

# Internet and HTTP



**SoftUni Team**  
**Technical Trainers**



**SoftUni**



**Software University**

<https://softuni.bg>

[sli.do](https://sli.do)

**#python-web**

# Table of Contents

1. Introduction to **Internet**
2. **HTTP** Basics
3. **URL**
4. **Tools** for Developers
5. **MIME**
6. **HTTP Request** and **HTTP Response**



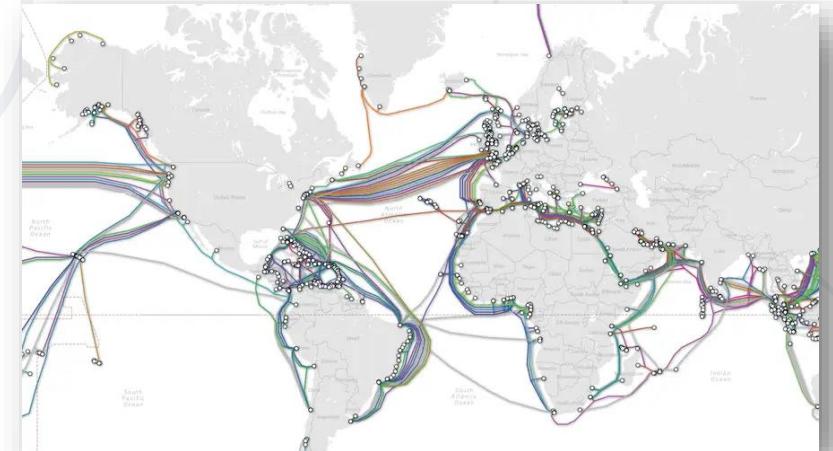


# Introduction

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** through **ISPs** (Internet Service Providers)



# 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



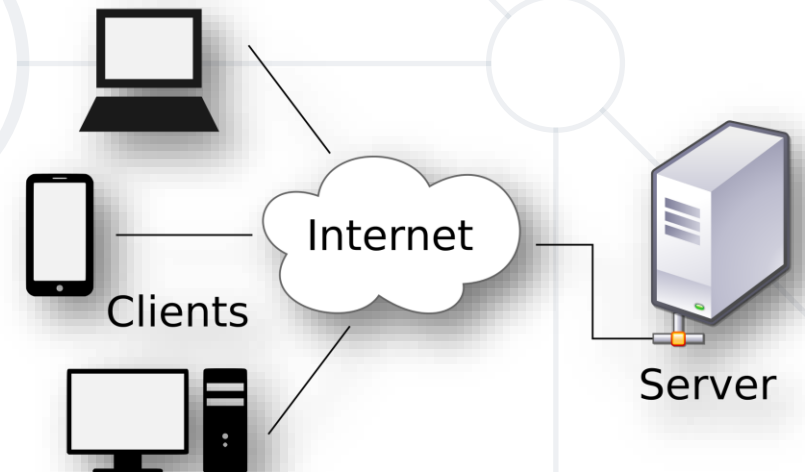


# **Important Definitions**



# 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, SMTP, 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 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



# 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 to the **DNS**



IP Address	Domains
216.58.214.46	Google.com
217.174.159.195	Softuni.bg

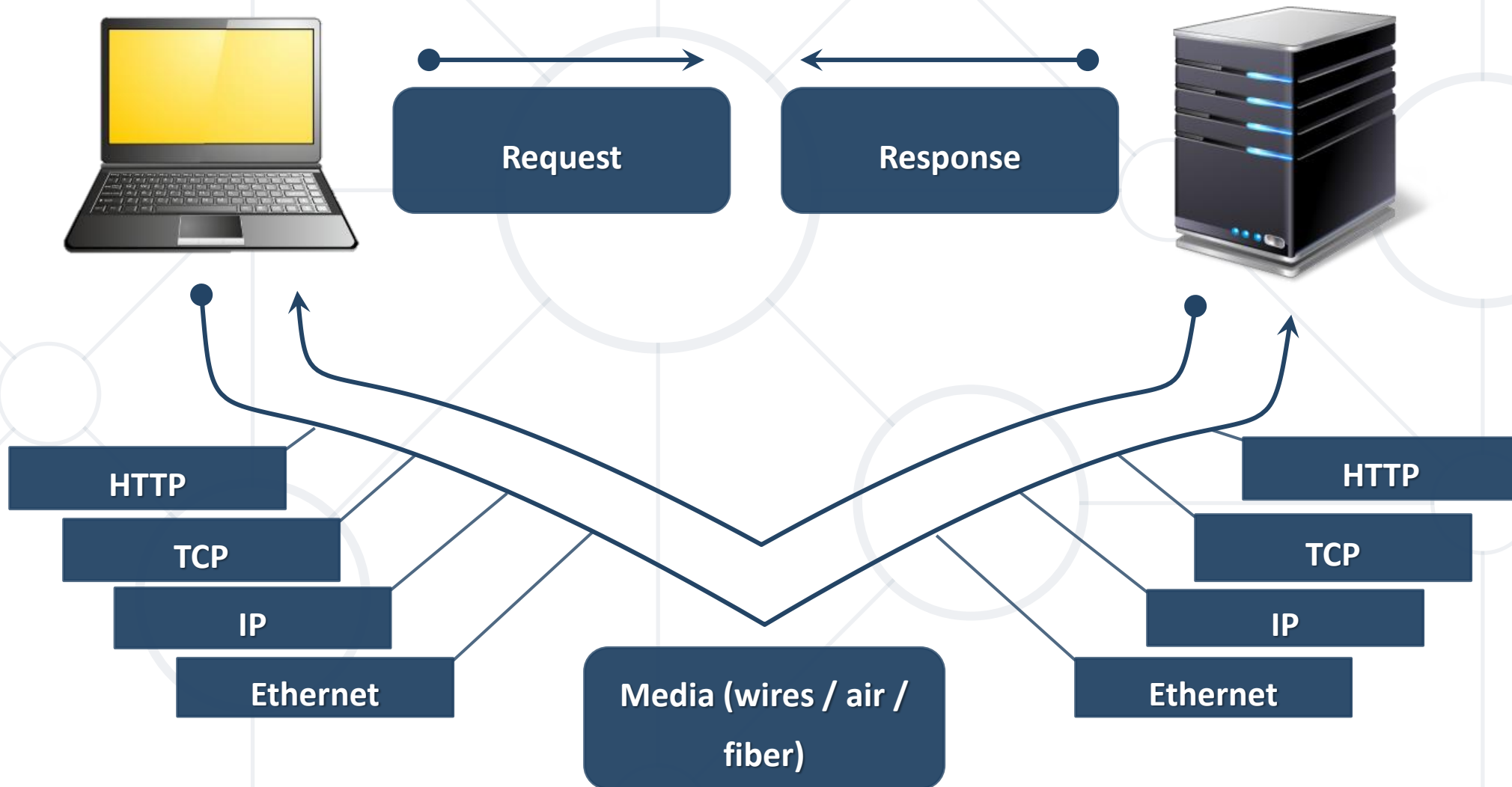
A background network diagram consisting of a central dark blue circle containing the text 'http://'. Surrounding this central circle are several smaller, light gray circles connected by thin gray lines, forming a web-like structure. The central circle is the largest and most prominent, while the surrounding circles vary in size and are connected in a non-uniform pattern, suggesting a decentralized network.

http://

# 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 persistent 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

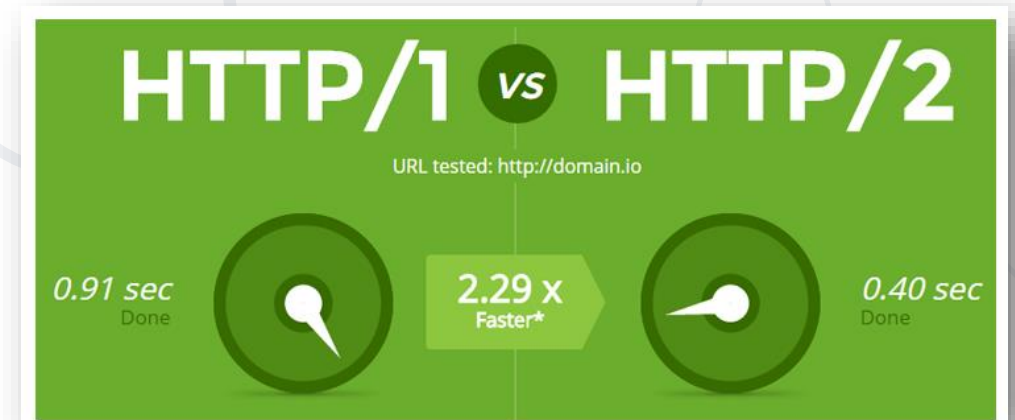
- HTTP **response**:

```
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**)

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



The diagram illustrates the components of the URL `http://localhost:8080/demo/index.html?id=27&lang=en#lecture`. Brackets are placed under the string to group it into six parts, each with a label below: **Protocol** (http), **Host** (localhost), **Port** (8080), **Path** (/demo/index.html), **Query String** (?id=27&lang=en), and **Fragment** (#lecture).

- 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`

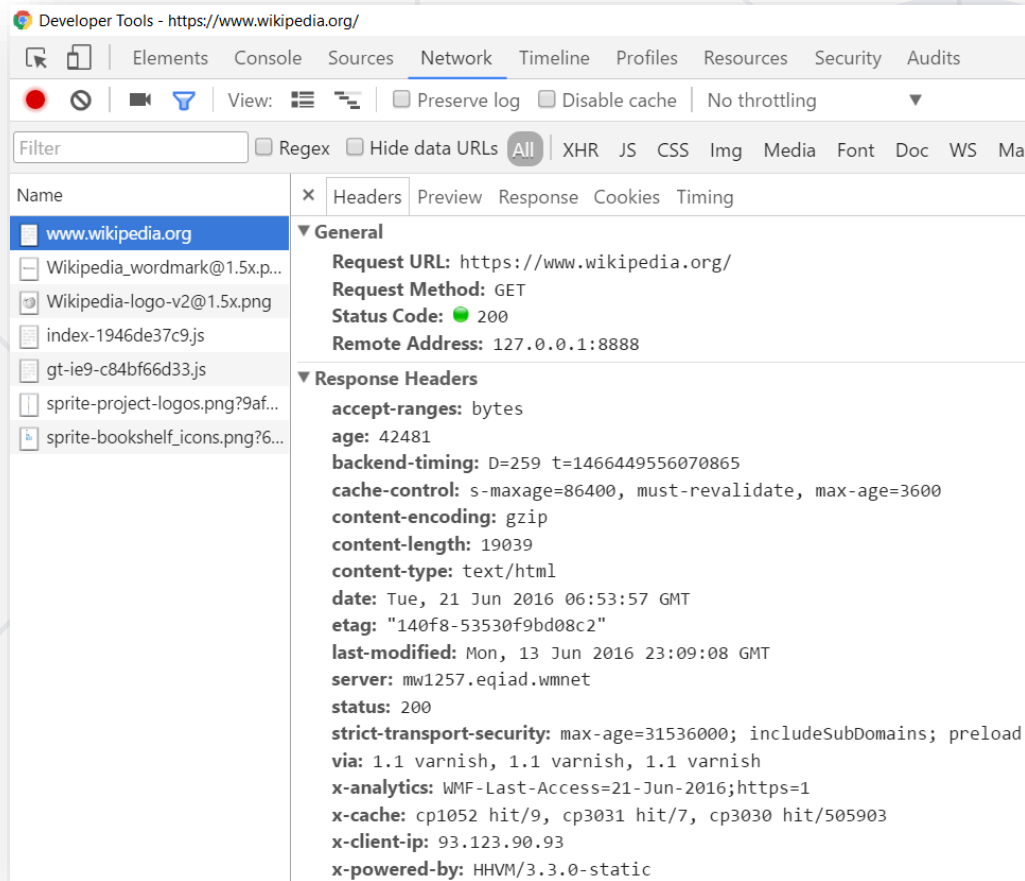
Char	URL Encoding
space	%20
щ	%D1%89
"	%22
#	%23
\$	%24
%	%25
&	%26



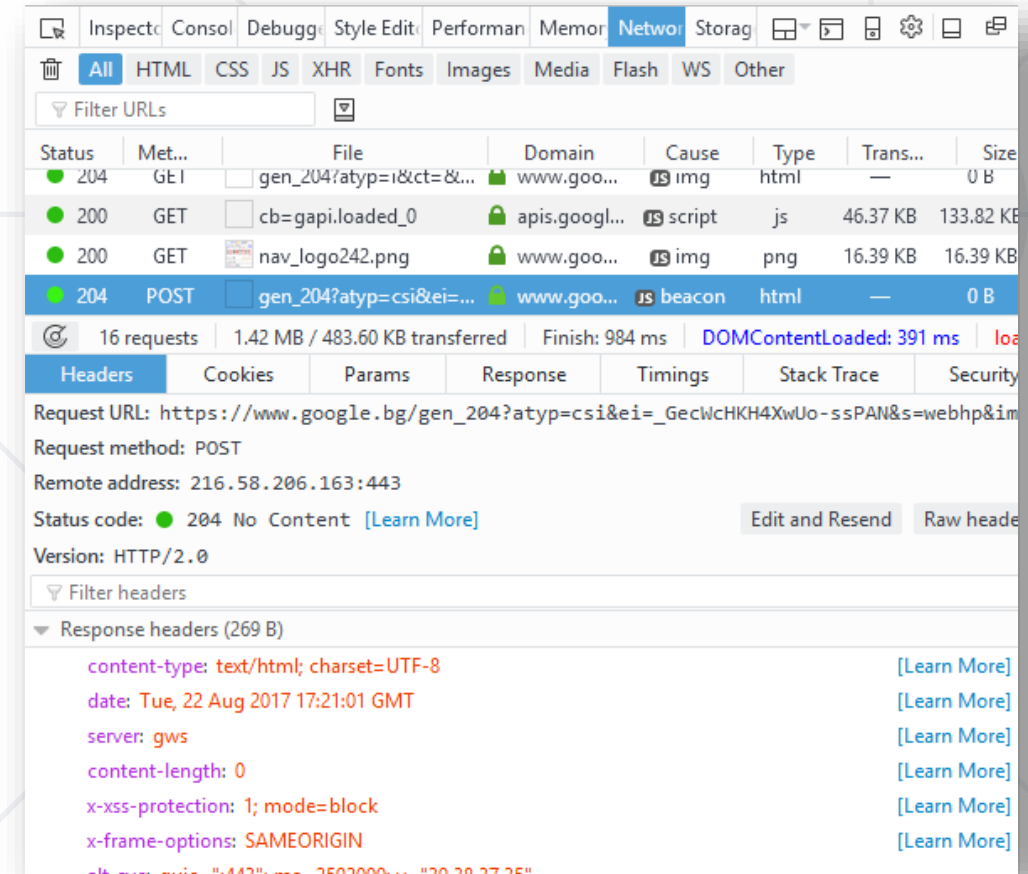
# Tools for Developers

Dev Tools

# Tools for Developers – Browser Dev Tools



Chrome Developer Tools



Mozilla Developer Tools





Postman – API platform



Rested – Firefox add-on

# Postman – Register an Account

## Build APIs together

Over 25 million developers use Postman. Get started by signing up or downloading the desktop app.

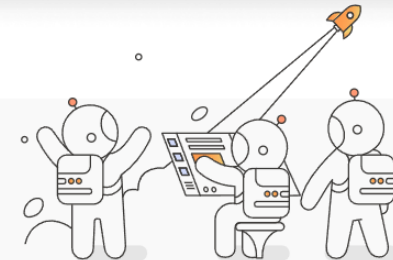
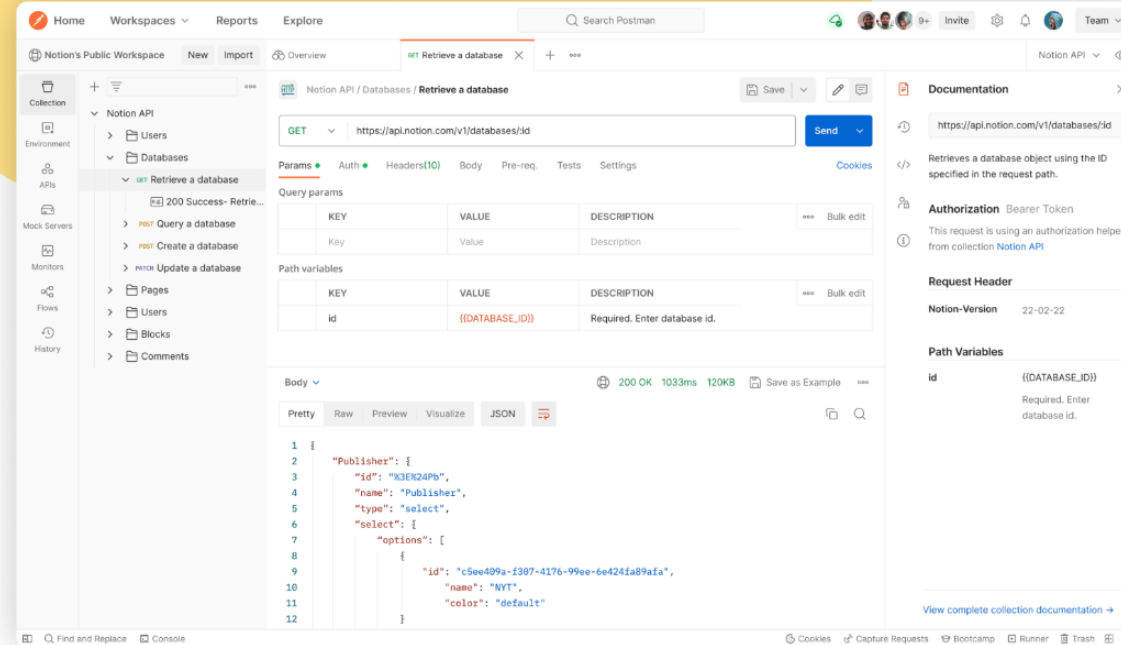
[Sign Up for Free](#)

Download the desktop app for

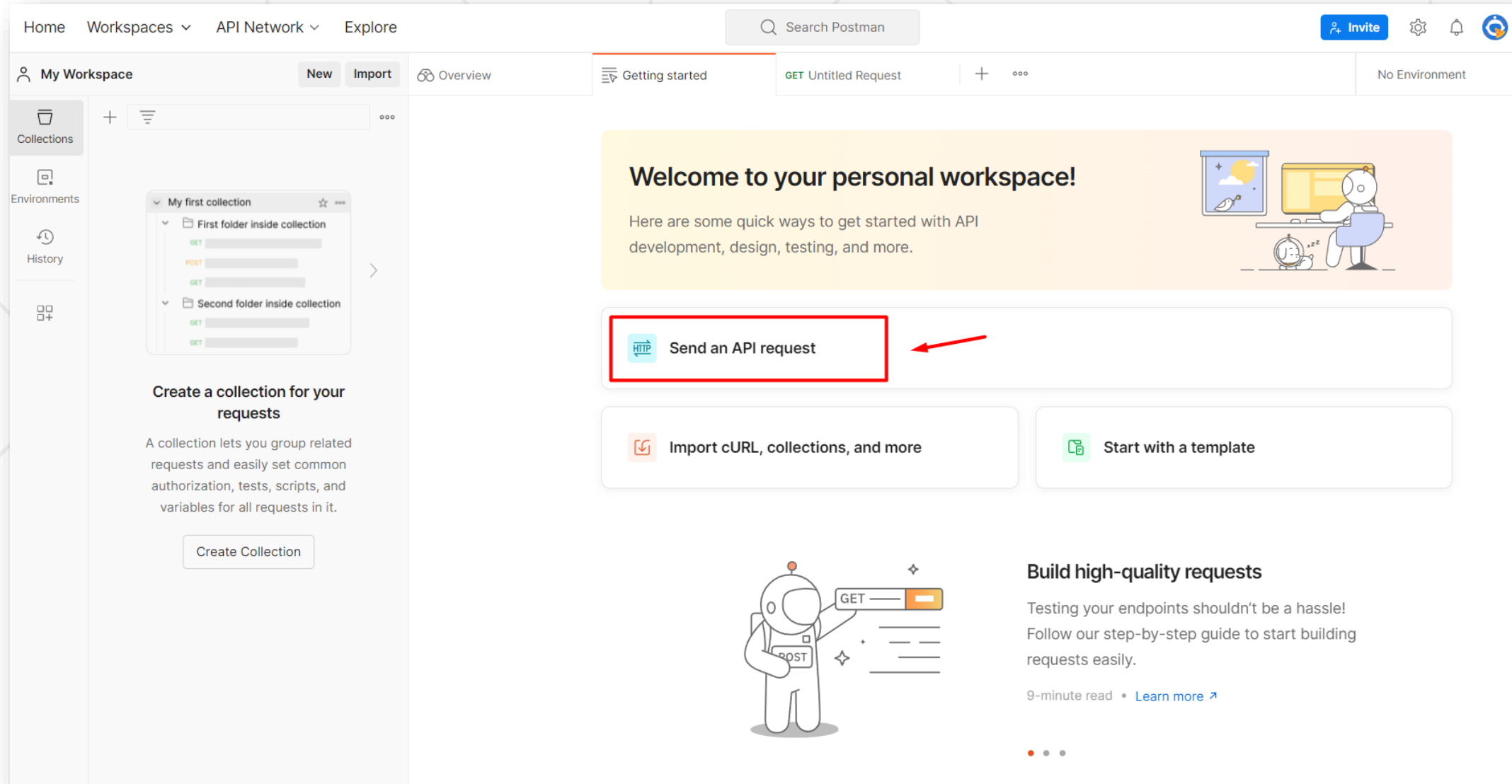


## What is Postman?

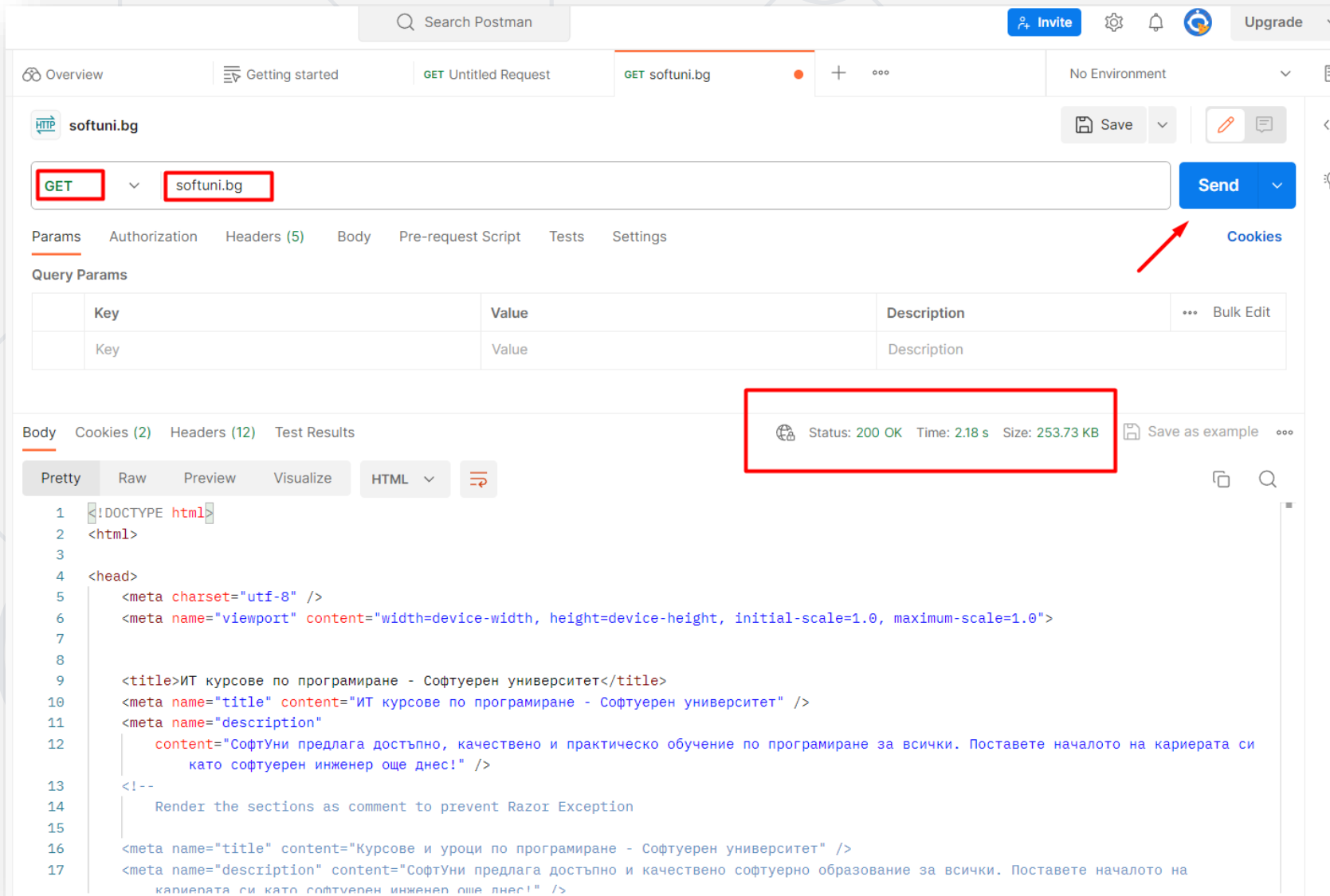
Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.



# Postman – Usage



# Postman – GET Request



The screenshot shows the Postman interface with a GET request configured for the URL `softuni.bg`. The `Send` button is highlighted with a red arrow. Below the request bar, the `Params` tab is selected, showing an empty table for query parameters. The `Body` tab is also visible, showing the response status `200 OK`, time `2.18 s`, and size `253.73 KB`. The response body is displayed in the `Body` tab, showing the HTML content of the page.

Query Params

Key	Value	Description
Key	Value	Description

Body

Status: 200 OK Time: 2.18 s Size: 253.73 KB

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8" />
6   <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, maximum-scale=1.0">
7
8
9   <title>ИТ курсове по програмиране - Софтуерен университет</title>
10  <meta name="title" content="ИТ курсове по програмиране - Софтуерен университет" />
11  <meta name="description"
12    content="СофтУни предлага достъпно, качествено и практическо обучение по програмиране за всички. Поставете началото на кариерата си като софтуерен инженер още днес!" />
13  <!--
14    Render the sections as comment to prevent Razor Exception
15  -->
16  <meta name="title" content="Курсове и уроци по програмиране - Софтуерен университет" />
17  <meta name="description" content="СофтУни предлага достъпно и качествено софтуерно образование за всички. Поставете началото на кариерата си като софтуерен инженер още днес!" />
```

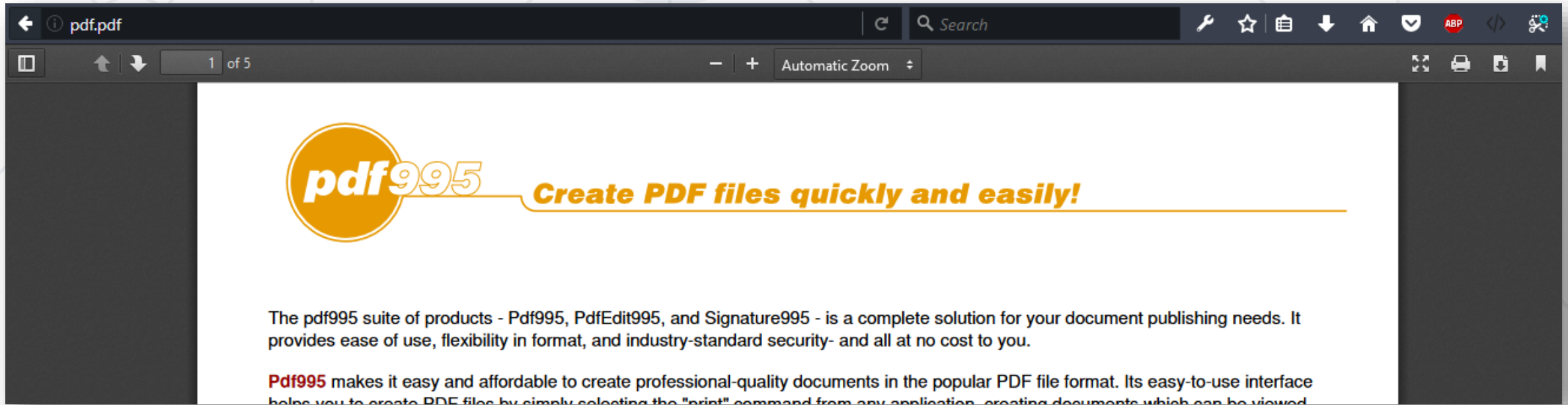


# **Multi-Purpose Internet Mail Extensions**

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

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 Response**



- 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>
```

GET /HTTP/1.1

Host: localhost

<CRLF>

HTTP request line

HTTP request headers

The request body is empty

# 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
```

HTTP request line

```
Host: localhost
```

```
Content-Length: 59
```

HTTP request headers

```
<CRLF>
```

```
username=mente&password=top*secret!
```

```
<CRLF>
```

The request body holds  
the submitted form data

- 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 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
```

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 – 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
```

HTTP response headers

```
<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>
```

The HTTP  
response body

- 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:

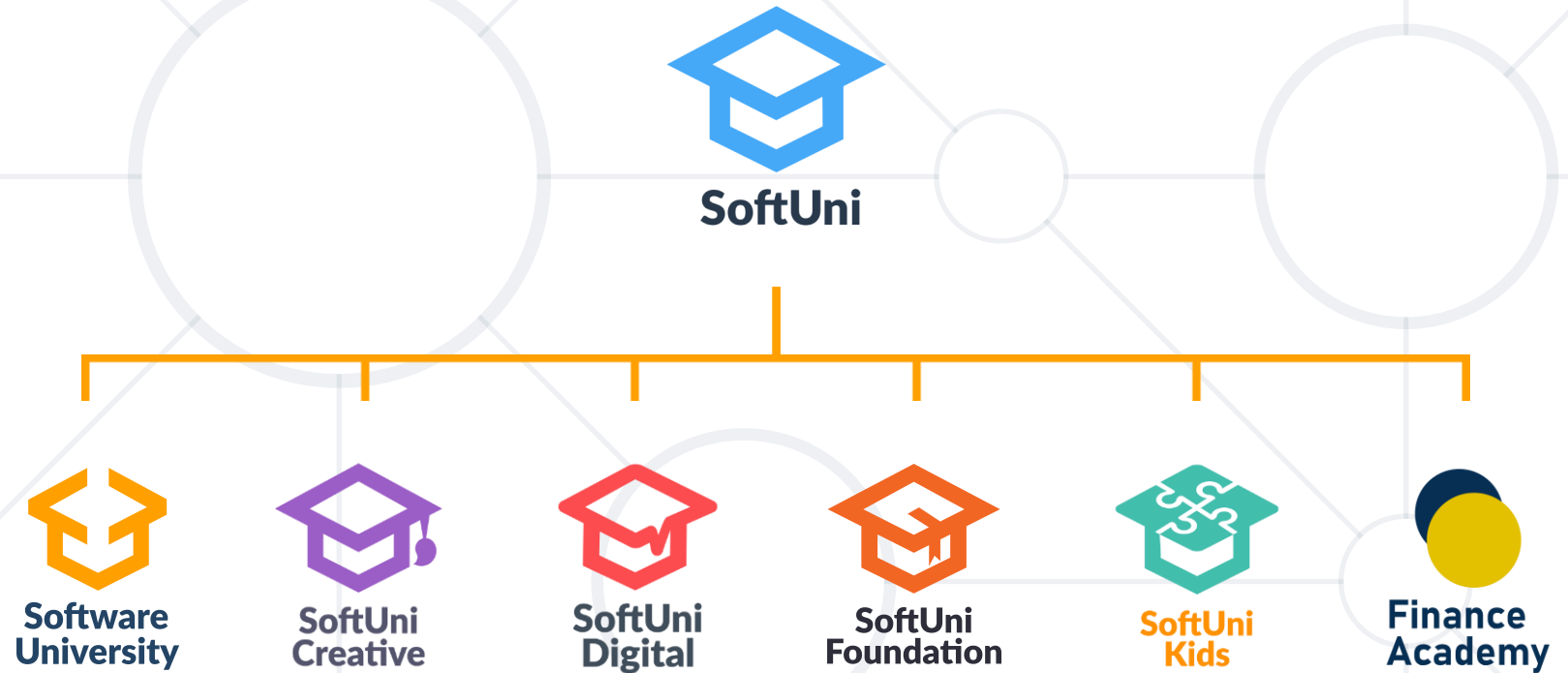
```
HTTP/1.1 301 Moved Permanently  
Location: http://softuni.bg  
...
```



- **Internet**, Definitions of Internet
- What is **HTTP**
- What is **URL**
- Browser **Tools** for **Developers**
- **Postman** API Platform
- What is **MIME**



# Questions?



# SoftUni Diamond Partners



- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)



- 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>

