

Table of contents:

[1.About Me](#)

[2.HTTP Introduction](#)

[3.TCP/IP](#)

[a\)TCP/IP](#)

[b\)TCP/IP – LAN](#)

[c\)TCP/IP – MAN](#)

[d\) TCP/IP – WAN](#)

[e\) TCP/IP – router, server](#)

[f\)TCP/IP – application server](#)

[4. DNS – slide](#)

[5. DNS – Tracert – demo](#)

[7. HTTP methods](#)

[8. HTTP codes](#)

[9. Headers](#)

[10. Client-to-server communications](#)

[11. Request](#)

[12. Response](#)

[13. Rest, JSON, HATEOAS](#)

[a\)REST](#)

[b\) JSON](#)

[c\) HATEOAS](#)

[14. Redirection, apache rewrite](#)

[Redirection](#)

[Apache rewrite](#)

[15. HTTP vs HTTPS](#)

[16. cURL – installation – Windows](#)

[17. cURL – demo – windows](#)

[18. Wget – installation, demo – windows](#)

[Wget – installation](#)

[Windows – not in Git](#)

[b\) Wget – demo](#)

[19. Ping – demo – windows](#)

[20. SSH – installation – Windows](#)

[21. SSH – demo – Windows](#)

[22. Live HTTP headers – installation, demo – chrome](#)

[23. Postman – installation – Windows](#)

[24. Postman – demo – Windows](#)

1.About Me

Hi! Welcome to the Software Development Academy. My name is Marcin. I have been a programmer for 8 years and I am a programming trainer in the SDA. Today you will learn the basics of HTTP. Please take a look at the prepared materials and complete the tasks.

2.HTTP Introduction

In this series of videos you will get to know the HTTP basics. We shall start with the fact that it is the Internet where we use HTTP, then you will learn about the basic concepts of this protocol, and finally you will get to know some useful tools. I hope this material will provide you with a huge portion of knowledge about the HTTP basics and encourage you to further explore this protocol.

3.TCP/IP

a)TCP/IP

TCP/IP is a way of communication between devices connected by network cables or by means of wireless solutions.

Within the network that uses TCP/IP, the devices have an IP address such as 192.168.0.2. Thanks to the IP address, it is easy to determine which computer should receive which information. For a device to receive an IP address, it must have a MAC address. The MAC address is assigned by the manufacturer to network adapters that are used to connect the device to wired and wireless networks.

Wired and wireless networks are built based on TCP/IP. From the next videos you will learn about three types of networks.

b)TCP/IP – LAN

LAN is one of the simplest networks. It is found, for instance, in homes and offices. It includes a relatively small number of devices distributed over a small area. IP addresses within LAN start with 192.168.

c)TCP/IP – MAN

Another next network is MAN – it is a network larger than LAN. This network typically covers a city or group of cities. For example, in the case of the Tri-City, there is one common network developed by research centres, such as the University of Gdańsk and the Gdańsk University of Technology. Most Internet providers in the Tri-City are connected to the Metropolitan network. This type of

network is managed by administrators whose task is to keep the network in good condition. The network may use high-power computers that can perform complex calculations. MAN consists of a larger number of computers than LAN, and LAN is part of it. This network is also spread over a larger area than LAN, e.g. the Tri-City or Warsaw.

d) TCP/IP – WAN

WAN is the type of network where the distances between computers are the largest. These networks make up the Internet. Connections between Internet providers are based on WAN networks, e.g. your local Internet provider must connect to a WAN network to connect to an Internet provider from another continent.

e) TCP/IP – router, server

A router is a device whose function is to interconnect various networks, for instance MAN with LAN. For example, it is used to connect your ISP's network to your home network. Router is usually the main Internet access point in your home. In the slide you can see an example router model.

A server is just a regular computer that performs tasks other than a standard laptop. Most often it has a very poor graphics card, but it is equipped with a powerful processor(s), a huge RAM and disks that can be replaced without turning off the device. Very often it is equipped with two power supplies and two network cards. Thanks to this, if one of the power supplies fails, the server will still be supplied through another unit and will not turn off, and the failure can be removed. The same applies to network adapters – if one of them fails, the server will still be connected to the Internet. Servers usually operate on Linux/Unix or Windows. As a fun fact I can say that Linux is the most common solution when it comes to using it for servers.

f) TCP/IP – application server

An application server means applications that run on a computer and their main task is to wait for HTTP requests from other computers. You can distinguish WWW servers, such as nginx or tomcat. Another type of application servers are FTP servers, used for storing files, e.g. ProFTPD, and e-mail servers used for sending e-mails, e.g. postfix, qmail, or exim.

We will focus on what a WWW server. It is a computer program that supports HTTP requests and sends HTTP responses. At present, WWW servers can be programmed in any programming language.

4. DNS – slide

DNS is the system that is responsible for translating a domain name into an IP address. Thanks to this, the client knows to which IP address they are to send requests and from which IP address they will receive responses. Domain addresses, e.g. wp.pl, are easier to remember than computer IP addresses, e.g. 88.123.56.1. The DNS system consists of 13 DNS root-servers. In addition to DNS root-servers, there are also complementary DNS servers. DNS root-servers do not store information about domains but about the fact that the IP address for a given domain is present on the complementary server. Each domain address has two DNS servers. DNS administration is handled by various institutions around the world, e.g. the USA government for “.mil” or “.gov” domains. In Poland, NASK is the institution that manages domains such as “.pl”, “.gov.pl” or “.com.pl”. To illustrate how DNS works, imagine you have a computer user who wants to open a wp.pl website. Of course, you need to enter wp.pl in the address bar. At this moment, the browser on the user's computer sends a query to the DNS server of your ISP whether they know the IP address for the given domain. If so, the ISP's DNS server sends a query to the DNS root-servers whether they know the IP address for the domain. The DNS root-server responds to your ISP's DNS server that wp.pl has IP address 88.123.56.1. Then your browser connects to the IP address of wp.pl and downloads the content of the page. To sum up, while using the network, the address www.wp.pl needs to be “translated” for the user into an IP address so that it can be found in the Internet resources and be connected to.

5. DNS – Tracert – demo

In order to check which servers the request is going through, we will use the Tracert command. Run the command line by entering cmd.exe in the start menu. Type in tracert google.com in the command line window and press Enter. This way you can view servers through which your request goes to google.com. Then we would like to see the map of the servers through which your request goes to google.com. Go to the page where the map is displayed (<https://stefansundin.github.io/traceroute-mapper/>). In the command line, type in “tracert google.com | clip” so you can copy the route that the request takes to google.com. Paste the clipboard contents from the map page and click the Map It button. This will show you through which servers your request goes to google.com. The end point is the place reached by the request and which can be connected to the point on the map. Not all points can be connected to geographical coordinates. As you can see, the request goes from

Gdańsk, which is true because I was in Gdańsk when recording this material. You can see in the box where you have pasted the traceroute that the second jump was in Gdańsk, while the first one was from my computer to the router at IP 192.168.0.1.

6. URL, URI

URI address consists of URL and URN, where URL is the server name and http or https.

URN is the text after the server name.

In the most extensive form, URI address can consist of:

- protocol, e.g. http
- username and password
- server name, e.g. wp.pl
- port number, which can be a value from 0 to 65535; ports 80 and 443 can be omitted when entering the URI address because these are the default ports
- path to the resource, e.g. News
- parameters e.g. question mark id = 1
- and as the last parameter, which is the anchor to the fragment of the page, e.g. #pagebottom (written together). Thanks to the anchor, you can move the page to a given place in the page's HTML.

7. HTTP methods

For HTTP, there are several methods of retrieving and sending data to and from the URI.

I will discuss some of them. Officially, there are seven operating methods. Each server, however, can support methods other than those official seven ones.

Each HTTP method has its own name and determines whether you can use it to send a value to the Request Body server and whether the server should return a value in the Response Body. In the slide I give an example of what a given HTTP method can be used for.

GET and POST are the two most basic HTTP methods. GET is used to download data from the server, therefore it has a Response Body but does not have a Request Body. In the POST method, there is a Request Body, i.e. the data you send to the server and a Response Body, i.e. the data you receive in the response from the server. PUT and DELETE belong to the remaining methods – they are similar to the POST method, but both are used to send data to the server and delete the resource from the server. An interesting method you can come across is HEAD. This method has neither Request Body nor Response Body, but it only downloads headers from the server.

8. HTTP codes

HTTP codes are codes that the server returns in order to inform you of what has happened to the request that was processed by the server. This is usually code 200 OK, which means that the request was processed correctly. Another interesting code is 500 – it indicates that the server encountered an error while handling the task and was unable to process it. Officially, there are 62 HTTP codes, but you can also create your own codes in your application. It is important not to change the meaning of the existing codes whose values are universal.

Complete the task now – find the meaning of code 418.

9. Headers

Each request can have headers thanks to which you transfer basic information to the server/application with which you communicate. For example, you can authorize your request with the Authorization header. Thanks to this, the server to which you send your request can confirm that you have access to a given URL. You can send a cookie to a server/application that identifies a specific browser – it is the Cookie header. You can specify the type of your request body – the Content-type header. There are many request headers, but you need to check in the documentation of the given server/application which request headers it supports. Often, when the server/application does not support the header, it simply ignores it.

Each response can have headers thanks to which you receive basic information from the server/application with which you communicate. You can get information about the methods supported by the server/application; the Allow header. You can get information that your browser has to save the cookie; the Set-Cookie header. There are many response headers; moreover, each server/application may have headers defined by the server/application owner.

10. Client-to-server communications

Communication between the browser (i.e. the client) starts with entering the URL into the address bar. The server is waiting for your request under the URL provided. The server will return you a response. Normally, the server should return a response to any request.

Open your browser. Open a new tab in it. Click More Tools in the settings to open Developer tools. In the Developer tools, go to the Network tab. Then type wp.pl in the address bar and click Enter. When the page has been loaded, you will see a list of requests to the wp.pl server.

11. Request

When you communicate with the server through its IP or domain, an example HTTP request might look like this. First, enter the HTTP method, i.e. GET, then type in the URI and the protocol name and versions, i.e. HTTP and 1.1. Then type in Host and enter the URL of your server, e.g. wp.pl. Then forward the headers along with their contents, e.g. User-Agent, Accept, Accept-Language, Accept-Encoding, Connection. For the GET method, enter one more empty line which will be your request body. For the POST method, enter the content of your request instead of an empty line. When you are using a browser, it does it all for you. You just enter the server's URL.

12. Response

An example response, i.e. response to a request, begins the determination of the HTTP version. In our example it is version 1.1. Then, the HTTP code is returned along with a brief description. In our example, the HTTP code with a brief description is 200 OK, i.e. the server has successfully processed the request and returned a response. More about HTTP codes in another video. Also Host is returned, e.g. wp.pl, thanks to which the browser knows in which tab the response content should be displayed. Next, headers can be returned. In our response the headers are as follows: Connection, X-Powered-By, Access-Control-Allow-Origin, Content-type. These are sample headers with their corresponding values. More about headers in another video. At the end of the response is its content: it can be HTML, JSON or a plain text. The browser usually tries to display the response content appropriately, and this depends partly on the Content-type header. In our example it will be displayed as html.

13. Rest, JSON, HATEOAS

a) REST

Rest is the way you can create URN addresses in your web application, making it easier to connect the client to the server. This architecture is based on the HTTP protocol because it uses HTTP methods. For example, GET/cars will retrieve a list of cars. While GET/cars/car/1 will retrieve only one car whose ID is 1. Note the plural form when retrieving the list.

Note also the fact that when retrieving a single car, the word car is included in the cars' list and has the singular form. Such naming provides an easy way to distinguish whether a given address will return one object or a list of objects.

And when you combine it with HTTP methods, you get a transparent interface for communication between the client and the server. Pay attention to the POST method and the fact that it does not apply when adding an element to the URN containing a singular form, because you cannot add a car to a car, but you can add a car to a car list.

b) JSON

JSON is a text format used for sending requests and sending responses. Its structure is very simple: the braces define an object, e.g. a car, that consists of fields, e.g. a name whose value is Ford. JSON can also be used to send Collections/arrays of objects by bracketing one or more objects and separating them with a comma, e.g. a collection of two cars. JSON can be very easily converted to objects in programming languages.

Task: Using JSON, write the Car object with the Passengers field in the Notepad. The Passengers field is a collection of 2 Person objects.

c) HATEOAS

HATEOAS – this is adding links to the server response, to which you can make a request. This allows you to retrieve further information about the resource. For example, by sending a GET request, you will retrieve the car with ID 1. You can see that its name is Ford. In the response, there is also a link to the HTTP method that returns the status of a given car.

14. Redirection, apache rewrite

a) Redirection

If the user of your site goes to an address that for some reason has been changed, you can return them a link to another location in response, thanks to which the user will be redirected to a new address. For example, the information about me had the aboutme.html link, but I decided to change it, so the link in the form of HTML on the page omnie.html page is returned to the page ja.html page. In this way you can redirect a user of your site to another location. This method is often used when your browser does not support the Location header. This header tells the browser where you would like to redirect your user when the URI is no longer available. In addition, the header also returns the appropriate HTTP code. The http codes that are responsible for redirection are as follows: 301, 302, 303, 307, 308. This type of redirection can be programmed in the application that will run on your server.

b) Apache rewrite

If a web server called Apache is used, you can redirect users by configuring it in such a way that when it sees a request coming to a given URN, it can immediately redirect it to a different address. For example, the user goes to URN /oldpage.html and is redirected to /newpage.html. In this configuration you need to use HTTP codes. The HTTP codes that are responsible for redirection are as follows: 301, 302, 303, 307, 308. This method is so much better than the usual redirection with the Location header that it does not launch the application located on your server. The downside of this method is that such redirections are usually configured by server administrators, and not by application developers. This means that as a programmer you have to wait for the administrator to change the server setting.

15. HTTP vs HTTPS

When you enter the HTTP website, information from and to your computer is sent in plain text, which means that a spy can eavesdrop your information. It is not a problem when you browse publicly available Internet resources, but when you go to the bank's website to check how much money you have on your account, you do not want everyone to get this information. To prevent possible eavesdropping, the HTTPS protocol has been introduced, thanks to which the traffic from and to the browser is encrypted and it makes it difficult to eavesdrop on what goes from and to our computer. For a given server to be available after HTTPS, it must have an SSL certificate, which confirms that the given server is real and not substituted by a spy. Thanks to this certificate, traffic to and from the server is secure.

16. cURL – installation – Windows

If you have Git for Windows installed, you can skip this step. Open your browser, type in curl.haxx.se in the address bar and press Enter. Then click Download in the top menu. Scroll down until you can see the Win64 link on the right. Click it. Then select the first link under Windows 64 bit. In this example it is 7.66.0. On the next page, click the link to cURL for 64 bit. Wait till it is downloaded. Then click the arrow and select Show in Folder. Right-click on the downloaded file and select Extract All. In the window that appears, click Extract. In the extracted directory there is a bin folder containing a programme called cURL. Go to the upper folder and change the bin to curl. Then click the cURL folder and select Cut. Go to the system drive (usually it is your computer's c: drive), go to Program Files and paste the cURL folder there. In the Access Denied window, click Continue. To make the cURL available in the Command Line, click This

Computer and its Properties. In the window displayed click Advanced System Settings. Then, in the Advanced tab, click Environment Variables. Find the path in User Variables and edit them. Create a new entry where you paste the path containing your cURL. If you are using Windows 7 or an older system, separate the entries with a semicolon. Then close the windows by clicking OK. At this point you need to restart the computer. After restarting the computer, launch the Command Line by entering cmd.exe in the Start menu. In the Command Line, type in curl.exe --version and press Enter. As you can see in the example, you have cURL version 7.55.1. Since you could read the version of your cURL program, it was correctly installed.

17. cURL – demo – windows

CURL is used for sending requests to the server. In our demo we will use a server where you can get to know various requests. To start using cURL, open the Start menu. Type cmd.exe in it and press Enter.

Our first request will use the GET method. Type in cURL and the remaining parameters and press Enter. In response, you get JSON with the Args field where your parameters are located. Then click the Up arrow to repeat what you have recently entered in the command line. Add a dash v to view the details of your request and response. As you can see, you have sent a GET request to the URN using HTTP version 1.1. The request was sent to the server called postman-echo.com. The arrows pointing to the right indicate that this is a request. Below you can see the response from the server with HTTP code 200 OK, the headers and the response content in JSON.

Your next request will be a POST request. In this request you have an additional date parameter where you enter the body of your request. After entering all the parameters click Enter, and a response from the server is displayed. Then click the Up arrow to repeat what you have recently entered in the command line. Add a dash v to view the details of your request and response. As you can see, you have sent a request via the HTTP method named POST to the URN post. This request was sent to the server called postman-echo.com. Below you can see the response from the server with HTTP code 200 OK, headers and response content in JSON.

Then you need to send the POST request in a different way using x-www-form-urlencoded. **To this end, enter parameters in the body as if it were a GET request**, and click Enter. Thanks to this, each parameter can be processed separately by the server, which can be seen in the response. Then click the Up arrow to repeat what you have recently entered in the command line. Add -v to view the details of your request and response. You can see that the POST request has been sent to the URN post using HTTP version 1.1. This time, the request headers present the header called content-type and the values:

application slash x-www-form-urlencoded. In the response you can see that you have used HTTP version 1.1 and you received the HTTP code 200 OK in the response, and in the content of the answer you have JSON.

In your next request, you will send request headers. To this end, add --header in the cURL parameters and, in the quotation marks, enter the header name written without spaces and, after the colon, the header content. Click Enter. Then click the Up arrow to repeat what you have recently entered in the command line. Add -v to see the details of your request and response. You get a JSON response, and in the headers field you can view the header sent. Look at the request you have sent. You can see the HTTP and URN method used and, more importantly, in this example it is your header sent.

The next request will show you that the server can return different headers for your request. Enter your request and click Enter. In the response you can see the text content-type with the value text/html. Then click the Up arrow to repeat what you have recently entered in the command line. Add a dash v to view the details of your request and response. In the response headers you can also see the content-type header with the text/html value. Then, change the values of this header to text/csv and send the request by clicking Enter.

Next, you will learn one of authentication methods called basic authorization, or basic auth for short. To begin, make a GET request. You will receive Unauthorized in response, i.e. it is not authenticated. Then click the Up arrow to repeat what you have recently entered in the command line. Add -v to view the details of your request and response. You can see that you have sent a GET request to the basic-auth URN. In response, you have received the HTTP code 401 Unauthorized. Then click the Up arrow to repeat what you have recently entered in the command line. Add -u, and in quotation marks enter the username and a colon after the password. In our example this is postman and password. This time you have received JSON and information about the correct authorization and HTTP code 200 OK. In your request, the Authorization header was added with value Basic and the encoded username and password.

At the end you will see examples of HTTP response codes starting from 200 OK. Then, 500 Internal server error, and finally, 301 Moved Permanently. To get more examples of what you can do in cURL, go to docs.postman-echo.com to find other examples of using cURL. You will also find examples of how to make requests in a different way. (Delete cookies from the video.)

18. Wget – installation, demo – windows

a) Wget – installation

Windows – not in Git

Open the browser and type eternallybored.org/misc/wget in the address bar. Press Enter. Find the link called exe in the 64-bit column. Click it and download the Wget program. In the folder to which you have downloaded it, right-click the Wget program and click Cut. Go to the system disk (usually it is drive c:) to the Program Files folder. Create a new folder called Wget there and paste the program into it. Copy the path to this folder. To make the cURL available in the Command Line, click This Computer and its Properties. In the window displayed, click Advanced System Settings. Then, on the Advanced tab, click Environment Variables. Find the path in User Variables and edit them. Click the New button and paste the path to the Wget folder. If you are using Windows 7 or an older system, separate the entries with a semicolon. Then close the windows by clicking OK. At this point you need to restart the computer. After restarting the computer, launch the Command Line by entering cmd.exe in the start menu. In the Command Line, enter `wget.exe --version`. As you can see in the example, you have Wget version 1.20.3. Since you could read the cURL program version, Wget has been correctly installed.

b) Wget – demo

The Wget tool is used to quickly download files from the server.

Run the Command Line by entering cmd.exe in the Start menu.

Run the File Explorer. Go to the path given in the terminal. In our example the path is `c:\Users\User`. Create the SDA folder there and go to it. In the Command Line, enter the SDA folder by typing in `cd sda` and clicking Enter. Then type in the Wget command along with the address from which you want to retrieve information. Add the parameter capital O and type in `response.txt`, which will be the name of the file you will download. Then click Enter, and Wget will download the file to your folders. Go to the File Explorer and open the SDA folder where you will find the Response file. Open it. In the file you will see the response downloaded from the server.

19. Ping – demo – windows

The ping tool is used to check whether a given server is available on the Internet.

Run the Command Line by entering cmd.exe in the Start menu. Type in `ping wp.pl`. Now the ping command is checking whether the wp.pl server responds when you send 32 bytes of data to it. As you can see in the summary, one packet sent was lost and the other three packets reached wp.pl and returned

to your computer. You have also a summary of the ping command execution with time indication of how long it took to ping the server.

Then let us try the wp.pl server, but in this case the number of repetitions of sending packets is not limited, due to the -t switch. To stop this command, press ctrl and the letter c.

Now limit the number of packets to 6 by entering -n 6. Another switch is -l that allows you to determine the packet size. In our example, we reduce it to 10 and we can see that the number of bytes is 10. When a ping is used next time, increase the packet size to 15500. You will see that all your packets have been lost, i.e. the wpl.pl server did not respond.

Finally, combine two parameters: -n 2 and -l 100, which means that two 100-byte packets will be sent.

20. SSH – installation – Windows

This method is required when you do not have Git for Windows installed. Open your browser, type sshwindows.sourceforge.net in the address bar and press Enter. Scroll down to the bottom of the page and find the link containing text: Downloads. Click on it. On the next page, you will find the Binary Installer Releases link. It will take you to the page where you can find the link to [setupssh371-20131015.zip](#). Click this link. You will need to wait a moment on the next page. Right-click the downloaded file and select Extract All. A folder with two files will be displayed. Double click the “application” one. A screen will be displayed showing you how to increase the user rights. Click Yes. Read the License, select I Accept, and click Next. On the next screen, leave the default option selected and click Next. Leave the default installation path and click Next. On the last installation configuration screen, click Install. During the installation, you will see a prompt to create the \etc\passwd file. Click OK in this message. At the end of the installation, uncheck the Show Quickstart Guide and click Finish. Go to the installation folder. In the etc folder, copy and paste the ssh_config file and change its name to password. Right click on it and open it with Notepad. Highlight the entire contents of the file and delete it. Finally, save the file and close the Notepad. Run the Command Line by entering cmd.exe in the Start menu. In the Command Line, type in ssh -v. You can see you have SSH version 3.7.1p1 installed.

21. SSH – demo – Windows

The SSH tool is used to connect to the server and execute commands on the server's terminal. SSH is the Internet communication standard and the successor of the telnet protocol.

To use this tool, run Git Bash by typing `git bash` in the Start menu. This tool will also work via the command line, i.e. `cmd.exe`. Then enter `SSH jan`, i.e. username and server address in the demo. It will be a test server in my home network at IP `192.168.0.20`. This server operates on port `32769`, thus we use the parameter `-p` and the parameter value of `32769`. By default, SSH operates on port `22` and if so, you do not need to enter the `-p` parameter. The first time you connect to a given server, you will see confirmation that you trust that server. Type in `yes` and press Enter. Then the server you connect to will ask for the password of the user you log in to. Enter a password, which cannot be seen for security reasons. When the password is entered, click Enter and log in to the server. Then execute the `ls` command with parameters `l` and `a` on the server. The home directory of user `jan` is displayed. To log out of this server and end the SSH session, type in `enter` and return to the terminal on the local computer.

Another function of SSH is to execute commands without entering the server. As before, type in `ssh jan@192.168.0.20 -p 32769`. In quotation marks, type in the command `ls -la`, and click Enter. As in the previous example, the server requests the user password. Type it in and click Enter. As you can see, exactly the same content is displayed as the one before, when you logged in to the server. Another (and more secure) option is to log in without using a password. To log in this way, you need to generate a public key for your computer. Type in `ssh-keygen.exe` and press Enter. Click Enter for the next 3 steps. In this way, you will generate a public key not protected by password. Then copy the public key to the server at IP `192.168.0.20`. After entering a fairly long command, click Enter and a password prompt will be displayed. Enter the password and your public key will be copied. Remember that the public key is a file with extension `.pub`. Then, to check whether the public key has been copied, run the `ls -la` command on the server without logging in to it. Later in the video, we will focus on configuring the SSH client. To start editing it, type in `vi tilde slash dot ssh slash config`. The vim window opens. To start entering the text you need to press the letter `"i"`, and you will see the entry at the bottom of the Git Bash window. Enter the IP of the host for which this configuration will be used. Then type in the port to be used when connecting to this host. To close the file and save the changes, press the `[escape]` key and type in: `wq`. Now you can connect to the test server without specifying the port. Then configure the SSH client so that you do not need to type in your login when connecting to the test server. Edit the configuration. Add `User space jan` in it. Now you can connect to the test server without specifying the login as well. Since it is more difficult to remember the computer's IP than to assign a nice name to it, edit the client's configuration to give it a name you invented. In the configuration, type in

HostName space 192.168.0.20. Change the value of Host to test.server. Then log in to your test server by typing in what you have entered in the HostName field. Add a configuration that does not allow the SSH server to disconnect with the client connected to it. Enter Host space asterisk so this setting will apply when connecting to any server. In the next line, type in ServerAliveInterval space 120, which means that every 120 seconds the client will send information to the server that the connection is still required. Log in to the test server and check whether the configuration is correct.

Finally, you may want to overwrite the configuration for the given server. Use the -o option. Type `ssh -o "User = root" test.server` and click Enter to log in to your test server as root.

22. Live HTTP headers – installation, demo - chrome

Open a browser where you have installed the live http headers plug-in. Click the plug-in icon to open the tab with the plug-in interface. On the left, your requests and responses from the server will be shown. In another tab, go to wp.pl and go back to the tab with the plug-in running. Plenty of requests are displayed there. Most of them have the HTTP code 200 OK. Open the first request and you will see the type of request for the URN. Open the settings. On the left, you can determine what will be captured, and on the right, how the headers will be displayed and how the list of requests and responses will be sorted. Click on "Show All" to see all information about the selected request and response. You can select displaying headers as they come to your browser, or uncheck the Raw option to have them coloured, thanks to which you can immediately see where the request headers and the response headers are located.

23. Postman – installation - Windows

Open the browser and type getpostman.com into the address bar. Click Download the App. On the next page, select Windows 64-bit. The download starts. When the download is finished, click on the downloaded file to start the installation. Wait a moment till the account creation form opens. Create an account by entering your email address, username and password. Accept the terms of use and create an account. Then you may check the appropriate checkbox on the left side of the screen, but it is not necessary. In the second point, choose the Postman appearance. Leave the default values here as well. Finally, the application suggests creating a team, but we will skip this step. At the end you get information about what you can do in the Postman app, which confirms that you have successfully installed the Postman.

24. Postman – demo – Windows

Postman is used to create API documentation. This means you can store and execute requests and share them with other developers. In the panel on the left, in the History tab you can find requests that you have made using postman. Then there are Collections where you store your requests. On the right in the upper part, you can see Environments. In Environments, you store variables to be used in your requests. In the middle of the upper part you can change your workspace.

Close the Welcome tab and open the tab with a new request. Postman can send requests using various HTTP methods. Select the HTTP method and go to the field where you enter the URL to which your HTTP request will be sent. Type in wp.pl there and click Send. A GET request will be sent to the wp.pl address. Under the request, there is a place for the response to be displayed. Also HTTP status is shown – in our example it is 200 OK. The content of the response is html and you can view it in the nicely coloured response window. You can also render this html by clicking Preview. The view displayed resembles a web browser.

Close the request tab without saving it. As you can see, this specific request has been moved to the request history on the left. Go to the Collection tab where you can find an example collection. There are 20 requests there, separated into folders. Create a new collection. Enter its name in the Name field and click Create. Now create a tab with a new request. This will be a GET request. Enter the request URL with parameters. In the Params tab under the URL, parameters to be entered into the URL are shown. Send your request and after a while you will get a response in JSON. In the Args field you will see the parameters you have sent. The response status is 200 OK. The next step is to save this request to the created request collection. First select the collection, enter the request name and click Save. The saved request appears on the left. Close the tab with this request. Now you can open the request and you will see that all the parameters have been saved.

The next request will be a POST request. Open another tab and select POST from the list of request types. Then enter the URL to which you will send the request. Open the Body tab – it is where you will be sending the content of the request. Select the Raw content type and enter the content. Then, send the request. The content in the Date field is exactly the same as the content you have sent. This request also has the 200 OK status. The next step will be to save this request in your collection. Enter the request name, select the collection and save it.

The next task will be to send a request with headers. The request type should be GET. Enter the address. Select Headers from the tabs under URL. Enter the header name in the Key column, and its value in the Value column. Remember

that the header name cannot contain a space character. Send the request. In the Headers field in JSON, there is a field which is named the same as the header you have sent. Add another header and send your new request. Finally, save the request to your collection.

The next task will be to send a request that will return headers in response. Postman Echo can return response headers that will be the same as the parameters you have sent to it. Enter the request parameters and click Send. In the Headers tab in the response, there is a header with the same value as the parameter sent. The next step is to change the value from text/html to text/vcf and submit the request. The header value in the response has changed.

Depending on the content type header, the Postman can format the response from the server. Thus, when you type in text/json as the next value of the header, in the preview you will see the coloured response. It is another example of your request, so save it with a relevant name.

Then send the request to the secured URL. After sending the request, the response from the server has HTTP code: 401 Unauthorized. To receive information from the server that your request is correct, you need to go to the Authorization tab and select the authentication type as Basic Auth. Then enter the username and password on the left and submit the request. In the response you will see that you have been authorized, and the HTTP code will be 200 OK. Then save your request.

In the next part of the video, you will see various HTTP codes in a response from the server. Enter the URL and click Send. The response from the server is 200 OK. Then change the URL and send another request. This time you get HTTP code 500 Internal Server Error. Change the URL once again and send a request. You will receive HTTP code 302 Found in the response. Finally, save your request. Another functionality of the Postman is Environments. At the beginning, you do not have any environment in your workspace. Click the gear on the right and then click the Add button in the window displayed to add a new environment. Give it a name and save it by clicking Add. Close the window. Your first environment will be shown in the list of environments. The environment affects all of your URLs in all collections, which means that all variables that you have entered in the environment are available for each request. By clicking the eye icon you can view what variables are present in the selected environment. You do not have any variables so far, so do not click the preview. Instead, add a variable to your environment. This variable will be called URL and its value will be the address to the test server. After entering the variable, click the Update button and close the Environments window. Then edit each URL so that it contains the URL variable. To use the variable, simply enter the two brace openings, type in the name you have entered in the environment, and finally enter two closing brackets. This way, the Postman will use the value of this variable when sending requests. When you hover with the cursor over this variable, its current value will be displayed. Finally, save the modified request.

Modify each of the previously added requests in your test collection. Remember to save your requests after modifying them.

In the last part of the video, a new workspace is created. All previous requests were located in the collection that was in the My Workspace (the default workspace). Click My Workspace in the upper part of the Postman and then click Create New. The Workspace creation window opens. Enter the name and select Personal so that this space will be available only to you. Then click Create Workspace. The new space that has been created contains no collection, no environment and no saved request. By clicking the name of the space, you can switch to another space.