1. Обяснете кодовете, които започват с 1хх, 2хх, 3хх, 4хх и 5хх

2. HTTP/2 - какво е multiplexing и server push?

3. Как работи TLS/SSL

4. Кавкви HTTP глаголи/операции познавате?

5. Опишете структурата на едно URL.

6. Каква е разликата между URI и URL?

7. Разлика между PUT/POST/PATCH.

8. Какво представлява модела POST-redirect-GET и защо се използва?

1. Обяснете кодовете, които започват с 1хх, 2хх, 3хх, 4хх и 5хх

1xx – Informational response, indicates that the request was received and understood. It is issued on a provisional basis while request processing continues. It alert to client to wait for a final response.

Информационен отговор(реакция), който показва че заявката е получена и е разбрана. Издава се временно докато обработката на заявката продължава. Сигнализира на клиента да изчака за крайния отговор(крайната реакция) .

2хх – This class of status codes indicates the action requested by the client was received, understood and accepted.

Индикира че заявката на клиента е получена, разбрана и приета.

3хх – Indicates the client must take additional action to complete the request. Many of these status codes are used in URL redirection.

Сигнализира на клиента че е нужно допълнително действие за да бъде завършена успешно заявката. Много от тези статуси се използват в пренасочване на URL.

4хх – This class of status code is intended for situations in which the error seems to have been caused by the client. Except when responding to a HEAD request, the server should include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method.

Този клас код на състоянието е предназначен за ситуации, в които изглежда, че грешката е била причинена от клиента. Освен когато отговаря на HEAD заявка, сървърът трябва да включва обект, съдържащ обяснение на ситуацията с грешка и дали тя е временно или постоянно състояние. Тези кодове на състоянието са приложими за всеки метод на заявка.

5xx – The server failed to fulfil a request. This status indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request.

Сървърът не успява да изпълни заявка. Това състояние показва случаи, в които сървърът знае, че е срещнал грешка или по друга причина не е в състояние да изпълни заявката.

2. HTTP/2 - какво е multiplexing и server push?

Multiplexing - method by which multiple analog or digital signals are combined into one signal over a shared medium. The art of handling multiple streams over a **single** TCP connection.

Метод, чрез който множество аналогови или цифрови сигнали се комбинират в един сигнал през споделен носител. Изкуството да се обработват множество потоци през една TCP връзка.

Server push - is the process of sending resources to clients, without them having to ask for it.

Allows an [HTTP/2](https://en.wikipedia.org/wiki/HTTP/2)-compliant server to send resources to a HTTP/2-compliant client before the client requests them. Server Push is a performance technique aimed at reducing latency by loading resources preemptively, even before the client knows they will be needed. In practice, Server Push frequently results in wasted bandwidth because the server rarely knows which resources are already loaded by the client and transmits the same resource multiple times.

HTTP/2 Server Push is not a notification mechanism from server to client. Instead, pushed resources are used by the client when it may have otherwise produced a request to get the resource anyway.

Процесът на изпращане на ресурси до клиенти, без да се налага те да искат това.

Позволява на HTTP/2-съвместим сървър да изпраща ресурси до HTTP/2-съвместим клиент, преди клиентът да ги поиска. Server Push е техника за производителност, насочена към намаляване на латентността чрез превантивно зареждане на ресурси, дори преди клиентът да разбере, че ще са необходими. На практика Server Push често води до загуба на честотна лента, тъй като сървърът рядко знае кои ресурси вече са заредени от клиента и предава един и същ ресурс няколко пъти.

HTTP/2 Server Push не е механизъм за уведомяване от сървър към клиент. Вместо това, изтласканите ресурси се използват от клиента, когато той така или иначе може да е произвел заявка за получаване на ресурса.

3. Как работи TLS/SSL

Transport Layer Security (TLS) - TLS is a cryptographic protocol that provides end-to-end security of data sent between applications over the Internet. It is mostly familiar to users through its use in secure web browsing, and in particular the padlock icon that appears in web browsers when a secure session is established. However, it can and indeed should also be used for other applications such as e-mail, file transfers, video/audioconferencing, instant messaging and voice-over-IP, as well as Internet services such as DNS and NTP.

Secure Socket Layers (SSL) – evolved to TLS.

TLS (на [английски](https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8_%D0%B5%D0%B7%D0%B8%D0%BA): Transport Layer Security) и неговият предшественик [SSL](https://bg.wikipedia.org/wiki/SSL) (на [английски](https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8_%D0%B5%D0%B7%D0%B8%D0%BA): Secure Sockets Layer) са криптографски  [протоколи](https://bg.wikipedia.org/wiki/%D0%9C%D1%80%D0%B5%D0%B6%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) , които осигуряват сигурност на комуникацията по [интернет](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). TLS и SSL криптирането са сегменти на мрежови връзки в [приложния слой](https://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9_%D0%BD%D0%B0_OSI_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B0) (над [транспортния слой](https://bg.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9_%D0%BD%D0%B0_OSI_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B0)), използвайки [асиметрична](https://bg.wikipedia.org/w/index.php?title=%D0%90%D1%81%D0%B8%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F&action=edit&redlink=1) [криптография](https://bg.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) за личния код на [автентикация](https://bg.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8E%D1%82%D1%8A%D1%80%D0%BD%D0%B0_%D1%81%D0%B8%D0%B3%D1%83%D1%80%D0%BD%D0%BE%D1%81%D1%82)) на съобщението, което увеличава надеждността на съобщението.

Няколко версии на протоколите се използват широко при сърфиране в интернет, [електронна поща](https://bg.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BF%D0%BE%D1%89%D0%B0), [интернет факс](https://bg.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D1%84%D0%B0%D0%BA%D1%81&action=edit&redlink=1), [чат](https://bg.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B7%D0%B0%D0%B1%D0%B0%D0%B2%D0%BD%D0%B8_%D1%81%D1%8A%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F) и [voice-over-IP](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D1%82%D0%B5%D0%BB%D0%B5%D1%84%D0%BE%D0%BD%D0%B8%D1%8F) (VoIP).

TLS е IETF standards track протокол, за последен път е актуализиран в [RFC 5246](https://tools.ietf.org/html/rfc5246), и се основава на по-ранните SSL спецификации, разработен от Netscape Communications.

|  |
| --- |
| [**OSI модел**](https://bg.wikipedia.org/wiki/OSI_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB) |
| **7. [Приложен слой](https://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9" \o "Приложен слой)** |
| [NNTP](https://bg.wikipedia.org/wiki/NNTP) • [SIP](https://bg.wikipedia.org/wiki/Session_Initiation_Protocol) • [SSI](https://bg.wikipedia.org/w/index.php?title=Simple_Sensor_Interface_protocol&action=edit&redlink=1) • [DNS](https://bg.wikipedia.org/wiki/Domain_Name_System) • [FTP](https://bg.wikipedia.org/wiki/FTP) • [Gopher](https://bg.wikipedia.org/w/index.php?title=Gopher_(protocol)&action=edit&redlink=1) • [HTTP](https://bg.wikipedia.org/wiki/HTTP) • [NFS](https://bg.wikipedia.org/w/index.php?title=Network_File_System_(protocol)&action=edit&redlink=1) • [NTP](https://bg.wikipedia.org/wiki/NTP) • [SMPP](https://bg.wikipedia.org/w/index.php?title=SMPP&action=edit&redlink=1) • [SMTP](https://bg.wikipedia.org/wiki/SMTP) • [DHCP](https://bg.wikipedia.org/wiki/DHCP) • [SNMP](https://bg.wikipedia.org/wiki/SNMP) • [SSH](https://bg.wikipedia.org/wiki/SSH) • [Telnet](https://bg.wikipedia.org/wiki/Telnet) • [Netconf](https://bg.wikipedia.org/w/index.php?title=Netconf&action=edit&redlink=1) • [други...](https://bg.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D1%8F:%D0%9C%D1%80%D0%B5%D0%B6%D0%BE%D0%B2%D0%B8_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%B8) |
| **6. [Представителен слой](https://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9" \o "Представителен слой)** |
| [MIME](https://bg.wikipedia.org/wiki/MIME) • [XDR](https://bg.wikipedia.org/w/index.php?title=External_Data_Representation&action=edit&redlink=1) • [TLS](https://bg.wikipedia.org/wiki/Transport_Layer_Security) • [SSL](https://bg.wikipedia.org/wiki/SSL) |
| **5.  [Сесиен слой](https://bg.wikipedia.org/wiki/%D0%A1%D0%B5%D1%81%D0%B8%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9" \o "Сесиен слой)** |
| [Named Pipes](https://bg.wikipedia.org/w/index.php?title=Named_Pipes&action=edit&redlink=1) •  [NetBIOS](https://bg.wikipedia.org/w/index.php?title=NetBIOS&action=edit&redlink=1)  • [SAP](https://bg.wikipedia.org/w/index.php?title=Session_Announcement_Protocol&action=edit&redlink=1) • [L2TP](https://bg.wikipedia.org/w/index.php?title=Layer_2_tunneling_Protocol&action=edit&redlink=1) • [PPTP](https://bg.wikipedia.org/w/index.php?title=PPTP&action=edit&redlink=1) |
| **4. [Транспортен слой](https://bg.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9" \o "Транспортен слой)** |
| [TCP](https://bg.wikipedia.org/wiki/Transmission_Control_Protocol) • [UDP](https://bg.wikipedia.org/wiki/User_Datagram_Protocol) • [SCTP](https://bg.wikipedia.org/w/index.php?title=Stream_Control_Transport_protocol&action=edit&redlink=1) • [DCCP](https://bg.wikipedia.org/w/index.php?title=Datagram_Congestion_Control_Protocol&action=edit&redlink=1) • [SPX](https://bg.wikipedia.org/w/index.php?title=IPX/SPX&action=edit&redlink=1) |
| **3. [Мрежов слой](https://bg.wikipedia.org/wiki/%D0%9C%D1%80%D0%B5%D0%B6%D0%BE%D0%B2_%D1%81%D0%BB%D0%BE%D0%B9" \o "Мрежов слой)** |
| [IP](https://bg.wikipedia.org/wiki/Internet_Protocol) ([IPv4](https://bg.wikipedia.org/wiki/IPv4), [IPv6](https://bg.wikipedia.org/wiki/IPv6)) • [ICMP](https://bg.wikipedia.org/wiki/Internet_Control_Message_Protocol) • [IPsec](https://bg.wikipedia.org/wiki/IPsec) • [IGMP](https://bg.wikipedia.org/wiki/IGMP) • [IPX](https://bg.wikipedia.org/w/index.php?title=Internetwork_Packet_Exchange&action=edit&redlink=1) • [AppleTalk](https://bg.wikipedia.org/w/index.php?title=AppleTalk&action=edit&redlink=1) • [IS-IS](https://bg.wikipedia.org/w/index.php?title=Intermediate_System_to_Intermediate_System&action=edit&redlink=1) • [OSPF](https://bg.wikipedia.org/wiki/OSPF) • [RIP](https://bg.wikipedia.org/wiki/RIP) • [BGP](https://bg.wikipedia.org/wiki/BGP) • [IGRP](https://bg.wikipedia.org/w/index.php?title=IGRP&action=edit&redlink=1) • [EIGRP](https://bg.wikipedia.org/w/index.php?title=EIGRP&action=edit&redlink=1) |
| **2. [Канален слой](https://bg.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB%D0%B5%D0%BD_%D1%81%D0%BB%D0%BE%D0%B9" \o "Канален слой)** |
| [MAC адрес](https://bg.wikipedia.org/wiki/MAC_%D0%B0%D0%B4%D1%80%D0%B5%D1%81) • [ATM](https://bg.wikipedia.org/w/index.php?title=Asynchronous_Transfer_Mode&action=edit&redlink=1) • [SDLC](https://bg.wikipedia.org/w/index.php?title=Synchronous_Data_Link_Control&action=edit&redlink=1) • [HDLC](https://bg.wikipedia.org/w/index.php?title=HDLC&action=edit&redlink=1) • [ARP](https://bg.wikipedia.org/wiki/Address_Resolution_Protocol) • [CSLIP](https://bg.wikipedia.org/w/index.php?title=CSLIP&action=edit&redlink=1) • [SLIP](https://bg.wikipedia.org/w/index.php?title=SLIP&action=edit&redlink=1) • [PLIP](https://bg.wikipedia.org/w/index.php?title=PLIP&action=edit&redlink=1) • [IEEE 802.3](https://bg.wikipedia.org/w/index.php?title=IEEE_802.3&action=edit&redlink=1) • [Frame Relay](https://bg.wikipedia.org/w/index.php?title=Frame_Relay&action=edit&redlink=1) • [ITU-T G.hn DLL](https://bg.wikipedia.org/w/index.php?title=G.hn&action=edit&redlink=1) • [PPP](https://bg.wikipedia.org/w/index.php?title=Point-to-Point_Protocol&action=edit&redlink=1) • [X.25](https://bg.wikipedia.org/w/index.php?title=X.25&action=edit&redlink=1) • [Суич](https://bg.wikipedia.org/wiki/%D0%A1%D1%83%D0%B8%D1%87) |
| **1. [Физически слой](https://bg.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8_%D1%81%D0%BB%D0%BE%D0%B9" \o "Физически слой)** |
| [EIA/TIA-232](https://bg.wikipedia.org/w/index.php?title=EIA/TIA-232&action=edit&redlink=1) • [EIA/TIA-449](https://bg.wikipedia.org/w/index.php?title=EIA/TIA-449&action=edit&redlink=1) • [ITU-T V-Series](https://bg.wikipedia.org/w/index.php?title=ITU-T_V-Series_Recommendations&action=edit&redlink=1) • [I.430](https://bg.wikipedia.org/w/index.php?title=I.430&action=edit&redlink=1) • [I.431](https://bg.wikipedia.org/w/index.php?title=I.431&action=edit&redlink=1) • [POTS](https://bg.wikipedia.org/w/index.php?title=Plain_old_telephone_service&action=edit&redlink=1) • [PDH](https://bg.wikipedia.org/w/index.php?title=Plesiochronous_Digital_Hierarchy&action=edit&redlink=1) • [SONET/SDH](https://bg.wikipedia.org/w/index.php?title=SONET/SDH&action=edit&redlink=1) • [PON](https://bg.wikipedia.org/w/index.php?title=Passive_optical_network&action=edit&redlink=1) • [OTN](https://bg.wikipedia.org/w/index.php?title=Optical_Transport_Network&action=edit&redlink=1) • [DSL](https://bg.wikipedia.org/wiki/DSL) • [IEEE 802.3](https://bg.wikipedia.org/w/index.php?title=Ethernet_physical_layer&action=edit&redlink=1) • [IEEE 802.11](https://bg.wikipedia.org/wiki/IEEE_802.11) • [IEEE 802.15](https://bg.wikipedia.org/w/index.php?title=IEEE_802.15&action=edit&redlink=1) • [IEEE 802.16](https://bg.wikipedia.org/w/index.php?title=IEEE_802.16&action=edit&redlink=1) • [IEEE 1394](https://bg.wikipedia.org/wiki/IEEE_1394) • [ITU-T G.hn PHY](https://bg.wikipedia.org/w/index.php?title=G.hn&action=edit&redlink=1) • [USB](https://bg.wikipedia.org/wiki/USB) • [Bluetooth](https://bg.wikipedia.org/wiki/Bluetooth) • [Хъб](https://bg.wikipedia.org/wiki/%D0%A5%D1%8A%D0%B1) |

4. Кавкви HTTP глаголи/операции познавате?

GET – Retrieve / load a resource

Получава / зарежда ресурси

POST – Create / store a resource

Създава / съхранява ресурси

PUT – Update a resource

Актуализира ресурси

DELETE – Remove a resource

Премахва / изтрива ресурси

5. Опишете структурата на едно URL.

URL – Uniform Resource Locator

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

**Protocol**

**Host**

**Port**

**Path**

**Query String**

**Fragment**

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

6. Каква е разликата между URI и URL?

Uniform Resource Identifier (URI) –

Similar to URL, URI (Uniform Resource Identifier) is also a string of characters that identifies a resource on the web either by using location, name or both. It allows uniform identification of the resources. A URI is additionally grouped as a locator, a name or both which suggests it can describe a URL, URN or both. The term identifier within the URI refers to the prominence of the resources, despite the technique used.

The former category in URI is URL, during which a protocol is employed to specify the accessing method of the resource and resource name is additionally laid out in the URL. A URL may be a non-persistent sort of the URI. A URN is required to exist globally unique and features a global scope.

|  |  |
| --- | --- |
| URL | URI |
| URL is used to describe the identity of an item. | URI provides a technique for defining the identity of an item. |
| URL links a web page, a component of a web page or a program on a web page with the help of accessing methods like protocols. | URI is used to distinguish one resource from other regardless of the method used. |
| URL provides the details about what type of protocol is to be used. | URI doesn’t contains the protocol specification. |
| URL is a type of URI. | URI is the superset of URL. |

7. Разлика между PUT/POST/PATCH

PUT is another HTTP method used to create a new resource at a specified URI or to update an existing resource. Although PUT can be used to create a resource, it is most often used to update resource. To create a new resource, using PUT we need to know the exact URI where the data needs to be put. If incase there is data in the specified URI, the entire data is overwritten which is update.

POST is a HTTP method used to create a new resource in a collection of resources. POST method should ideally be used only to create new resources. Since REST doesn’t have a standard set of rules, some APIs use POST to update a resource as well. This is better to avoid . Use POST only to create a resource.

PATCH is another HTTP method which is used to update a resource with partial data . Unlike PUT, PATCH does not need the full payload to update a resource. For example if a resource has 100 fields, using PATCH would be a better option than PUT as PUT requires all 100 fields to be sent again to update a resource.

Resource: <https://programmerspub.com/blog/general/difference-between-post-vs-put-vs-patch>

8. Какво представлява модела POST- Redirect -GET (PRG)и защо се използва?

PRG is one of many design patterns used in web development. It is used to prevent the resubmission of a form caused by reloading the same web page after submitting the form. It removes redundancy of content to strengthen the SEO and makes the website user friendly.  
   It is used by large, trusted online shops and other robust websites which are intended to be user friendly.

**Problem:**

When we try to submit a web form then a HTTP POST request is sent to the server. The server process the request and send response to the client with response code 2XX. When the client try to refresh/reload the web page, he/she unintentionally sends another HTTP POST request to the server with the same data as just before. This may cause undesired results, such as duplicate web purchases.

**Solution :**

To avoid this problem many web developer use the *POST/REDIRECT/GET* pattern, instead of returning a web page directly, the *POST*  returns a redirect to another web page or same depending on the requirements.

Resources: <https://www.geeksforgeeks.org/post-redirect-get-prg-design-pattern/>