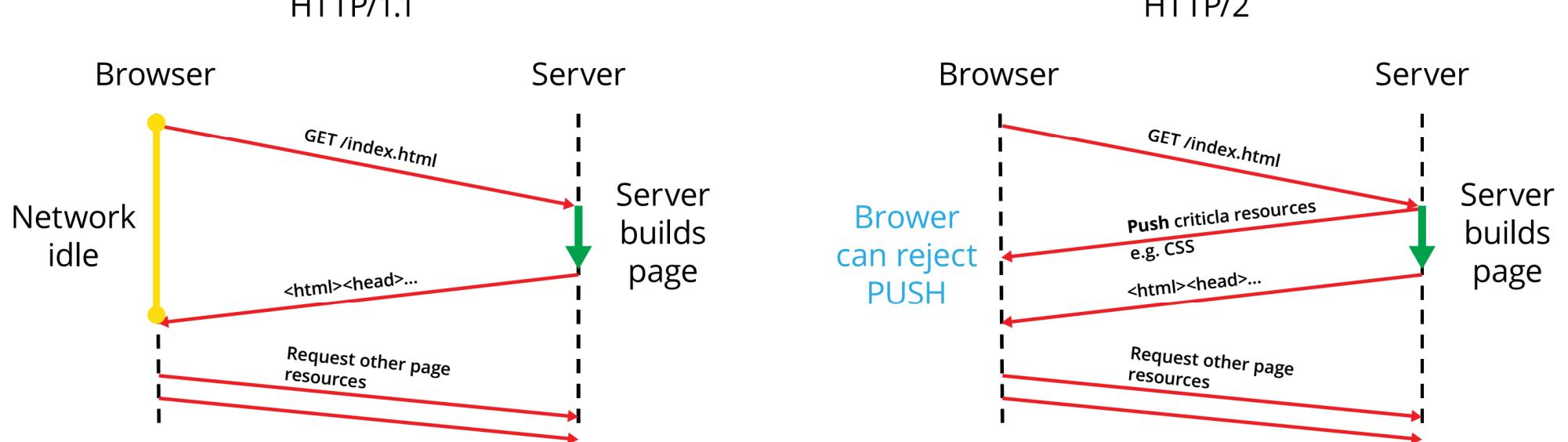
- HTTP/2 maintains a **HTTP Header Table** across requests.
- Optimizes communication drastically.
- The process is essentially a **de-duplication**, rather than compression.

## HTTP/2 Header Compression



# HTTP/2 Server Push

- **HTTP/2 Server Push** is the process of sending resources to clients, without them having to ask for it.



# Summary

- What is **HTTP**?
  - **HTML** Forms & Actions.
  - **URLs**.
- What is a **Web Server**?
  - Beneath the sheets of **Web Communication**.
- How to handle **HTTP Requests**?
- How to construct **HTTP Responses**?



# Questions?



**SoftUni**



**Software  
University**



**SoftUni  
Svetlina**



**SoftUni  
Creative**



**SoftUni  
Digital**



**SoftUni  
Foundation**



**SoftUni  
Kids**

<https://softuni.bg/trainings/courses>

# SoftUni Diamond Partners



# SoftUni Organizational Partners



ИНФОРМАЦИОННО  
ОБСЛУЖВАНЕ

One  
SOFTW



Lukanet.com



# Trainings @ Software University (SoftUni)



- Software University – High-Quality Education and Employment Opportunities
  - [softuni.bg](http://softuni.bg)
- Software University Foundation
  - <http://softuni.foundation/>
- Software University @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- Software University Forums
  - [forum.softuni.bg](http://forum.softuni.bg)



# License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

