Internet and HTTP

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Table of Contents

Have a Question?

Software University

sli.do

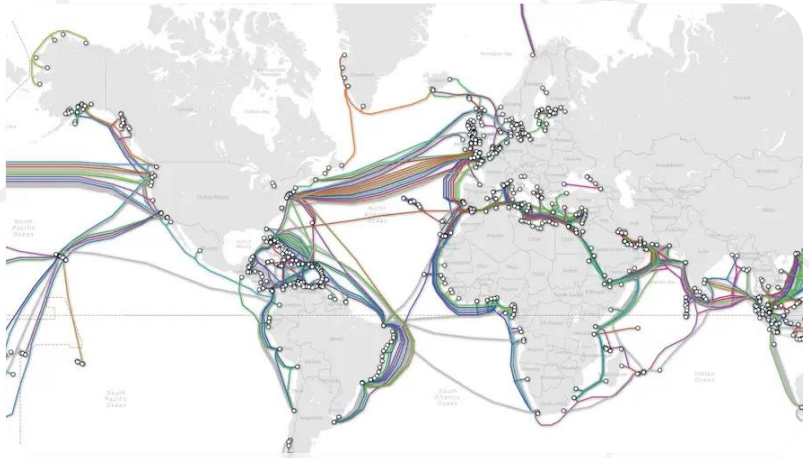
#python-web

Introduction

An Introduction to the Internet

What is the Internet?

- **Vast network** that connects billions of devices together all over the globe
- Through **fiber optics**, **copper**, **satellites** or **cell phone network**
- We get indirectly **connected** though **ISP**s (**I**nternet **S**ervice **P**roviders)



Networks and Internet

- **Network** is a group of **two or more devices** that can communicate
- **The internet** is made of hundreds of thousands of **networks**
- These different systems **connect to each other**, **communicate with each other** and **work together** because of standards for how data is sent

Web Server Work Model

Software University

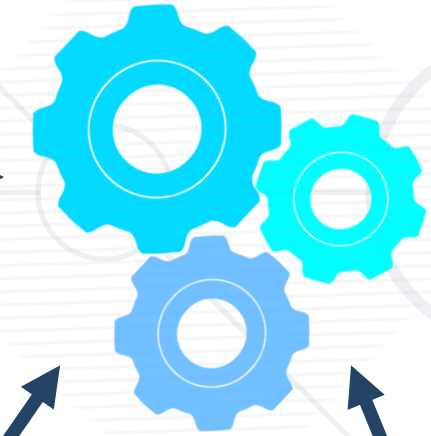
Web Client

Web Server

Technology

Request

Response



Web Resources

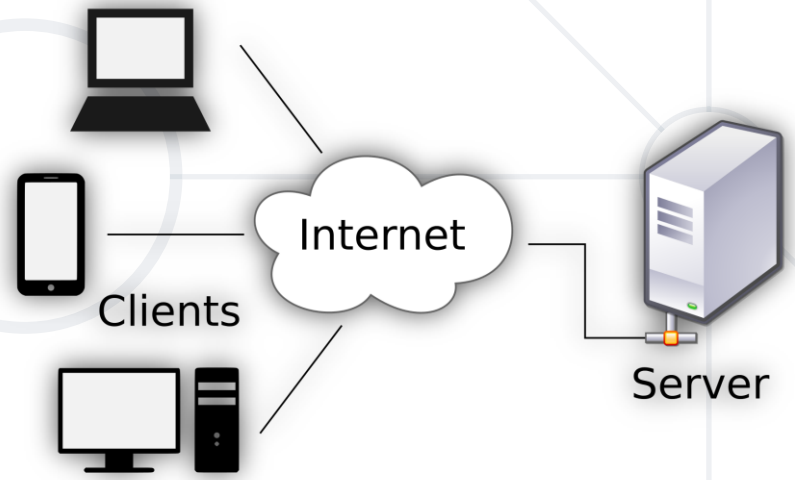
HTML, PDF, JPG…

Database

Important Definitions

Servers and Clients

- **Servers** are the machines that provide services to other machines
- **Clients** are the machines that are used to connect to those services

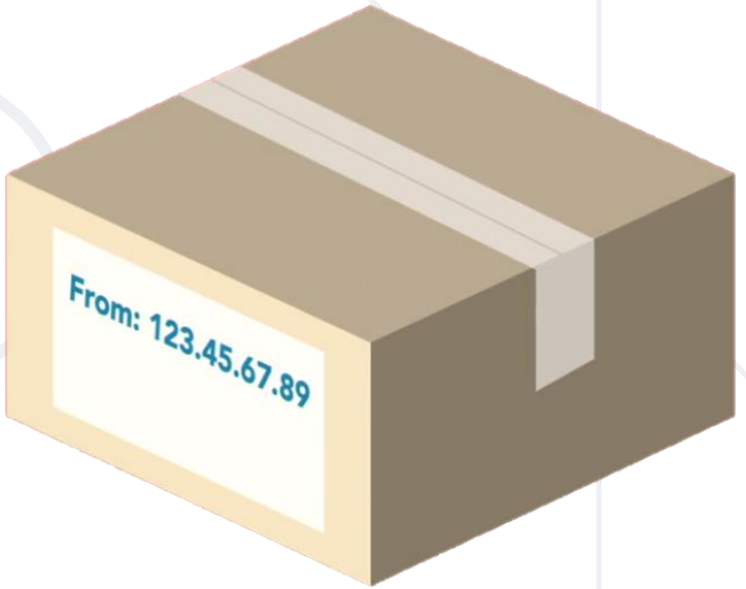


Network Protocol

- Set of **rules** and **standards**, that allow communication between network devices
- Include **mechanisms** for devices to **identify** and make **connections** with each other
- Example for standard network protocols:
  - TCP, UDP, IP, ARP
  - HTTP, FTP, TFTP, SMPT, SSH

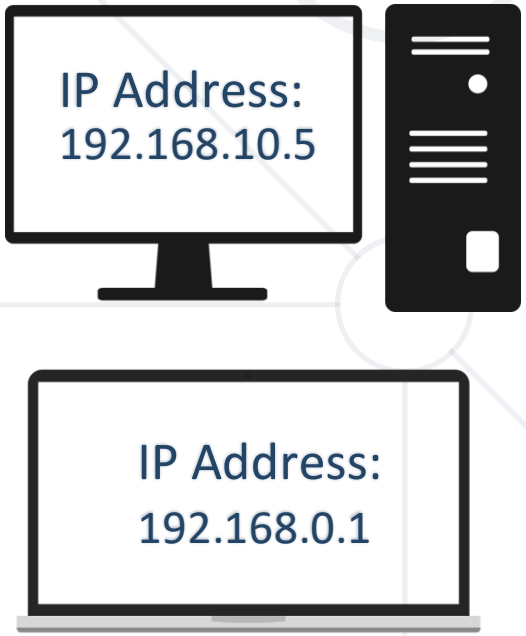
Packets

- Every message, file or stream of information **sent over computer networks** is broken down into small chunks called **packets**
- Each packet contains **important information** inside of it called a **header**:
  - Contents
  - Origin
  - Destination



Internet Protocol (IP)

- All the devices on the Internet have **IP Addresses**
- Each IP address is **unique** to each computer or a device at the edge of the network



IP Address

- An **IP Address** has many parts, organized in a hierarchy

Subnetworks

192.168.14.120

Device address

- This version of IP Addressing is called **IPv4**
  - Provides more than 4 billion **32 bits** unique addresses

IP address classes

Software University

Class	Address range	Supports
Class A	1.0.0.1 to 126.255.255.254	Supports 16 million hosts on each of 127 networks.
Class B	128.1.0.1 to 191.255.255.254	Supports 65,000 hosts on each of 16,000 networks.
Class C	192.0.1.1 to 223.255.254.254	Supports 254 hosts on each of 2 million networks.
Class D	224.0.0.0 to 239.255.255.255	Reserved for multicast groups.
Class E	240.0.0.0 to 254.255.255.254	Reserved for future use, or research and development purposes.

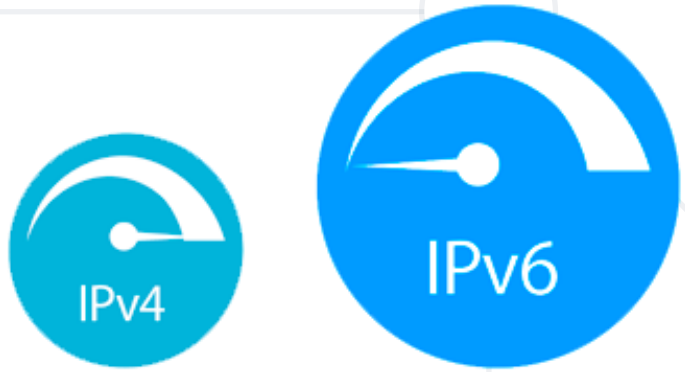
Classless Inter-Domain Routing

- CIDR is an **IP addressing scheme** that improves the allocation of IP addresses
- Replaces the old system based on classes A, B, and C
- Helps greatly **extend the life of IPv4**

CIDR	IP address range	Class
10.0. 0.0/8	10.0. 0.0 – 10.255. 255.255	A
172.16. 0.0/12	172.16. 0.0 – 172.31. 255.255	B
192.168. 0.0/16	192.168. 0.0 – 192.168. 255.255	C

IPv6

- **IPv6** uses **128 bits**
- Example of a full IPV6 address:
  - **3FFE:F200:0234:AB00:0123:4567:8901:ABCD**
  - The **leading zeros** can usually be left out
- Provides about 340 undecillion **unique** addresses



Domain Name Server

- The **domain name** is a human way to access IP addresses for devices and websites around the world
- When a domain name is entered in the browser, a request is made the **DNS**

IP Address	Domains
216.58.214.46	Google.com
217.174.159.195	Softuni.bg

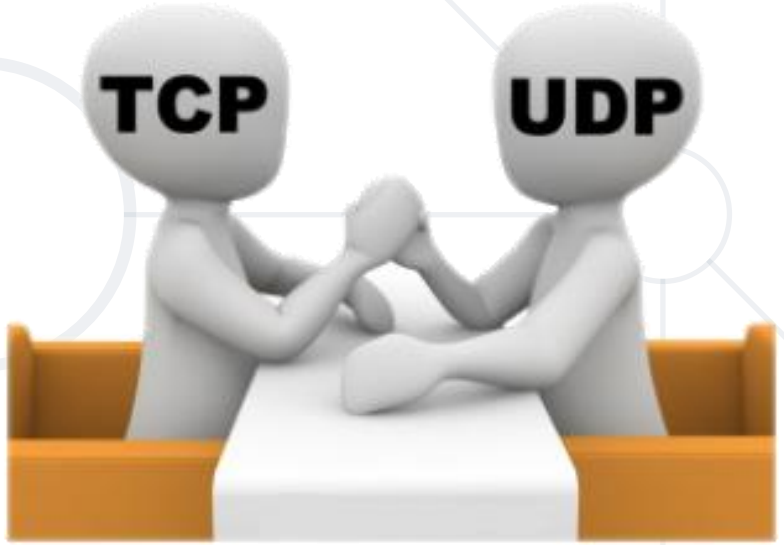
Transmission Control Protocol (TCP)

- Designed to **send packets** across the internet and **ensure the successful delivery** of data and messages over networks
- Uses a process, where it looks at **all the packets** in a message and **checks them**
- **TCP verifies** that all the packets are:
  - In the right order
  - Free of any issues

- After that it **certifies the data**, and the packets are **merged** to recreate the **original** file that was on the sender's device

TCP vs UDP

- **TCP** places **reliability** in a higher priority than speed
- For instances where reliability isn't as important, but **speed** is, there is another protocol called **UDP** (**User Datagram Protocol**)
- UDP doesn't do excessive reliability checks, but it can send information at a faster rate

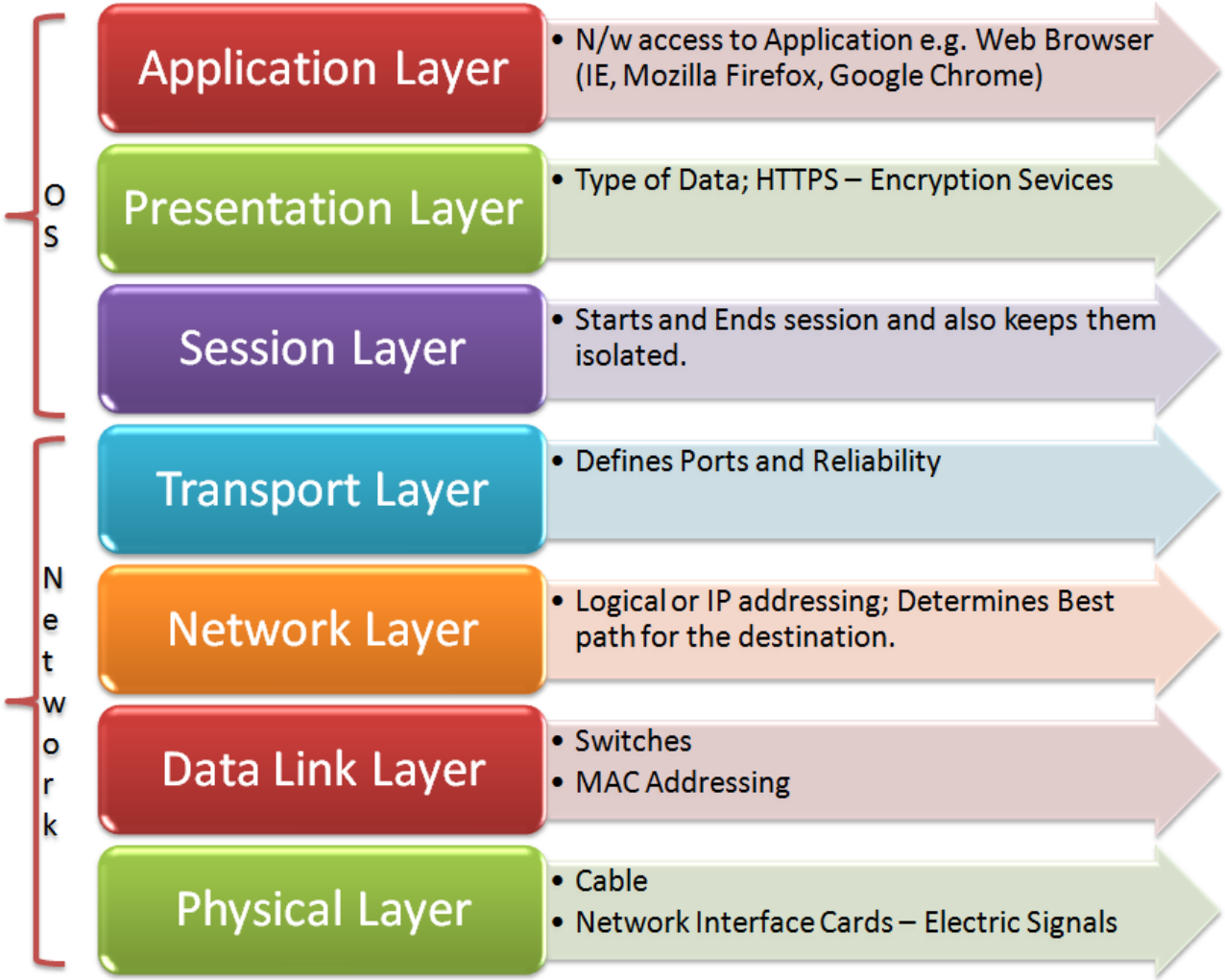


Open System Interconnect Model

- OSI Model consists of **7 layers**
  - Each layer serves the layer above it and in return, it is served by the layer below it
- Understanding each layer of the model helps us with:
  - **Troubleshooting**
  - **Communicating** better with technical and non-technical individuals about any system

OSI Layers

- OSI Model consists of 7 layers:



Example Protocols

HTTP, DNS, FTP, SMTP

TLS, SSL, compression

NetBIOS, PPTP, Sockets

TCP, UDP

IP, IPsec

ATM, Ethernet, MAC, LLC

USB, Bluetooth, 802.11a/b/g/n

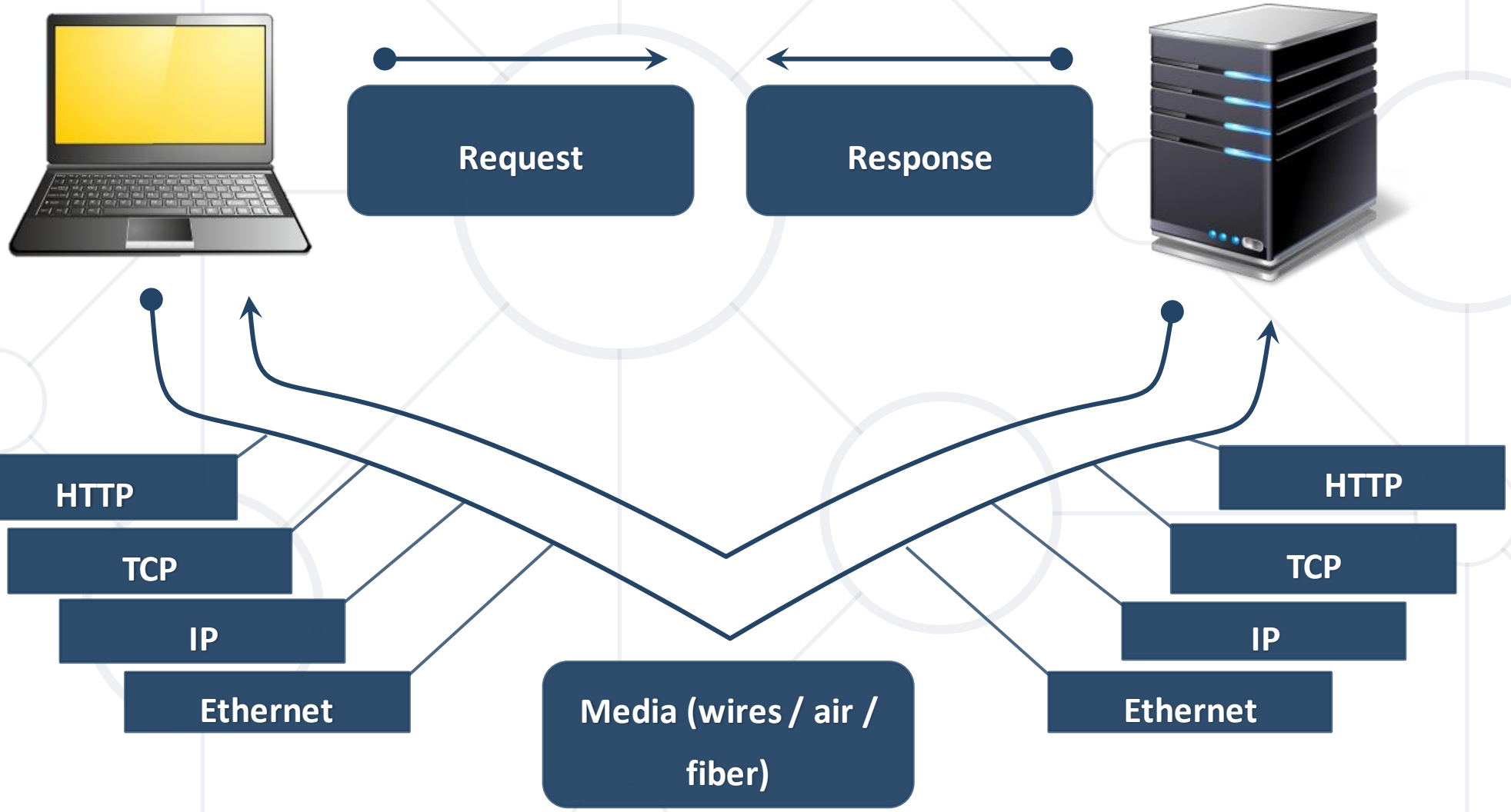
Basic Hardware Components

- **Cables** - transfer data from one device to another
- **Routers** - transfer data packets between different computer networks (operates on level 3 of OSI)
- **Repeaters**, **Hubs** and **Switches** - connect network devices together so that they can function as a single segment
- **Network Interface Card** (**NIC**) - computer component that connects the device to the network

HTTP Basics

Web Communication Explained

Hyper Text Transfer Protocol



HTTP Request Methods

Method	Description
POST	Create / store a resource
GET	Read / retrieve a resource
PUT	Update / modify a resource
DELETE	Delete / remove a resource

The four basic functions of persistence storage.

Other HTTP Methods
CONNECT
HEAD
OPTIONS
TRACE

HTTP Conversation: Example

- HTTP **request**:

```
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 headers

- HTTP **response**:

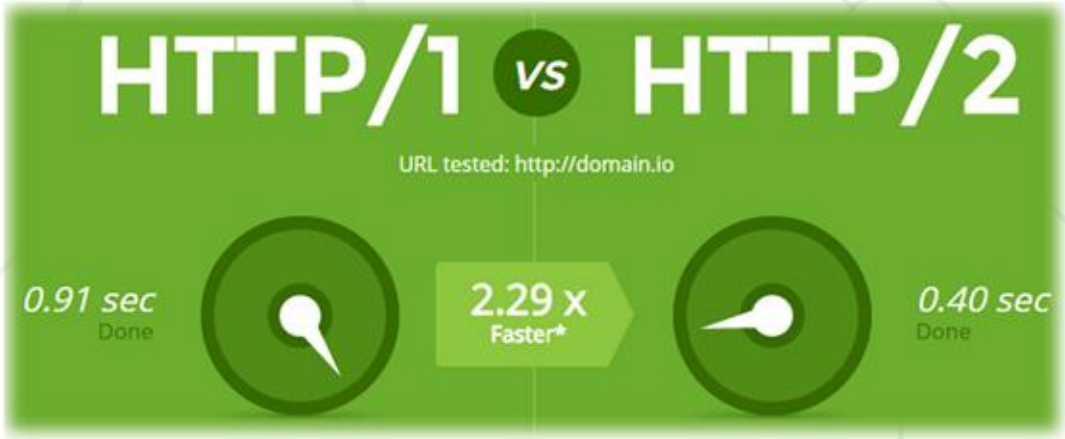
```
HTTP/1.1 200 OK
Date: Mon, 5 Jul 2010 13:09:03 GMT
Server: Microsoft-HTTPAPI/2.0
Last-Modified: Mon, 12 Jul 2014 15:33:23 GMT
Content-Length: 54
<CRLF>
<html><title>Hello</title>
Welcome to our site</html>
```

The empty line denotes the end of the response headers

What's HTTP/2.0

- Major revision of the **HTTP** network protocol used by the **World Wide Web**
  - Supported by most of the popular web browsers
- Fast and optimized, meets modern web usage requirements
- Completely Backwards-Compatible

- Almost **50%** of all the websites support **HTTP/2** (W3Techs statistics)



URL

Uniform Resource Locator

Uniform Resource Locator (URL)

- A **URL** is a reference to a web resource that specifies its location on a network and a mechanism for retrieving it
- A URL is a specific type of URI (**Uniform Resource Identifier**)

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-爱-SoftUni
```

  - URL-encoded string:

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

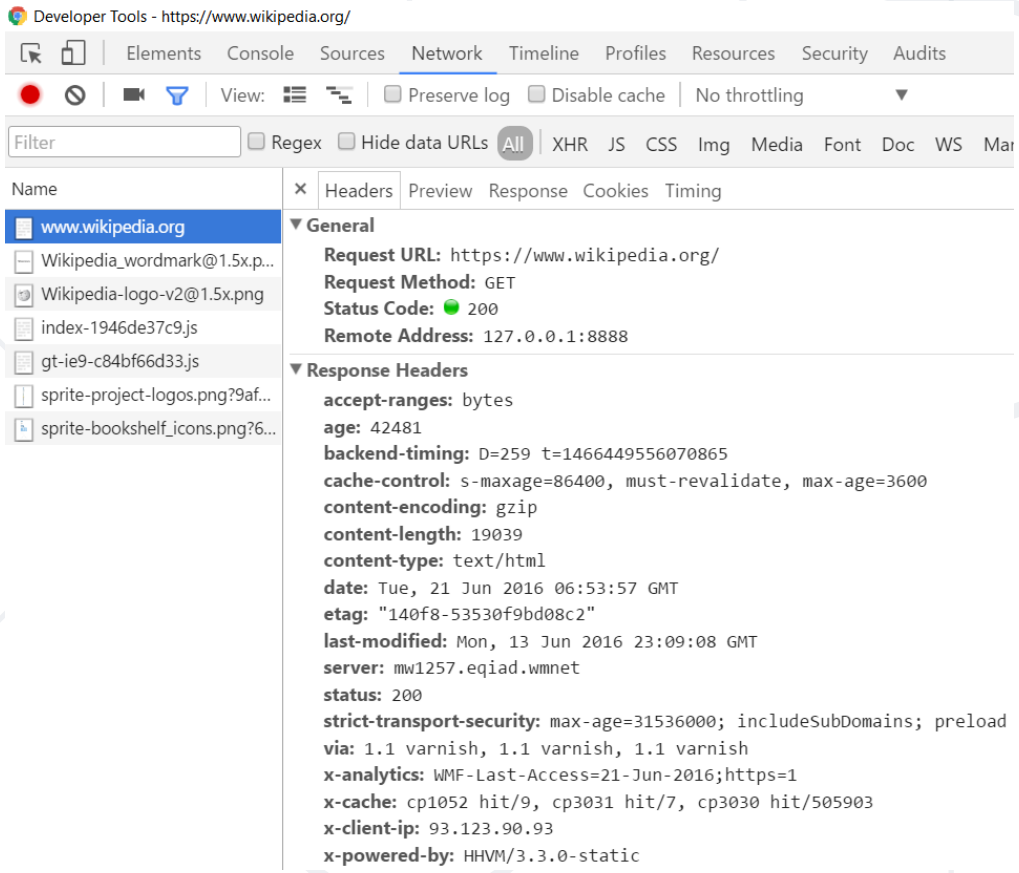
Char	URL Encoding
space	%20
щ	%D1%89
"	%22
#	%23
$	%24
%	%25
&	%26

Tools for Developers

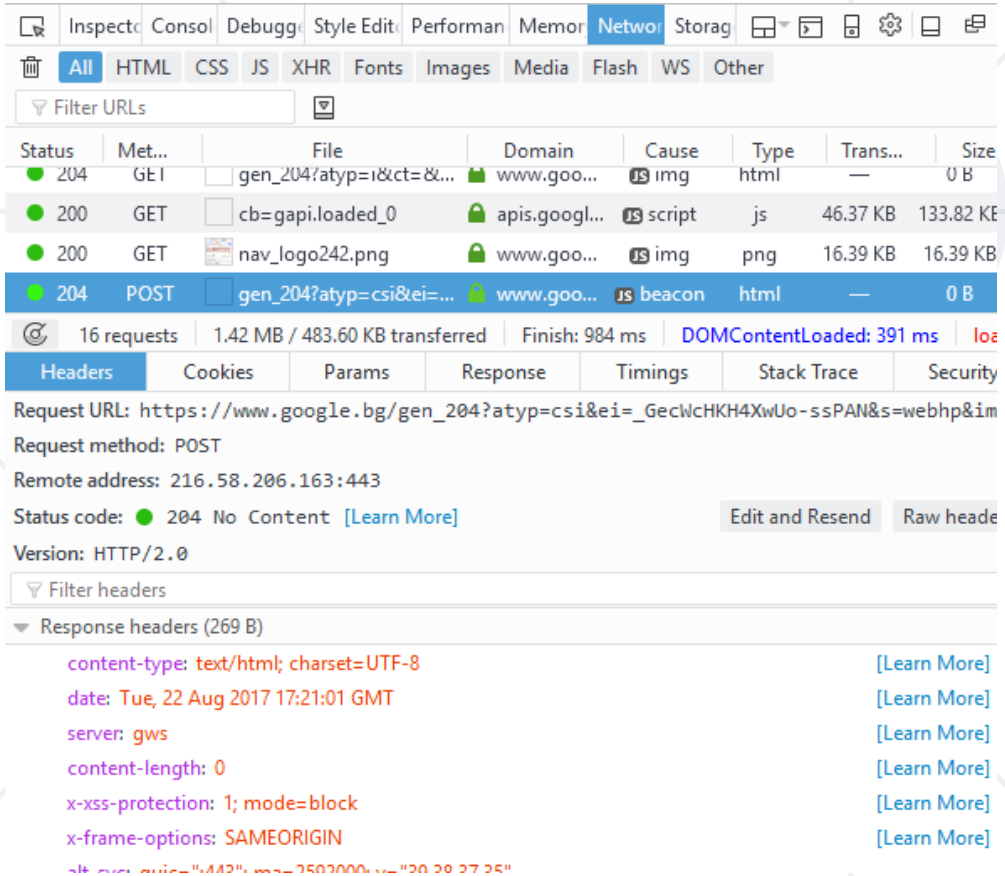
Dev Tools

Tools for Developers – Browser Dev Tools

Software University



Chrome Developer Tools



Mozilla Developer Tools

Tools for Developers – Browser Add-ons

Postman - Chrome

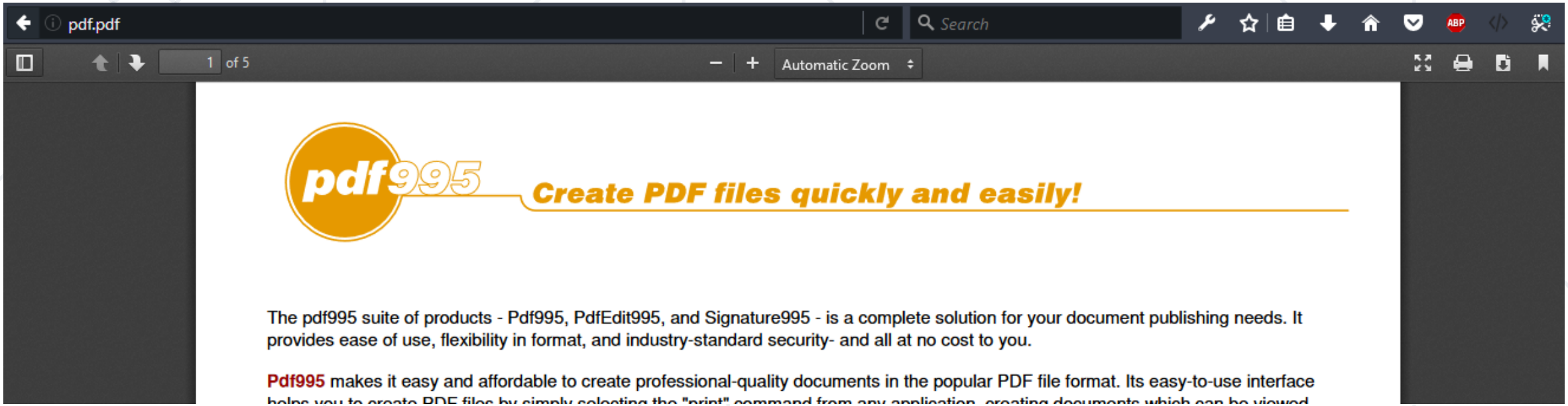
Rested - Firefox

Multi-Purpose Internet Mail Extensions

MIME and Media Types

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



Common MIME Media Types

Software University

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 Request / HTTP Respond

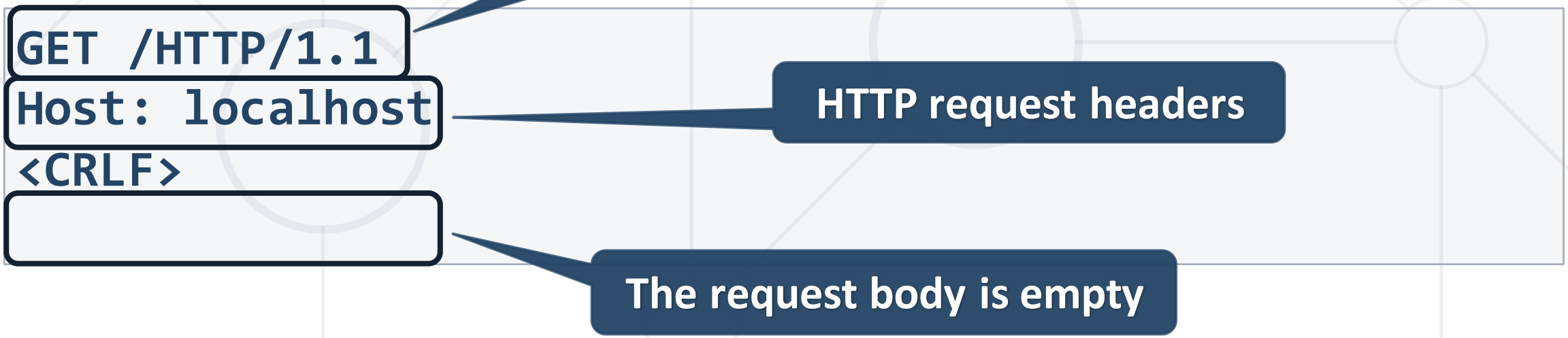
HTTP Request Message

- 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>
```

GET Request Method – Example

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



POST Request Method – Example

- The **POST** method transfers data in the HTTP body
- **POST** can send text and binary data e.g., upload files

```
POST /login 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 Message

- The **response message** sent by the HTTP server consists of:
  - HTTP response **status line**
    - Protocol version
    - Status code
    - Status text
  - 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>
<CRLF>
<response body - the requested resource>
```

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** from the Web server:

```
HTTP/1.1 200 OK
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 **status line**

HTTP response **headers**

HTTP response **body**

HTTP Response – Example

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

```
HTTP/1.1 404 Not Found
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>
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 status line

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
...
```

Summary

- Internet, Definitions of Internet
- Sending and Receiving Information
- What is HTTP
- What is URL
- Browser Tools for Developers
- What is MIME

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

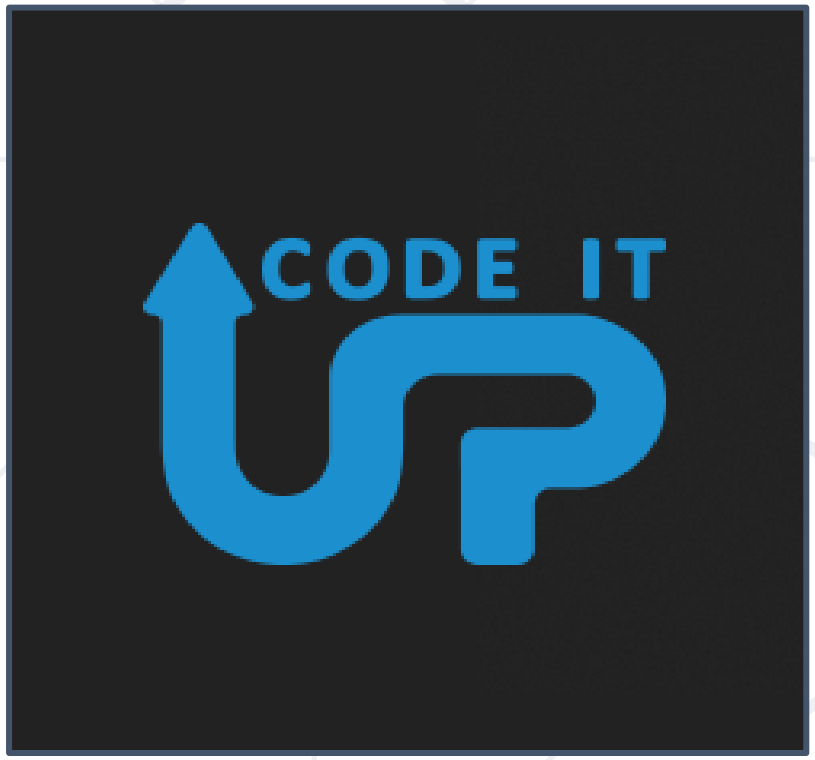
SoftUni Kids

SoftUni Diamond Partners

Software University

Educational Partners

Software University



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