HTTP Protocol

Handling Requests,
Constructing Responses, HTTP/2

HTTP

SoftUni Team Technical Trainers







Software University

https://softuni.bg

Table of Content



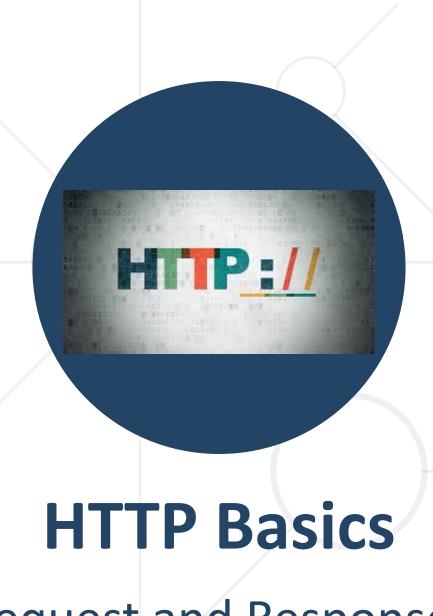
- 1. Http Basics
- 2. URL
- 3. Http Request
- 4. Http Response
- 5. MIME and Media types
- 6. HTTP Tools
- 7. Web Server
- 8. Html Forms
- 9. Http/2

Questions?





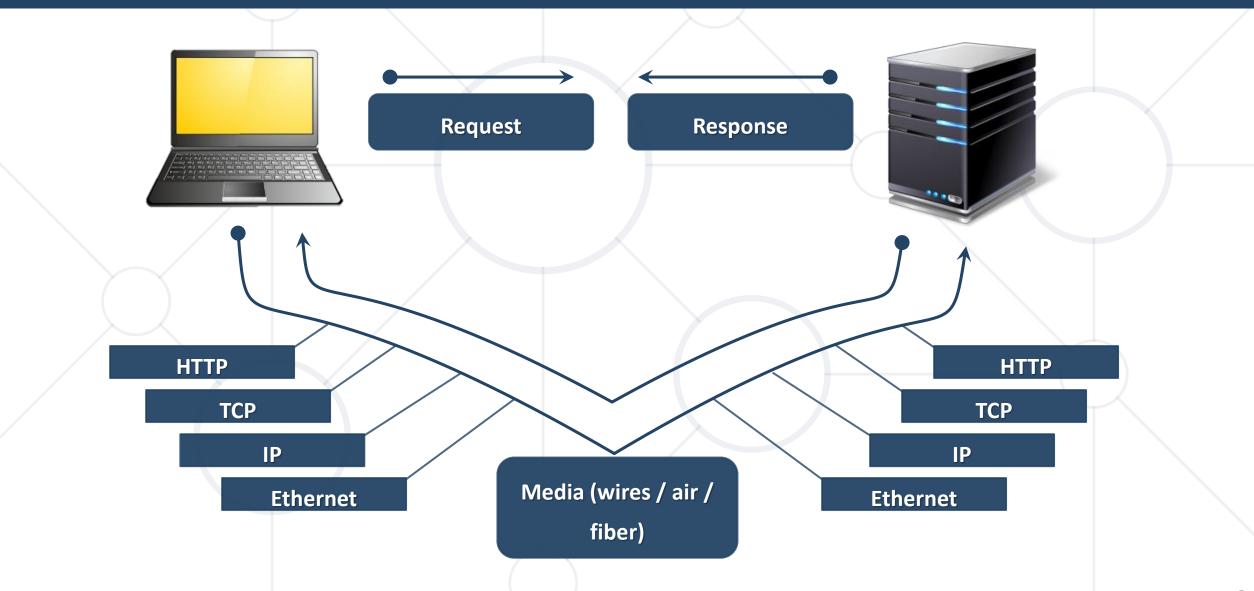
#java-web



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:

GET /courses/javascript HTTP/1.1

Host: www.softuni.bg

User-Agent: Mozilla/5.0

<html><title>Hello</title>

Welcome to our site</html>

<CRLF>

The empty line denotes the end of the request header

HTTP response:

HTTP/1.1 200 OK
Date: Mon, 5 Jul 2020 13:09:03 GMT
Server: Microsoft-HTTPAPI/2.0
Last-Modified: Mon, 12 Jul 2014 15:33:23 GMT
Content-Length: 54
<CRLF>
The empty line denotes the

end of the response header



Uniform Resource Locator

Uniform Resource Locator (URL)



```
http://localhost:8080/demo/index.html?id=27&lang=en#lecture

Protocol Host Port Path Query Fragment String

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.html)
- 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"

Наков-爰-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
П	%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:

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

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

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

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



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
- <method> <resource> HTTP/<version>
 <headers>
 (empty line)
 <body>

HTTP request body – optional data, e.g. posted form fields

HTTP GET Request – Example



Example of HTTP GET request:

```
GET /index.html HTTP/1.1 HTTP request line
Host: localhost

CRLF>
HTTP request headers
```

The request body is empty

HTTP POST Request – Example



Example of HTTP POST request:

```
POST /login.html HTTP/1.1 HTTP request line

Host: localhost
Content-Length: 59
Content-Type: application/x-www-form-urlencoded

<CRLF>
username=testUser&password=topSecret

<CRLF>
The request body holds
```

the submitted form data

15



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 response status line
HTTP/1.1 200 OK
Date: Fri, 17 Jul 2020 16:09:18 GMT+2
Server: Apache/2.2.14 (Linux)
Accept-Ranges: bytes
                                  HTTP response
Content-Length: 84
                                     headers
Content-Type: text/html
<CRLF>
<html>
                                         HTTP response
  <head><title>Test</title></head>
                                             body
  <body>Test HTML page.</body>
</html>
```

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 Error Response – Example



Example of HTTP response with error result:

```
HTTP response status line
HTTP/1.1 404 Not Found
Date: Fri, 17 Nov 2020 16:09:18 GMT+2
Server: Apache/2.2.14 (Linux)
Connection: close
                                          HTTP response headers
Content-Type: text/html
<CRLF>
<html><head><title>404 Not Found</title></head>
<body>
                                                  The HTTP response body
<h1>Not Found</h1>
The requested URL /img/logo.gif was not found on this server.
<hr><address>Apache/2.2.14 Server at Port 80</address>
</body></html>
```

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 eoutput should be processed

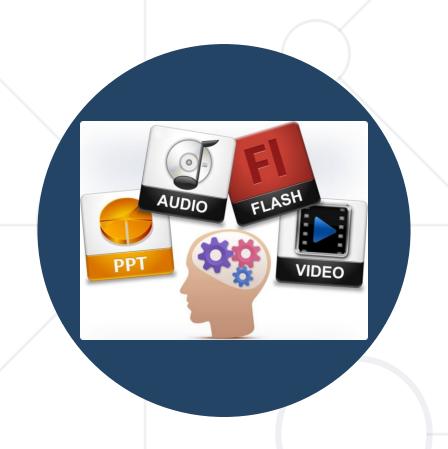
Examples:

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-2020.pdf"
```

This will download a PDF file named Report-April-2020.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

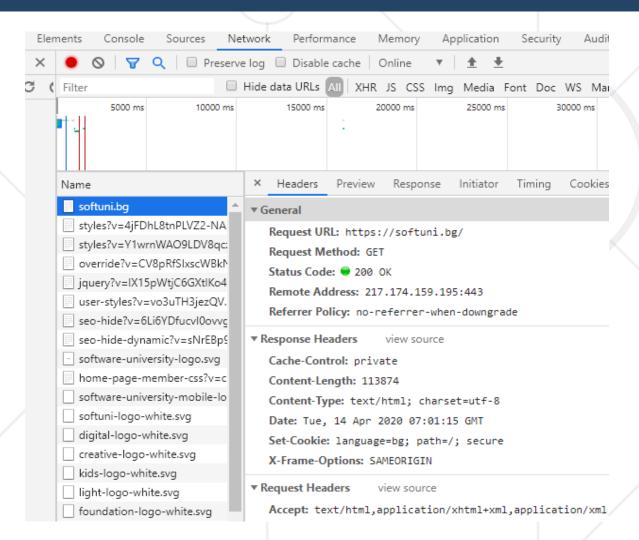


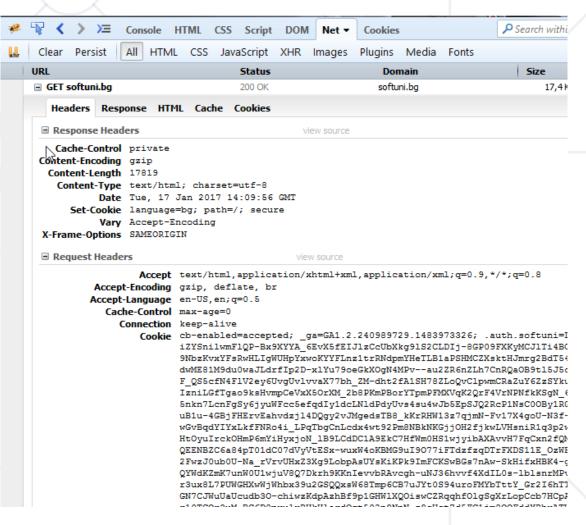
MIME Type / Subtype	Description
application/json	JSON data
image/png	PNG image
image/gif	GIF image
text/html	HTML
text/plain	Text
text/xml	XML
video/mp4	MP4 video
application/pdf	PDF document



HTTP Tools for Developers – Browser (1)





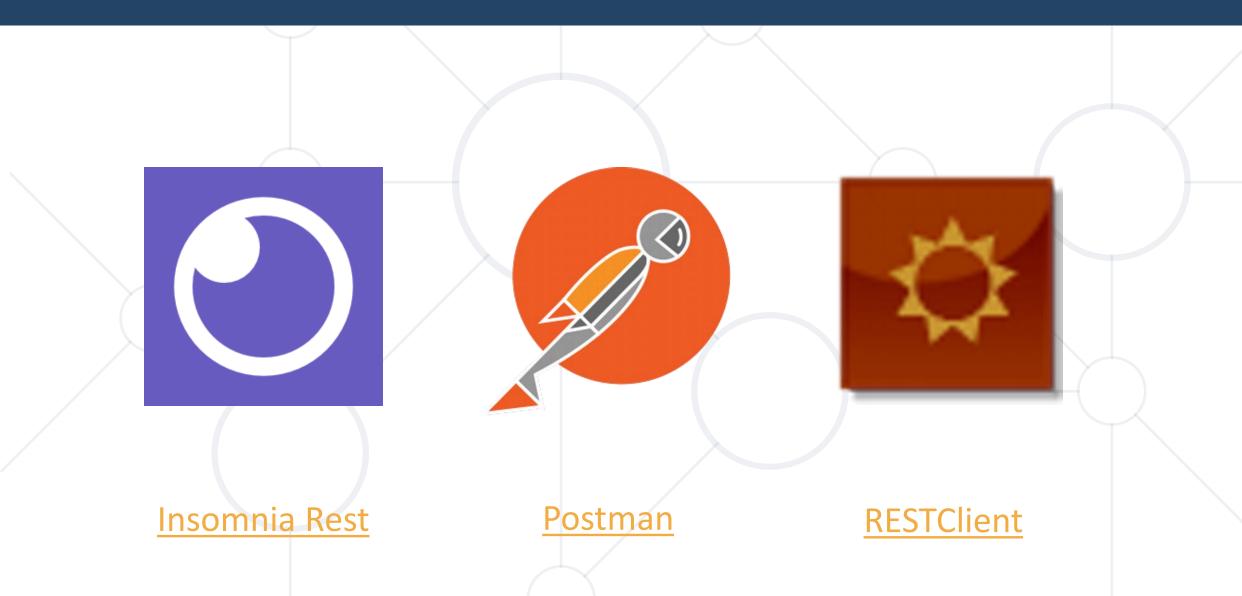


Chrome Developer Tools

<u>Firebug</u>

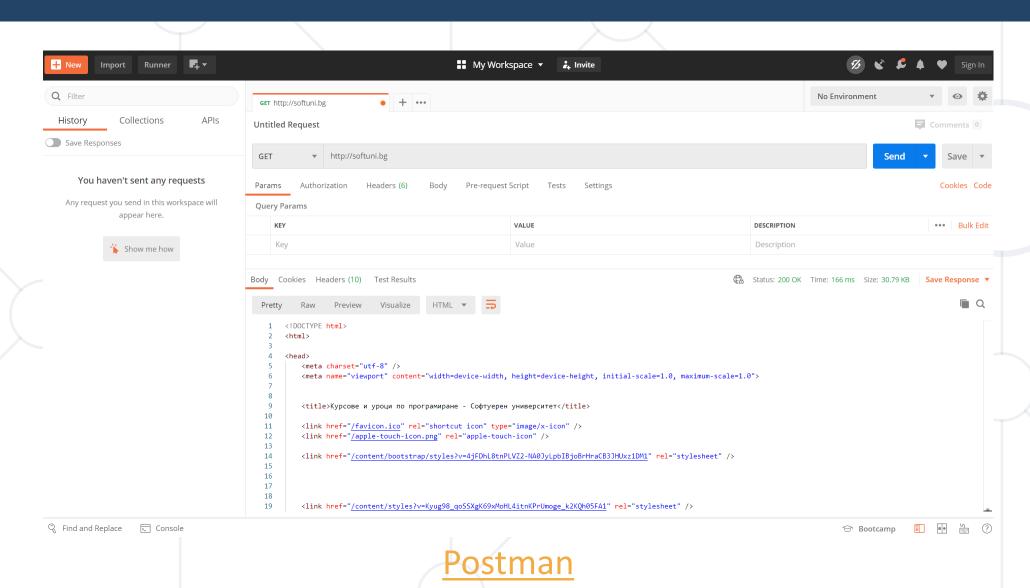
HTTP Tools for Developers – Browser (2)





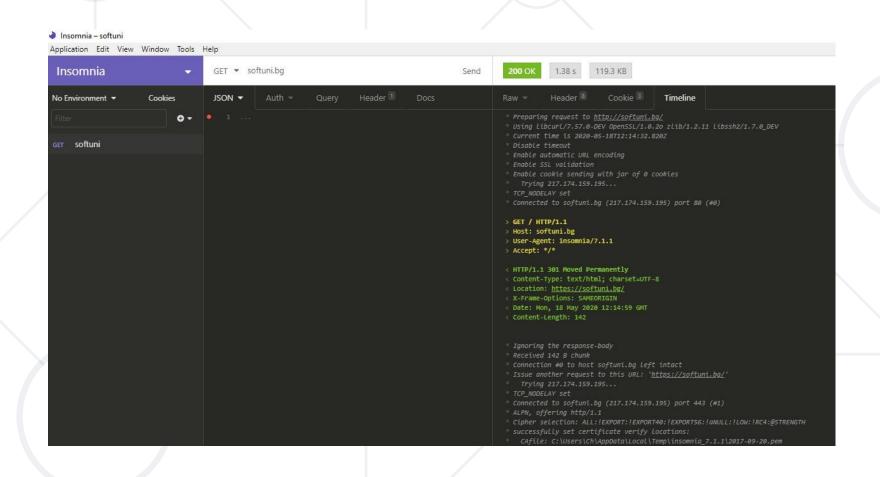
HTTP Tools for Developers – Postman





HTTP Tools for Developers – Insomnia





Insomnia Rest



What is a Web Server?



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

Web Client









Communication





Web Server Work Model (1)



Web Client







Protocols:

FTP

WebSocket



Resource Port Host Protocol **URL**

http://localhost:8000/softuni.jpg

Web Server Work Model (2)



Web Client





Request

Protocol

Response





Technology



Web Resources



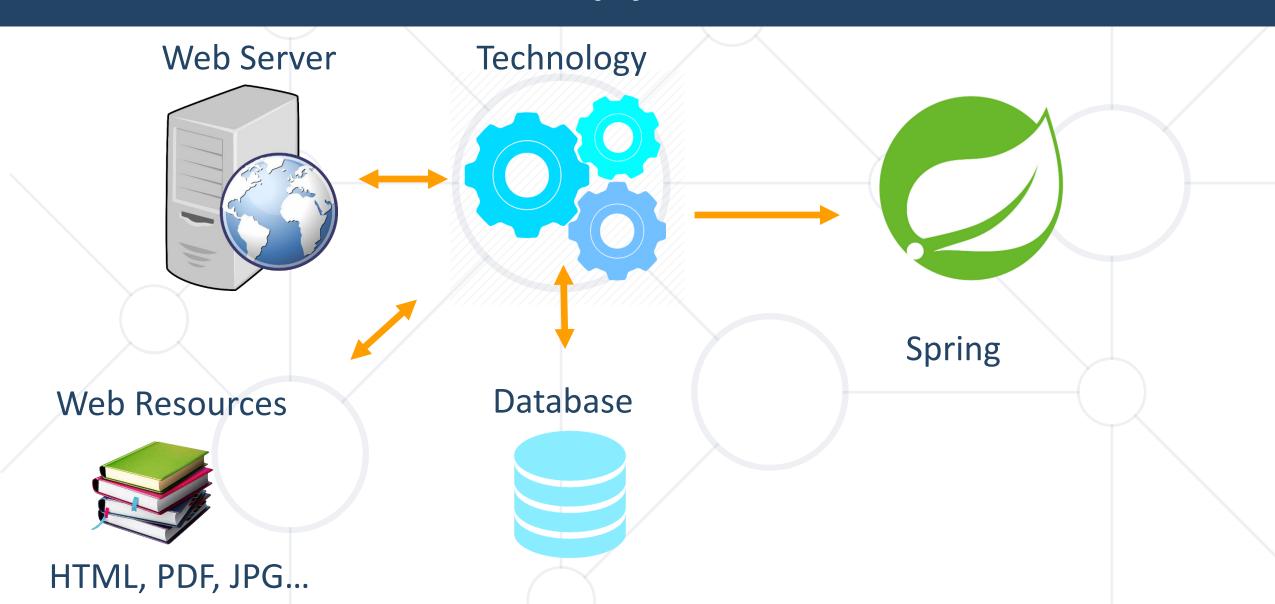
HTML, PDF, JPG...

Database



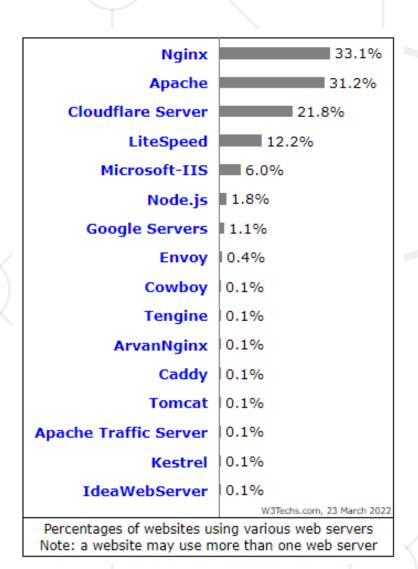
Web Server Work Model (3)





Most Popular Web Servers (W3Techs)







Most Popular Web Servers (NetCraft)



Developer	November 2021	Percent	December 2021	Percent	Change
nginx	408,226,319	34.73%	384,347,394	32.88%	-1.85
Apache	286,494,600	24.37%	283,409,491	24.25%	-0.13
OpenResty	76,480,927	6.51%	78,902,138	6.75%	0.24
Cloudflare	58,629,365	4.99%	59,904,450	5.13%	0.14

Developer	January 2022	Percent	February 2022	Percent	Change
nginx	377,019,054	32.29%	364,956,731	31.10%	-1.19
Apache	279,709,815	23.95%	277,928,961	23.68%	-0.27
OpenResty	80,238,470	6.87%	90,652,376	7.72%	0.85
Cloudflare	60,881,028	5.21%	62,423,819	5.32%	0.11



HTML Forms – Action Attribute



Defines where to submit the form data:

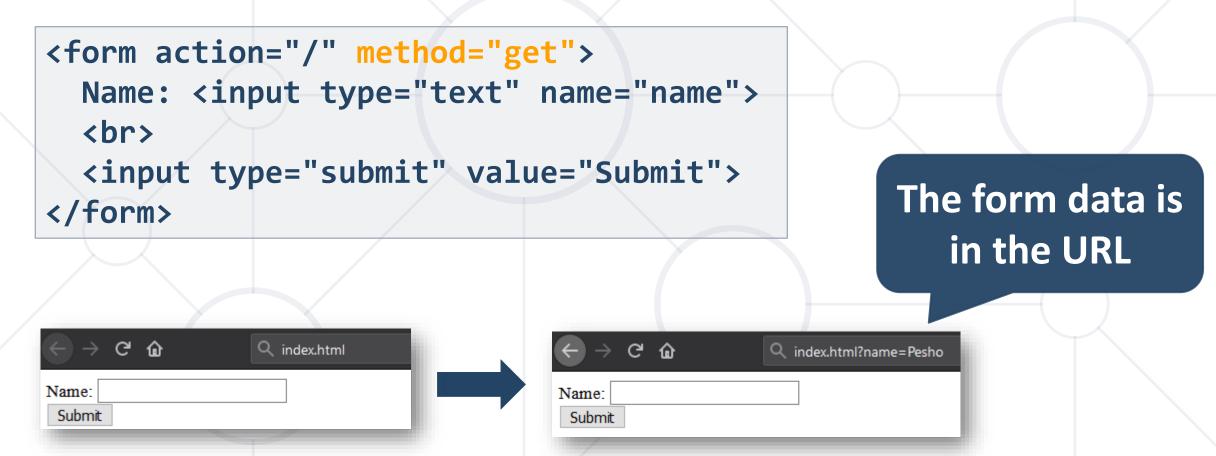
```
<form action="home.html">
    <input type="submit" value="Go to homepage"/>
</form>
```



HTML Forms - Method Attribute (1)



Specifies the HTTP method to use when sending form-data



HTML Forms – Method Attribute (2)



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 action="/" method="post">
Name: <input type="text" name="name"/> <br/>
Age: <input type="text" name="age"/> <br/>
<input type="submit" />
</form>
Name: Maria Smith
Age: 19
Submit Query
```

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

Host: localhost

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

Content-Length: 23

File uploads are name=Maria+Smith&age=19

not supported
```



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 Jan 2021, 50% of all the websites support HTTP/2 (W3Techs statistics).



What's New?



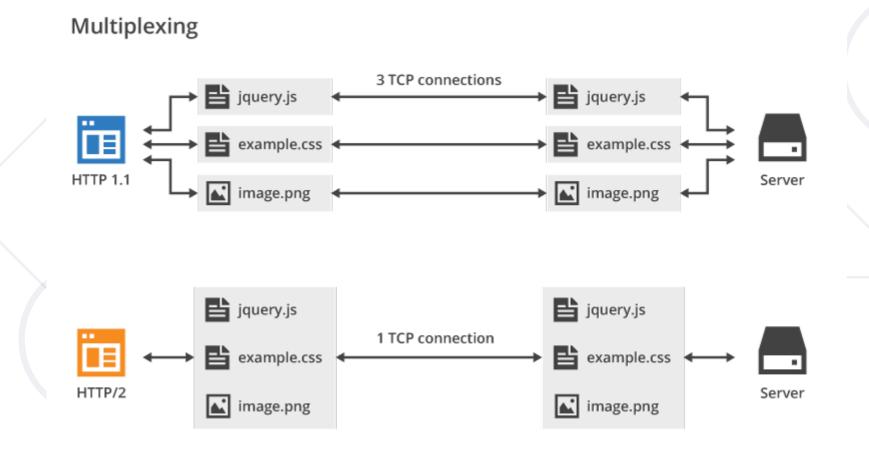
- HTTP/2 is meant to erase the need of maintaining complex server infrastructures in order to perform well. HTTP/2
- HTTP/2 communicates in binary data frames.
- HTTP/2 introduces several new important elements
 - HTTP/2 Multiplexing
 - HTTP/2 Header Compression
 - HTTP/2 Server Push



HTTP/2 Multiplexing



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

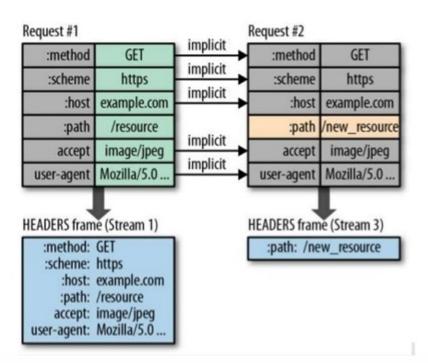


HTTP/2



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

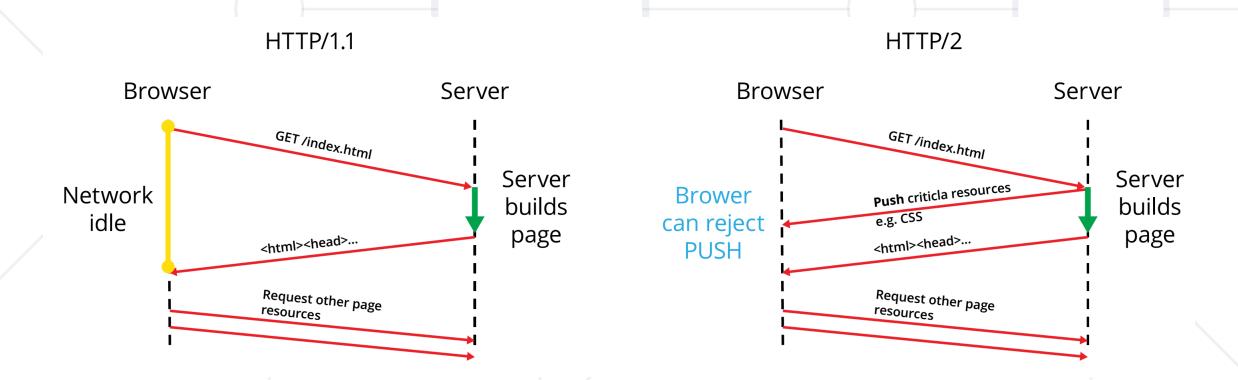




HTTP/2 Server Push



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



What's HTTP/3



- HTTP/3 is a new standard in development that will affect how web browsers and servers communicate.
- Significant upgrades for user experience
- Performance, Reliability, and Security

 HTTP/3 runs on QUIC, a new transport protocol designed for mobile-heavy Internet usage.



Summary



- HTTP:
 - HTML Forms & Actions.
 - URLs
 - Request
 - Response
- Web Server:
 - Web Communication





Questions?

















SoftUni Diamond Partners































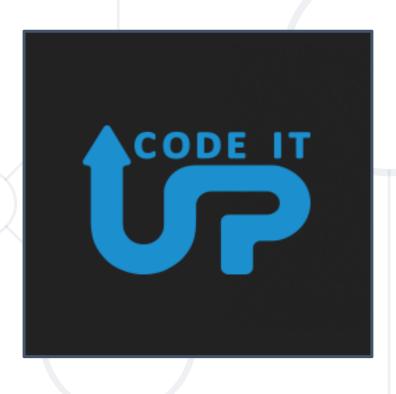






Educational Partners





VIRTUAL RACING SCHOOL



Trainings @ Software University (SoftUni)



- Software University High-Quality Education,
 Profession and Job for Software Developers
 - softuni.bg, about.softuni.bg
- Software University Foundation
 - softuni.foundation
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg









License



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is copyrighted content
- Unauthorized copy, reproduction or use is illegal
- © SoftUni https://about.softuni.bg/
- © Software University https://softuni.bg

