

# Web basics

Stanislav Protasov

# Agenda

- Internet in a nutshell
- Web history and basics
  - *Idea of the Web*
  - *Web transport and formats*
  - *Browsers and DOM*
- Crawling basics
  - ***robots.txt*** and ***sitemap***
  - *Terms and Conditions*
  - *APIs*

Ok, Google, what is internet?

# IP protocol and address

Construct URL which looks like whatever, but leads to mail.ru:

1. Obtain IP address of mail.ru website
2. Convert IP to hex
3. Create https://**whatever:@ip** url

213.159.212.4 (IU) - fails with browser

81.19.74.0 (LJ) - works and redirects

ping 192.34.57.61

ping 0xC022393D

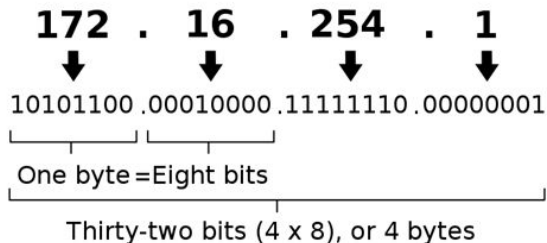
ping 3223468349 # same server

http://meduza.io:@0x51134a00/

# typical phishing

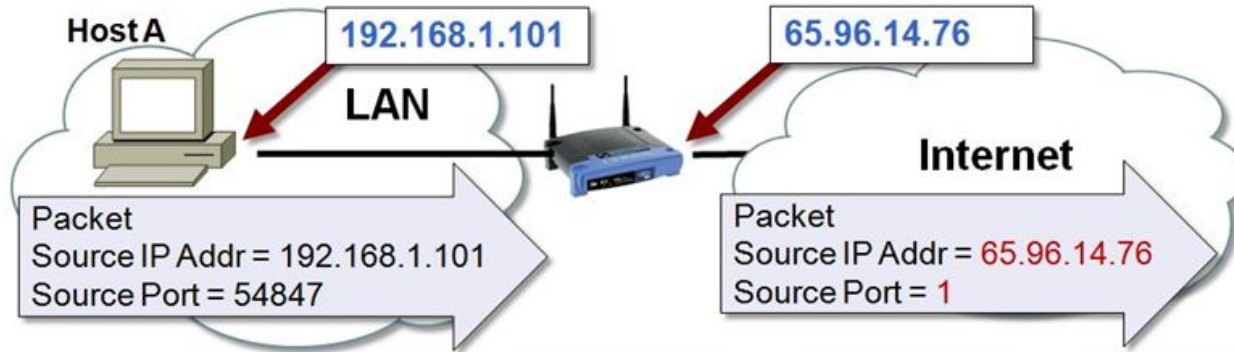
IPv6 - 128 bits

An IPv4 address (dotted-decimal notation)



192.0.2.235	
0xC0.0x00.0x02.0xEB	
0300.0000.0002.0353	
0xC00002EB	
3221226219	
030000001353	4

# IP != machine



NAT Translation Table				
	Local IP Address	Source Port #	Internet IP Address	Source Port #
process X, Host A →	192.168.1.101	54,847	= 65.96.14.76	1
Host B →	192.168.1.103	24,123	= 65.96.14.76	2
process Y, Host A →	192.168.1.101	42,156	= 65.96.14.76	3
Host C →	192.168.1.102	33,543	= 65.96.14.76	4

Try on your machine

```
W> ipconfig.exe | findstr IPv
```

```
L/M> ifconfig | grep inet
```

How many addresses do you have?

# Country and IP

Using IP is one of pretty (surprisingly) reliable ways of geo location.

- [GeoIP](#) 99% for country detection, 95% for city detection

To detect country you need a **database**.

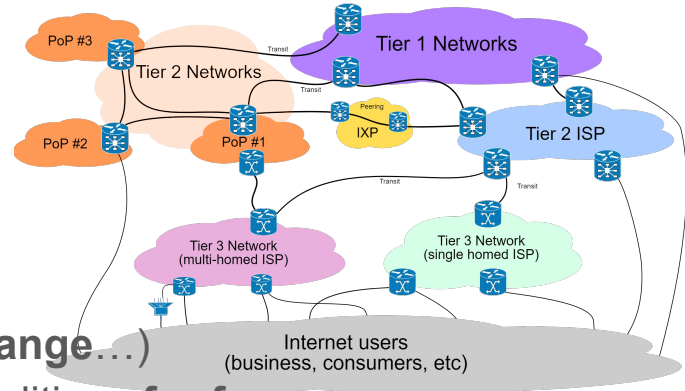
**Service processes IP of the last visible node** in a chain, so:

- NAT
- Proxy
- Turbo mode and VPN

will fool the service.

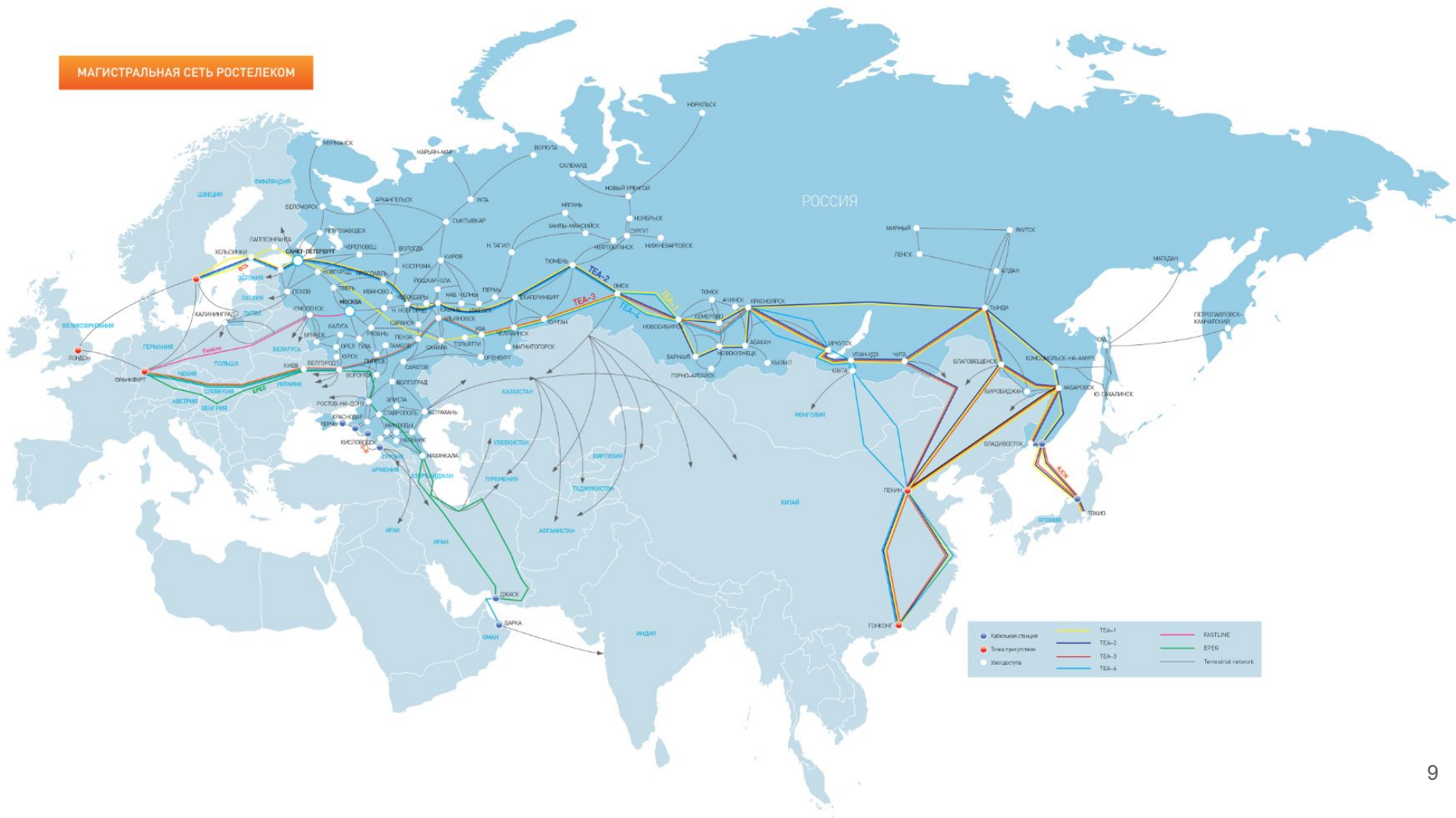
# Tier-X operators ([podcast in rus](#))

- **Tier-1 operators**
  - (e.g. Rostelecom\*, MTS\*, ... for Russia | **AT&T, Orange...**)
  - These networks exchange traffic ([IXP](#)) on peer conditions **for free**
  - Altogether can be considered as a **backbone** of the Internet
- **Tier-2 operators** have **partially free** peering with some segments but **paid transit** to other segments
- **Tier-3** have only paid access to the Internet
- **Takeaways:**
  - This is all about level-3 (routing) of ISO model
  - Local operators are **subjects of regulations** (RKN)
  - **Backup** channels, VPNs and so on are **important for service quality**
  - Think twice about **where to attach/rent/build your DC**
  - Small networks share the same IP-address for other networks
  - **IPv4 is over**





# МАГИСТРАЛЬНАЯ СЕТЬ РОСТЕЛЕКОМ



C:\Users\s\protosov\tracert -h 250 -w 1 rutracker.org

Трассировка маршрута к rutracker.org [195.82.146.214]  
с максимальным числом прыжков 50:

1		2 ms	1 ms	1 ms	10.91.64.1
2		22 ms	20 ms	39 ms	87.226.217.249
3		*	17 ms	18 ms	213.59.212.235
4		48 ms	48 ms	48 ms	188.254.83.102
5		*	*	*	Превышен интервал ожидания

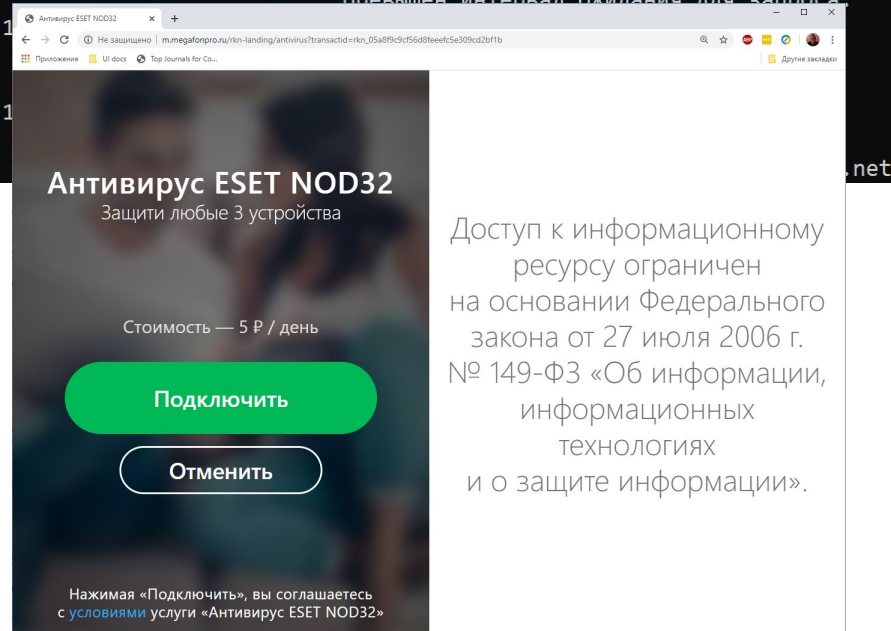
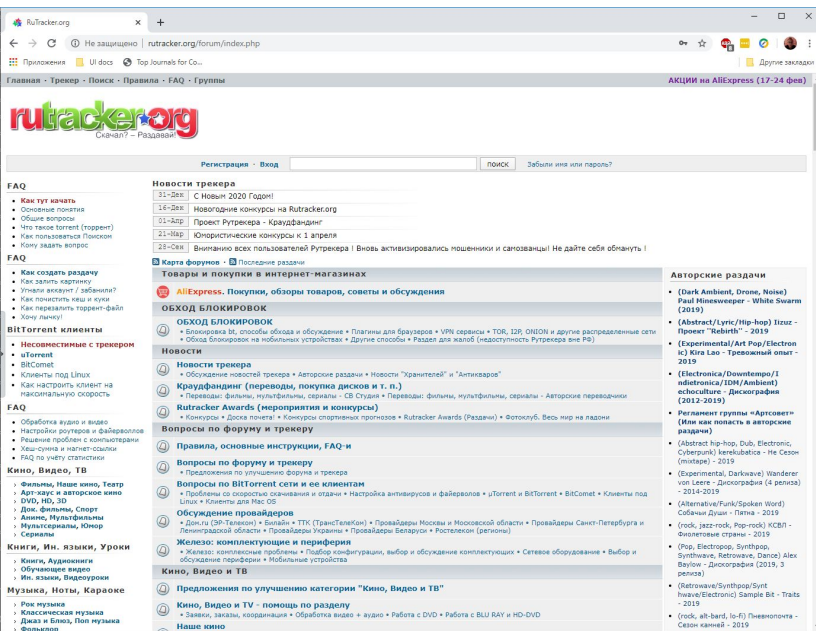
RTK

C:\Users\s\protosov\tracert -h 250 -w 1 rutracker.org

Трассировка маршрута к rutracker.org [195.82.146.214]  
с максимальным числом прыжков 50:

1	*	*	3 ms	192.168.43.116
2	*	*	*	Превышен интервал ожидания для запроса.
3	*	*	*	Превышен интервал ожидания для запроса.
4	*	*	*	Превышен интервал ожидания для запроса.
5	*	*	*	Превышен интервал ожидания для запроса.
6	*	*	*	Превышен интервал ожидания для запроса.
7	*	*	*	Превышен интервал ожидания для запроса.
8	57 ms	38 ms	47 ms	37.29.5.65
9	*	*	*	Превышен интервал ожидания для запроса.
10	*	*	*	Превышен интервал ожидания для запроса.
11	*	*	*	Превышен интервал ожидания для запроса.

Megafon



Try on your machine

**W: tracert -w 1 ya.ru**

**L/M: traceroute -w 1 ya.ru**

**\*\* sudo apt-get install traceroute**

also not sure about Linux Subsystem for Windows

Where is provider, where is Yandex?

```
C:\Users\stani>tracert -w 1 ya.ru
```

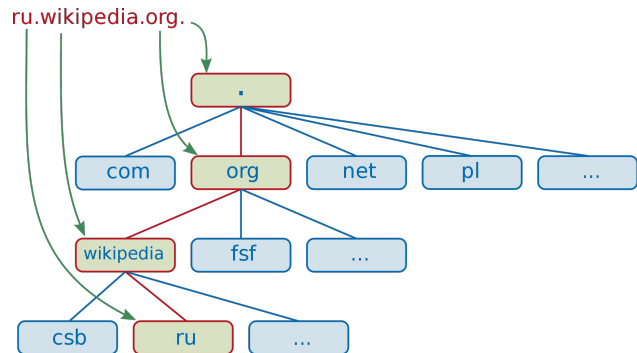
```
Tracing route to ya.ru [87.250.250.242]  
over a maximum of 30 hops:
```

				<b>IU corporate network</b>	
1	1 ms	1 ms	1 ms	10.91.64.1	<b>Tattelekom</b>
2	4 ms	4 ms	4 ms	1.123.18.84.in-addr.arpa [84.18.123.1]	<b>RETN (T2)</b>
3	3 ms	3 ms	3 ms	ae11-498.rt.itp.kzn.ru.retn.net [87.245.231.208]	
4	18 ms	17 ms	19 ms	ae3-4.rt.m9.msk.ru.retn.net [87.245.233.89]	
5	*	59 ms	61 ms	gw-yandex.retn.net [87.245.229.253]	<b>MSK-IX</b>
6	79 ms	66 ms	63 ms	sas-32z5-ae2-1.yndx.net [87.250.239.203]	
7	*	*	*	Request timed out.	<b>Sasovo Yandex DC</b>
8	62 ms	65 ms	62 ms	ya.ru [87.250.250.242]	



# Name services

**DNS — domain name system**, allows to operate human-readable names instead of addresses



There are 13 core [Root Servers](https://www.root-servers.net) ([a..m].root-servers.net) responsible for the Internet. Lower level responsible for domains, subdomains, ...

DNS supports forward (domain → IP) and reverse (IP → domain) requests

`nslookup sprotasov.ru`

`nslookup 192.34.57.61`

`nslookup code-test.ru`

```
root@simpletrack:~# ping -c 1 yandex.ru
PING yandex.ru (77.88.55.80) 56(84) bytes of data.
64 bytes from yandex.ru (77.88.55.80): icmp_seq=1 ttl=244 time=116 ms

--- yandex.ru ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 116.697/116.697/116.697/0.000 ms
root@simpletrack:~# logout
Connection to sprotasov.ru closed.
(base) stranger@sprotasovn:~$ ping yandex.ru
PING yandex.ru (213.180.193.56) 56(84) bytes of data.
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=1 ttl=56
```

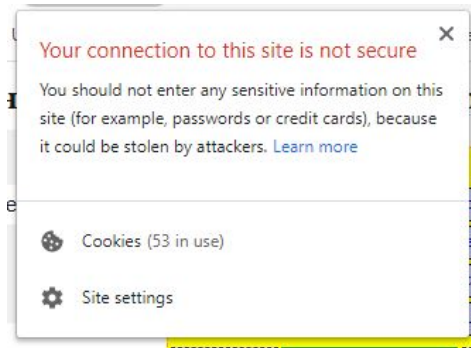
# Resource record

The screenshot shows the DigitalOcean Control Panel interface for managing DNS records. The left sidebar contains navigation links: PROJECTS, MANAGE, DISCOVER, and ACCOUNT. The main content area is titled "cloud.digitalocean.com/networking/domains/sprotasov.ru?i=bcb047". It features a search bar, a "Create" button, and a "USAGE \$4.32" indicator. Below the search bar, there are tabs for different record types: A, AAAA, CNAME, MX, TXT, NS, SRV, and CAA. A text box explains: "Use @ to create the record at the root of the domain or enter a hostname to create it elsewhere. A records are for IPv4 addresses only and tell a request where your domain should direct to." The form includes three input fields: "HOSTNAME" (with a placeholder "Enter @ or hostname"), "WILL DIRECT TO" (with a placeholder "Select resource or enter custom IP"), and "TTL (SECONDS)" (with a placeholder "Enter TTL 3600" and a green checkmark). A "Create Record" button is located to the right of the TTL field. Below the form, the "DNS records" section displays a table with the following data:

Type	Hostname	Value	TTL (seconds)
NS	sprotasov.ru	directs to ns3.digitalocean.com.	1800 <a href="#">More</a>
A	sprotasov.ru	directs to 192.34.57.61	1800 <a href="#">More</a>
NS	sprotasov.ru	directs to ns1.digitalocean.com.	1800 <a href="#">More</a>
NS	sprotasov.ru	directs to ns2.digitalocean.com.	1800 <a href="#">More</a>

# Short quiz. Test yourself

1. One website opens for me, but another — does not. Same for my neighbour in the office. Why?
2. One website opens for me, but another — does not. Both work for my neighbour in office. Why?
3. I clicked a link starting with *google.com* in my email. I know that all google pages are secured, but browser says that Why?



# Before we continue - Internet

1. **Internet and IP.** **IP** manages (routes) how data is flowing from one machine (e.g. server) to another (e.g. smartphone). Providers are working on this level
  - a. Tools: `tracert`, `ping`, `ifconfig/ipconfig`
  - b. **TCP** manages how to transfer more than one packet of data preserving order and integrity
    - i. Tools: `nc`, `telnet`
2. **DNS.** Assigning string names to IP-addresses allow to establish many-to-many relations, thus, make infrastructure faster and reliable. Domain names are objects of legal regulations ([whois](#))
  - a. Tools: `nslookup`, `ping`



# Web and HTTP

# ~~KEY~~ BUZZ WORDS

**Internet** – network for transferring information among devices

**WWW (Web)** – graph of documents (hypertext), placed at web-servers, that are connected to Internet

**Hypertext** – text, that contain references to other texts

**HTML** (hypertext markup language) – standard of hypertext for Internet

**Web 2.0** – everything, that is beyond static HTML documents: social networks, blogs, video-hostings, internet-marketing, web application and services. Service-oriented network.

**Web 3.0** (*semantic web*) – graph of machine-readable, semantically rich documents. Content-oriented network.

# HYPERTEXT EVOLUTION

Web 1.0	Web 2.0	Web 2.0, 3.0
<b>HTML</b> – Subset of SGML (markup language)	<b>xHTML</b> – fusion of HTML tags and XML standard	<b>[x]HTML5</b> – valuable layout changes, semantic tags were added
<b>SGML</b> parsers	<b>XML</b> parsers	<b>HTML5/XML</b> parsers
For presentation in browsers	For displaying interactive and media content	For creation of web-applications that support semantic markup

# URI VS URL

**Uniform resource identifier (URI)** – machine-readable text identifier of the resource, created according specific rules

URI common syntax: *scheme:scheme-specific-part*

**Uniform resource locator (URL)** – subset of URI, describes location and way (protocol) to access object in the Internet

<http://www.mail.ru/>

**Uniform resource name (URN)** – subset of URI, identifies object, but does not locate it

<urn:isbn:0451450523> or

[Magnet links](#): <magnet:?xt=urn:btih:c12fe1c06bba254a9dc9f519b335aa7c1367a88a>

# URL SYNTAX

protocol  
**http**:// login:password  
**user:pass**@  
hostname or IP-address **www.example.com**: port **80**  
path to resource on the server **/Path/to/Res**  
parameters **?k1=v1&k2=v2** identifier inside the document **#hash**

## Examples:

- <https://mail.google.com/>
- <ftp://root:qwerty@ftp.example.com/>
- <wss://server.name:443/method/name>
- <http://sprotasov.ru/index.html#author:Aleksandr%20Buyanov>
- <https://innopolis.com/index.php/fdgdflgkdjf;lgk%20djf;lgkdjf;gl%20kfdj;glkdkgj%20;dlkdj;g%20lkdjg%20;df>

# HTTP

**HTTP** (hypertext transfer protocol) – application (7) level protocol to deliver text data. Created to transfer hypertext. Provide communication between *client* (usually browser) and *server* (web-server) using client requests and server responses.

**HTTP v1.0** – does not support using single TCP session for multiple requests. Supports following client request methods:

- GET – get content from the server
- HEAD – get only header from the server without content ("what to expect")
- POST – sent data to the server

**HTTP v1.1.** – supports also PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH

**HTTP/2** - SDPY (Google) based update. Binary. Header compression, Server pushes, conveyor requests, request multiplexing over single TCP

# HTTP HEADER

HTTP request =

Request URI + **HOST** + **[[headers]]** + <empty\_line> + **body**

```
POST /index.html HTTP/1.1
```

```
HOST: example.org:8080
```

```
Cache-Control: max-age=0
```

```
Accept: text/html,application/xhtml+xml
```

```
...
```

```
param1=value1&param2=value2
```

HTTP response=

Response code + **headers** + <empty\_line> + **body**

```
HTTP/1.0 200 OK
```

```
Content-Type: text/html; charset=UTF-8
```

```
Content-Length: 3123
```

```
...
```

```
<html>...
```

# Demo with telnet

```
o sprotasov.ru 80
```

```
HEAD / HTTP/1.1
```

```
host:sprotasov.ru
```

```
HEAD / HTTP/1.1
```

```
host:code-test.ru
```

```
GET / HTTP/1.1
```

```
host:code-test.ru
```



# IMPORTANT HEADERS

## REQUEST

Accept, Accept-Charset, Accept-Encoding – formats, that your browser understands (text/plain, application/xml), encodings (utf-8) and supported compression algorithms (gzip, deflate)

Authorization – header that stores authentication type, credentials/keys,...

Authorization: Basic QWxhZGRpbjpvGVuIHNIc2FtZQ==

Content-Length – request body length (same for response)

User-Agent – browser and operating system

Upgrade – request to change communication protocol (Upgrade:websocket)

## RESPONSE

Cache-Control – time to store document in a browser cache

Content-Encoding, Content-Language, Content-Type – content characteristics

# Browser fingerprint: domain I see first time

The screenshot shows a web browser window with the address bar displaying "Example Domain" and "Not secure | example.com". The browser's developer tools are open, showing the Network tab. The left sidebar of the developer tools contains a message: "Example Domain. This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission. More information..."

The Network tab displays a list of requests. The first request is to "example.com". The request details are shown in the right pane:

- Name:** example.com
- Method:** GET / HTTP/1.1
- Host:** example.com
- Connection:** keep-alive
- Upgrade-Insecure-Requests:** 1
- User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.116 Safari/537.36
- Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9

The bottom of the developer tools shows the Console tab with the message "What's New".

# RESPONSE CODES

- 1xx – information
  - 101 - Switching Protocols
- 2xx – success
  - 200 – OK
  - 201 – created (new resource)
  - 206 – partial content
- 3xx – redirection (specified header Location: addr)
  - 301 – moved permanently
  - 304 – not modified
- 4xx – client-side error
  - 401 – unauthorized
  - 404 – not found
- 5xx – server-side error
  - 503 – server unavailable

# REQUESTS AND SYSTEM STATE: SAFE AND IDEMPOTENT REQUESTS

Safe request does not change server/object state. For getting some information

Idempotent request if you make 2 or more identical requests, second and other requests do not change server/object state

$F(\text{state}) == F(F(\text{state}))$

	SAFE	NOT SAFE
IDEMPOTENT	HEAD, GET, OPTIONS, TRACE	PUT, DELETE
NON-IDEMPOTENT	--	POST, PATCH

# PARAMETERS

GET params:

- `http://server.name/path?param1=value1&param2=value2`

POST params:

- `POST /path HTTP/1.1`

`param1=value1&param2=value2`

# COOKIES

- Cookies – small **drive space** to store data sent by server to browser. Max – 4KB
- We need cookies for **stateful services** (e-shop cart, etc) or for storing session keys
- Cookies have life period and are **sent** to server with **each request**

# COOKIES EXAMPLES



# Before we continue - Web

1. **Hypertext, HTML and HTTP.** Hypertext is an approach to represent **linked** documents (altogether = The Web). xHTML5 is a de-facto standard. HTTP - a protocol for transferring [hyper]text data, or text-encoded media (base64). Defines methods (GET, POST, ...), status codes (200, 403, 502), headers (metainformation), sessions (1.1+). Works over TCP (means one HTTP message can be bigger than 1 IP frame).
  - a. Tools: telnet, curl, wget, postman
2. **URI  $\supset$  URL.** URL is a standard way to define together:
  - a. Where is the document (domain + port + path)
  - b. How to access the document (protocol, credentials)



# FTP

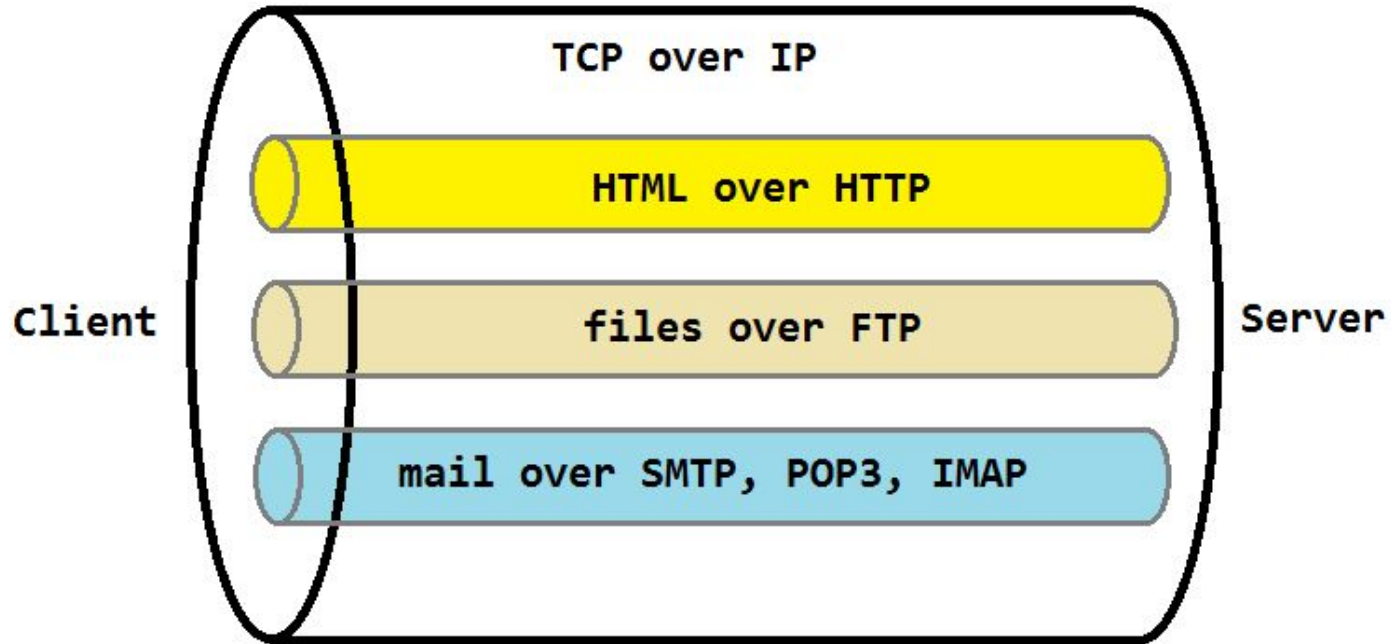
**FTP (file transfer protocol)**, 1971. Consists of greeting session and request session (VERB param [params]\015\012), and server responses

Client:		Connects @server:21
Server:	220 Hello,...	
Client:	USER MB1234	
Server:	331 Password required to access user account MB1234.	
Client:	PASS QXJ4Z2AF	PLAIN TEXT
Server:	230 Logged in.	
Client:	CWD Bills	Change directory to "Bills."
Server:	250 "/home/MB1234/Bills" is new working directory.	
Client:	PORT 192,168,1,2,7,138	accepts data @client:1930 [=7*256 + 138]
Server:	200 PORT command successful.	
Client:	LIST	Send the list of files in "Bills."
Server:	150 Opening ASCII mode data connection for /bin/ls.	server connects out from its port 20 to port client:1930
Server:	226 Listing completed.	succeeded
Client:	PORT 192,168,1,2,7,139	
Server:	200 PORT command successful.	
Client:	RETR Yoyodyne.TXT	Download "Yoyodyne.TXT."
Server:	150 Opening ASCII mode data connection for Yoyodyne.TXT	
Server:	226 Transfer completed.	succeeded
Client:	QUIT	
Server:	221 Goodbye.	

# EMAIL: SMTP, IMAP, POP3

- **SMTP** (simple mail transfer protocol, @:25) – for **transferring messages between servers** and for **server-client** communication. FTP's brother.
- ~~POP3~~ (post office protocol v3, @:110) – standard protocol for **client to get messages from server**
- **IMAP** (internet message access protocol, @:143) – **standard protocol for client to get messages from server**; has **sending implementation** (considered bad), keeps session, supports multiple clients for 1 mailbox.

# OVERVIEW



# Web security (client side)

# TERMS

**Identification** – assigning labels (IDs) to objects, as long as process of comparing one label with the list

**Authentication** – procedure of checking authenticity, proving match between ID and object. We can authenticate user (ID + password), machine, document (digital signature). Can be multi-factor, one-way, both-way

**Authorization** – granting access to perform some action

# FUNCTION OF HTTP-AUTHENTICATION

- Limiting access by means of HTTP protocol
  - Rare for sites. Most sites use forms-based authentication
  - Common for **services** and **APIs**. (access not via browser UI, but server or ajax code)

# COMMON FACTS ABOUT AUTHENTICATION

If server returns 401, this means it wants to authenticate you. Server must send [WWW-Authenticate](#) header to you.

HTTP/1.0 401 Unauthorized

Cache-Control: no-cache

Pragma: no-cache

Content-Length: 58

Content-Type: text/html

Expires: -1

Server: Microsoft-IIS/8.0

WWW-Authenticate: Basic realm="area to be accessed"

# BASIC AUTHENTICATION

Easiest way to setup authentication

```
GET /sometail.aspx HTTP/1.1  
Host: somehost  
Authorization: Basic bG9naW46cGFzc3cwcmQ=
```

where

`“bG9naW46cGFzc3cwcmQ=” == base64(“login:password”)`

NB:

- Login and password are not secured in fact! Only way to use – over HTTPS
- You can send this without challenge
- With each request



# DIGEST AUTHENTICATION

HTTP/1.1 401 Unauthorized

WWW-Authenticate: Digest realm="testrealm@host.com",

nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",

opaque="5ccc069c403ebaf9f0171e9517f40e41"

... ..

Authorization: Digest username="Mufasa",

realm="testrealm@host.com",

nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",

uri="/dir/index.html",

response="e966c932a9242554e42c8ee200cec7f6",

opaque="5ccc069c403ebaf9f0171e9517f40e41"

$HA1 = MD5(A1) = MD5(\text{username} : \text{realm} : \text{password})$

$HA2 = MD5(A2) = MD5(\text{method} : \text{digestURI})$

$\text{response} = MD5(HA1 : \text{nonce} : HA2)$

[RFC 2617](#):

$HA1 = MD5(A1) = MD5(\text{username} : \text{realm} : \text{password})$

$HA2 = MD5(A2) = MD5(\text{method} : \text{digestURI})$

Если значение директивы QOP равно «auth-int», то HA2 равняется:

$HA2 = MD5(A2) = MD5(\text{method} : \text{digestURI} : MD5(\text{entityBody}))$

Если значение директивы QOP равно «auth» или «auth-int»,

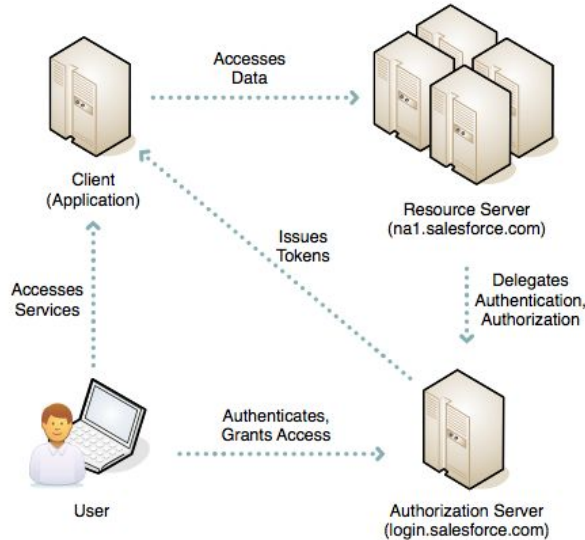
$\text{response} = MD5(HA1 : \text{nonce} : \text{nonceCount} : \text{clientNonce} : \text{qop} : HA2)$

Если директива QOP не определена, то ответ вычисляется так:

$\text{response} = MD5(HA1 : \text{nonce} : HA2)$

# OAUTH

- Authenticates application on behalf of user (or anonymously)
- Based on Access Tokens. [Twitter example](#)



# FORMS AUTHENTICATION

- Not a part of HTTP protocol
- Based on HTML <FORM>-tag and request parameters mechanism

The diagram shows an HTML form with the following code and annotations:

```
<form action="Default.aspx" method="get">
  Login: <input type="text" name="username" />
  <br/><br/>
  Password: <input type="password" name="password" />
  <br/><br/>
  <input type="submit" value="Log me in"/>
</form>
```

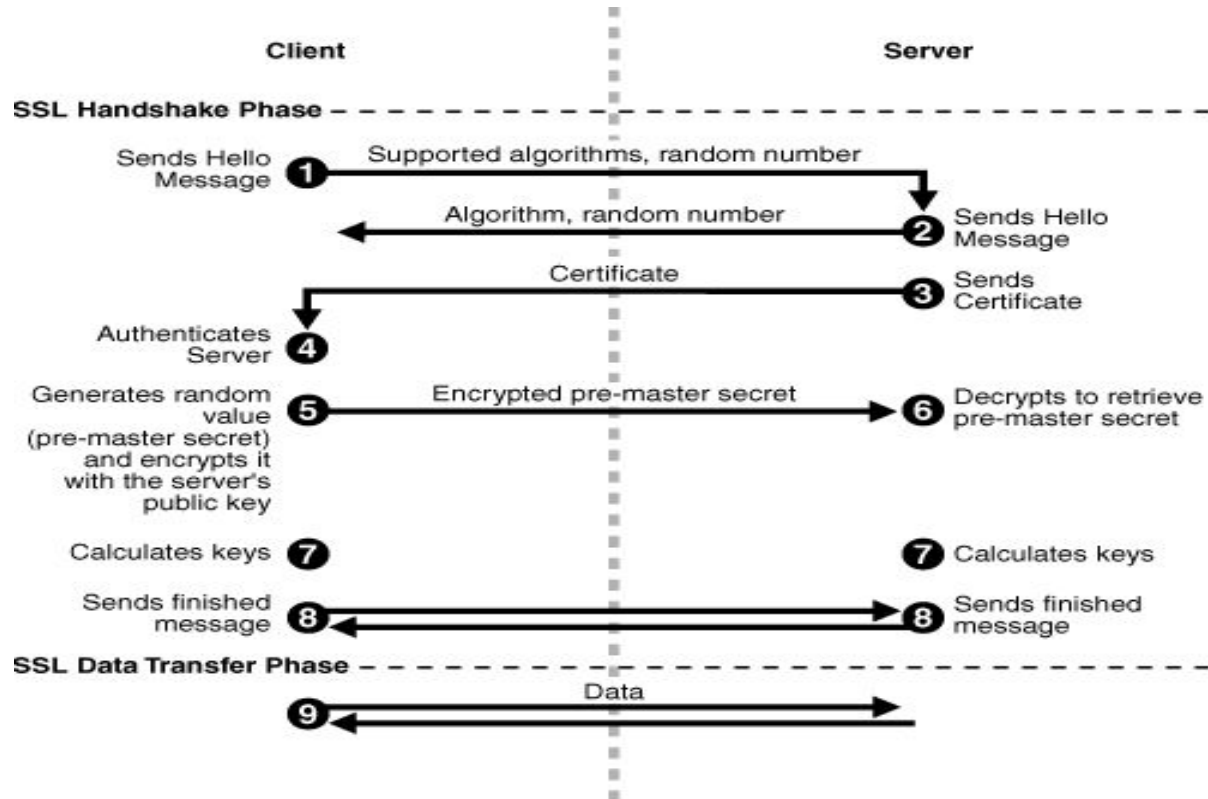
Annotations:

- request destination**: points to `action="Default.aspx"`
- request method (get/post)**: points to `method="get"`
- "username=..." param will be created**: points to `name="username"`
- "password=..." param will be created**: points to `name="password"`
- covers field with "stars"**: points to `type="password"`
- button that triggers submission (request)**: points to `type="submit"`

# HTTP SECURE

- **HTTPS** = HTTP over SSL/TLS
  - SSL – protocol with asymmetric cryptography and symmetric encoding
  - TLS = SSL v3
- **HTTP** (FTP, telnet) work transparently over SSL/TLS
  - Firstly client's application (browser) performs “handshake”.
  - Then the channel is created and data is sent over this channel using standard protocol (e.g. HTTP)

# HTTPS HANDSHAKE



# CERTIFICATES

**Digital certificate** – electronic document (file), ensuring that public key belongs to bearer. Certificate must be signed by certification authority.

- Mandatory cert parts:
  - resource ID (**Subject**)
  - **public** key
  - certification authority (**Issuer**)
- Optional cert parts
  - private key
  - usage restrictions

# Certificates examples

<https://tv.eurosport.com/>

W: certmgr.msc

# Before we continue - Security

1. HTTP supports **Basic and Digest authentication** of a user from the box (defined in standard). Mostly used for service-to-service interaction. **Data is still plaintext.**
2. **OAuth** is a new way to grant access to the service. Access to APPLICATION on behalf of a user. 3-sided:
  - a. **User** passes login/pass to **authentication service**
  - b. **Auth service** issues a token for **an app** to act on behalf of a user.
  - c. **Application** uses token to interact with **a service**.

**Data is still plaintext.**

3. **TLS/SSL** is used to establish **secure channel over TCP**. Uses asymmetric keys to build end-to-end encrypted communication (session save overhead!). Certificates are used as containers for keys + validation tool. HTTP over SSL = https://....:443/....



# HTML and DOM

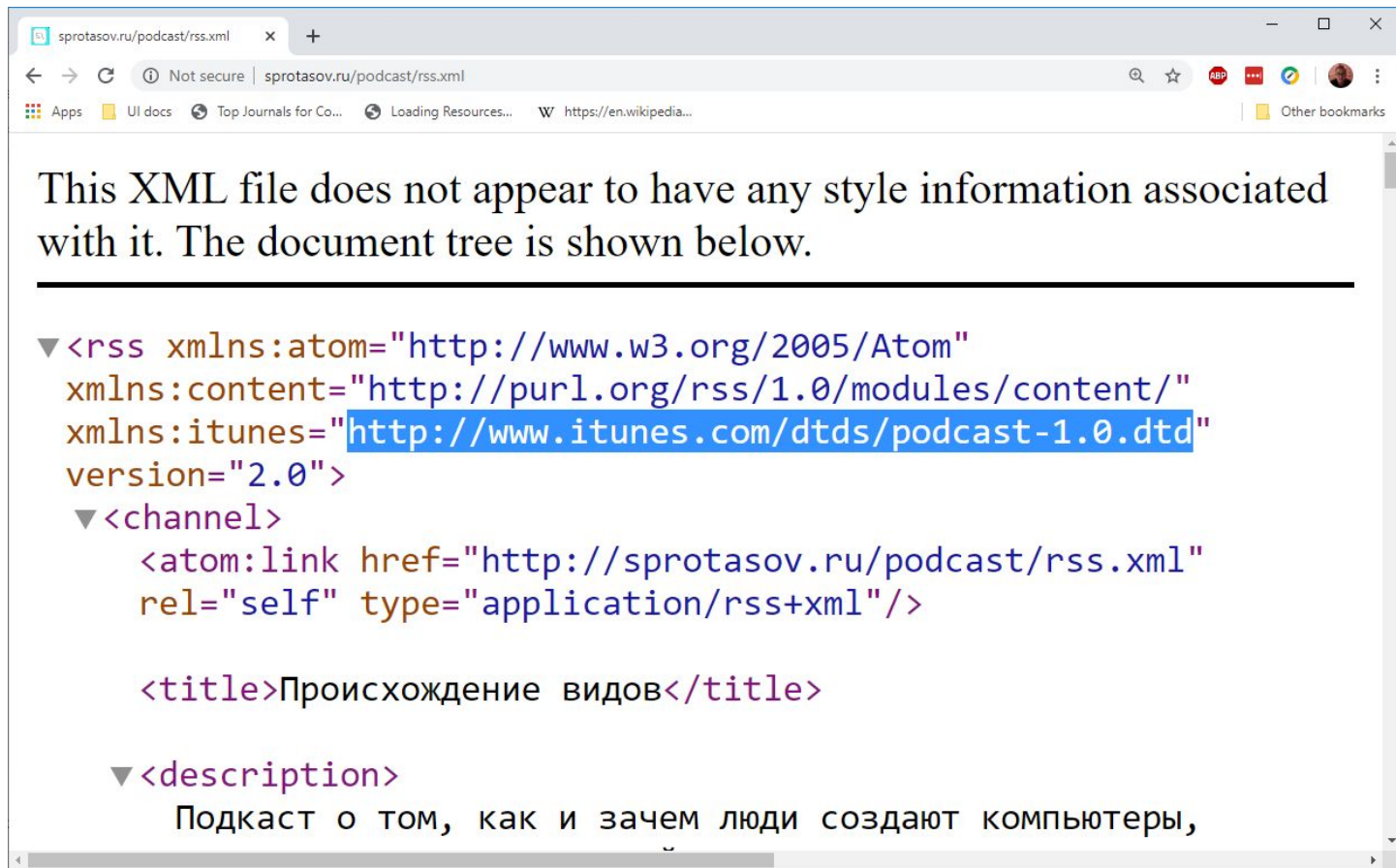
# LANGUAGE TYPOLOGY

- SGML – meta language for description of markup languages. It defines
  - Allowed symbol alphabet (SGML declaration)
  - DTD (data type definition) – markup syntax + semantics
- XML – simplified subset of SGML
  - XML Schema languages (DTD, W3C XSD)
- HTML – application of SGML (initially)
  - xHTML – application of XML

# SCHEMA LANGUAGES

- **DTD, XML Schema** – define document structure and node constraints
- Used for
  - Defining semantic rules (for values, number of children...)
  - Document pre-validation
- Document can be:
  - **type-valid** – meet all DTD constraints
  - **tag-valid** – meet all [SGML/XML] **tag** constraints

# RSS - Rich Site Summary



The screenshot shows a web browser window with the address bar displaying `sprotasov.ru/podcast/rss.xml`. The page content indicates that the XML file does not have associated style information and shows the document tree below. The tree structure is as follows:

- ▼ `<rss xmlns:atom="http://www.w3.org/2005/Atom" xmlns:content="http://purl.org/rss/1.0/modules/content/" xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd" version="2.0">`
  - ▼ `<channel>`
    - `<atom:link href="http://sprotasov.ru/podcast/rss.xml" rel="self" type="application/rss+xml"/>`
    - `<title>Происхождение видов</title>`
  - ▼ `<description>`
    - Подкаст о том, как и зачем люди создают компьютеры,

# DTD EXAMPLE

```
<!ELEMENT people_list (person*)>
<!ELEMENT person (name, birthdate?, gender?, socialsecuritynumber?)>
<!ELEMENT name (#PCDATA) >
<!ELEMENT birthdate (#PCDATA) >
<!ELEMENT gender (#PCDATA) >
<!ELEMENT socialsecuritynumber (#PCDATA) >
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE people_list SYSTEM "example.dtd">
<people_list>
  <person>
    <name>
      Fred Bloggs
    </name>
    <birthdate>
      27/11/2008
    </birthdate>
    <gender>
      Male
    </gender>
    <socialsecuritynumber>
      1234567890
    </socialsecuritynumber>
  </person>
</people_list>
```

```
<!-- Валидация простого HTML 4.01 -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
```

# XML SCHEMA (XSD) EXAMPLE

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="country_name" type="xs:string"/>
        <xs:element name="population" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<?xml version="1.0" encoding="utf-8"?>
<country>
  <country_name>France</country_name>
  <population>59.7</population>
</country>
```

# HTML5 DOCUMENT STRUCTURE

```
1  <!doctype html>
2  <html>
3  |   <head>
4  |       <title>my title</title>
5  |   </head>
6  |   <body>
7  |       body
8  |       <footer>
9  |           <!-- html5 specific -->
10 |           footer
11 |       </footer>
12 |   </body>
13 </html>
```

# \*ML-DOCUMENT PARSING METHODS

## **SAX** (Simple API for XML)

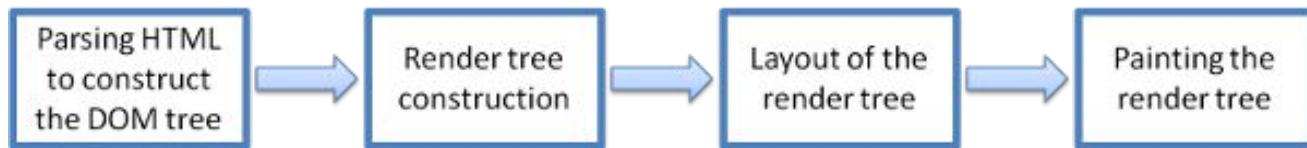
- Raises an event/error when new element (token) appears (considering document as stream of tokens and errors)
- Works either as
  - callback-methods (push) or
  - cursor (pull, StAX)
- Requires constant memory
  - Good for embedded systems
- Does not know anything about document's model

## **DOM** (**Document object model**, DOM tree)

- Creates full document model
- Used in browsers
- Unpredictable memory usage
  - [XML-bombs](#) using DTD
- Query languages (CSS-selectors, xpath, xquery)



# BROWSER ENGINES = LAYOUT ENGINE + JS + ...



- [Good article about browser architecture](#)
- Browser Layout Engine (html + css)
  - ~~Trident~~ (IE), "~~Edge~~" (Spartan) → Chromium (2019)
  - **Gecko** (Mozilla)
  - ~~WebKit~~ (Safari, Chromium-family), WebCore
    - **Blink** (Chrome 28+, Opera 15+, Chrome for Android)
  - Others (KHTML, ~~Presto~~)

# HTML TAGS

- Tag
  - **tag name** from HTML Schema - mandatory
    - `<td>..</td>`
  - **Closing tag** - mandatory
    - `<div> ... </div>`
    - ``
  - **Attributes** – define semantics
    - `<div id="div1" style="border:1px" class="myDiv">`  
`</div>`
  - Can have **inner tags** or **inner content** (text)
    - `<script> console.log(text); </script>`
    - `<div>`  
`<span> ... </span>`  
`<input />`  
`</div>`
  - Layout of the tag is defined by the **style**
    - In attribute
    - In CSS specification
    - By default

# STYLE

- inline-styles

```
<div  
  style="border: 1px solid gray; color: red"/>
```

- styles inside document

- `<head><style> ..... </style></head>`

- styles in separate CSS file

- `<link  
 rel="stylesheet" type="text/css"  
 href="xxx.css"/>`

# STYLE SYNTAX

```
some css_selector1  
{  
    property1: value1 ;  
    property2: value2 ;  
}
```

```
some css_selector2  
{  
    property3: value3 ;  
    property4: value4 ;  
}
```

# Small comment on http, html, and privacy

# Browser fingerprint: visited site

The screenshot shows a web browser with the Google Translate page open. The address bar shows the URL: `translate.google.com/#view=home&op=translate&sl=ru&tl=en&text=конкурентное%20преимущество`. The page is in Russian, and the interface is in English. The developer tools are open, showing the Network tab. A list of requests is displayed, with the first request selected. The request details show the following headers:

```
status: 200
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 0
```

The Request Headers section shows:

```
authority: translate.google.com
method: GET
path: /
scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding: gzip, deflate, br
accept-language: en-US;q=0.9,ru;q=0.8
cache-control: max-age=0
cookie: HSID=A_fjuBNIHhY6poq; SSID=Aq91QKTPYTLKIwmt; APISID=w7vJ7udEo60n_USY/AEjX3gYh7Pt1PeAo; SAPISID=geX17RjFNKvy5t7M/AKEp400h02v-31wmc; _SECURE-HSID=A_fjuBNIHhY6poq; __Secure-SSID=Aq91QKTPYTLKIwmt; __Secure-APISID=w7vJ7udEo60n_USY/AEjX3gYh7Pt1PeAo; __Secure-3PAPISID=geX17RjFNKvy5t7M/AKEp400h02v-31wmc; S=billing-ui-v3=ao5VWZh8RLCa-h0ky5vu-_jDA_luIak; billing-ui-v3-efe=ao5VWZh8RLCa-h0ky5vu-_jDA_luIak; _ga=GA1.3.70818346.1579711978; SID=uAeKO_AS1Edg-1oLPHjv3tkuZrFmSC2eEjFdRU4gkeQXIdnJ2HSREUe0mxcgFuaepkRkUA; __Secure-3PISID=uAeKO_AS1Edg-1oLPHjv3tkuZrFmSC2eEjFdRU4gkeQXIdnJ2HSREUe0mxcgFuaepkRkUA; SEARCH_SAME_SITE=CgQI1I8B; _gid=GA1.3.27975413.1582622833; NID=198-ljU-KjFv4PmxcvRkPvVvXm731dtaCxrGn757A2zoX-QtdsFDjd87xvX91Yv7W9wZ8Q0T:Ww0_e00_C2XukS037eqB5n7srT-Ot-zJ3uA00120T:K6nIS4qI8euTLbH-bAey7bKXN81XvsiD6U760TPVI-fwoYQROKmsOy2y2xFzFo3gm-GcCRV3xoyeQ-lw2Hc34H_JD01302Q7Uv357F6W6DvJpS5mWoo7H1qHMYR06pgkrf6FvICsEdv0EfdQ6yCB7XmVZRso; 1P_1AR=2020-02-25-14; SIDCC=AW0-TYsh23XRY1YmuI8f1l_HQK_bsbLuKc890N3I2H7359V9V5Xlgn8zNurL-f0mEz34U4qVvUQ
referer: https://translate.google.com/
sec-fetch-dest: document
sec-fetch-mode: navigate
sec-fetch-site: same-origin
sec-fetch-user: ?1
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.116 Safari/537.36
x-client-data: C162QE1PbJ4QJtscK8C1mSyE1q23QAQ13qso8CHuuygE16/KAQ19slo8C3e1yG1E7bXKAQ1Nuso8G16yG+
```

The bottom of the screenshot shows a notification from Google Translate: "Как изменится история переводов" (How the translation history will change). The notification states that in the future, access to the translation history will require logging in to an account, and the history will be moved to the "My Actions" page. It also mentions that the history will be deleted after updates, so users should save their translations.

## FingerprintJS: 99.5% accuracy [\[bonus: fresh leak of Safari\]](#)

- `navigator.userAgent`, `navigator.language`
- `new Date().getTimezoneOffset()`
- `screen.height`, `screen.width`, `screen.colorDepth`
- HTML5 features support (yes/no)
- `doNotTrack/DNT` flag (`blblblbl`), `cpuClass`, `platform`
- Installed extensions
- canvas fingerprint (draw on canvas and `toDataURL()`) — fonts depend on platform
- WebGL fingerprint (for iOS)
- Installed fonts

Google DNT - HTTP | MDN

https://www.google.com

Почта Картинки Войти

Инспектор Консоль Отладчик Сеть Стили

Поиск URL Отключить кэш Без ограничения

Все HTML CSS JS XHR Шрифты Изображения Медиа WS Прочее

Статус	Мет...	Домен	Файл	Инициатор	Тип	Передано	Раз...
200	GET	www.goo...	/	document	html	35,08 КБ	107,...
204	POST	www.goo...	gen_204?atyp=i&ei=g0jlYauoJYORrwsYz...	m=cDOS,cr,d...	html	357 Б	0 Б

26 запросов 1,33 МБ / 43,31 КБ передано Передано за: 39,41 с DOMContentLoaded: 238 мс load: 830 мс

Заголовки Куки Запрос Ответ Тайминги Защита

Поиск заголовков Блокировать Повторить отправку

Cookie: 1P\_JAR=2022-01-17-10; NID=511=iPBuZr8V\_F5R0F5ve2gCtbXBA5hb0LLrKExdV6WyGSdpaGRq1OWgCLN2ZtTS8K-nyoMEqcQ3wYKuvvp6N5J-SXiVfc6GvltIHCGff\_rJK4pGzUpdIrKjEW72ezdsK-UAYVxZyhXTXWRCrgp8sxkh5x9yNEUJY8\_dhj3KdSaKj8; ANID=AHWqTUmz4IB8UqFqx6LSaPbVkuDnP-wBAHlyjwOXMxH6a7Mj9PiK2PstShMRI64W

DNT: 1

Host: www.google.com

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate



# Crawling problems ... and solutions

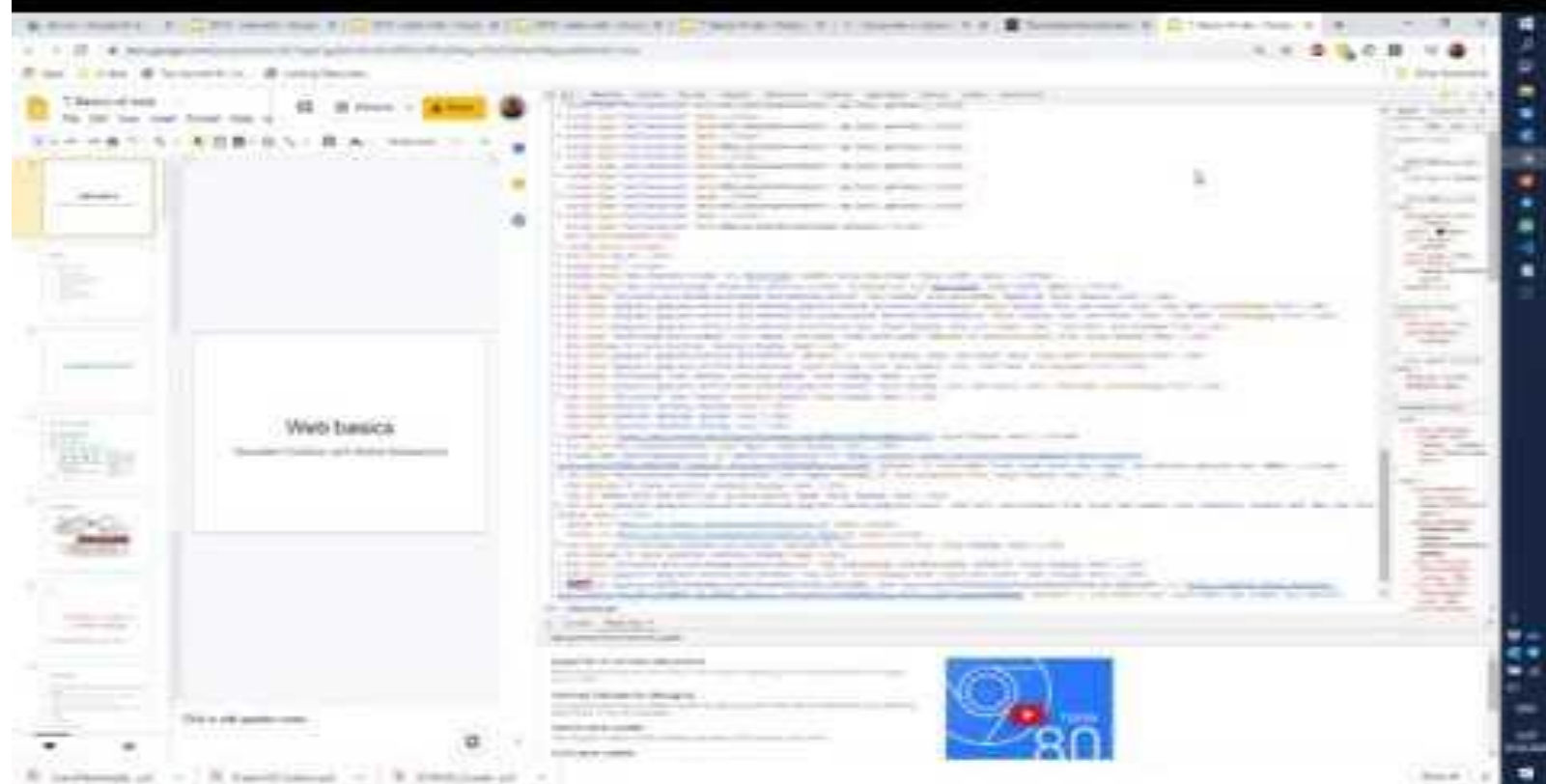
# Problem 1: SaaS vs Documents

On your \*nix machines run:

```
wget --no-check-certificate -O doc.txt http://tiny.cc/00dhkz
```

```
cat doc.txt | sed -e "s/;/;\n/g" | grep "QuadTree search"
```

What's wrong?



# How to?

You need a browser engine + JS

1. [Headless] browsers
2. Drivers to manage browsers
3. Automation software:
  - a. [Selenium](#)
  - b. [Puppeteer](#)

# Problem 2: But wait... software engineering?!



How would you parse all links from the Wikipedia article page?

```
import requests
from bs4 import BeautifulSoup
...
```

**Stop here!** Every big **company knows** that you will parse it's data. It wants to minimize harm you can do. **APIs**! Free anonymous, free authenticated, paid.

1. [Wikipedia API](#)
2. [VK API](#)
3. [Yandex Search API](#)
4. [Google Open Search API](#)
5. ...

# Problem 3. I was downloading ... but it stopped working

1. Company **considers** the data (and service) it has as:
  - a. A property
  - b. A competitive advantage
2. Thus, company **protects** its data from grabbing (and services from proxying):
  - a. With API regulations ([Ya](#), ...) — what is the allowed rate
  - b. Etiquette ([Wiki](#))
  - c. Access keys to control grabbing and proxying speed
  - d. Special legal statements that prohibit grabbing ([ASOS](#))
3. To **enforce** you to obey
  - a. Access key restriction
  - b. IP [range] blocking
  - c. Browser **fingerprint** blocking
4. So, to speed or just enable you crawling ...

# Problem 4: The last but not the least...

## Allow and Disallow

- [robots.txt](#) prohibits

- <http://innopolis.ru/robots.txt>
- <https://yandex.ru/robots.txt>
- Wiki :)

```
# Sorry, wget in its recursive mode is a frequent problem.  
# Please read the man page and use it properly; there is a  
# --wait option you can use to set the delay between hits,  
# for instance.  
#  
User-agent: wget  
Disallow: /
```

- Also,

```
Sitemap: https://yandex.ru/znatoki/sitemap/sitemap-index.xml
```

- [sitemap.xml](#) helps. Start with robots : <https://www.kinopoisk.ru/robots.txt>
  - Sitemap: <https://www.kinopoisk.ru/sitemaps/sitemap.xml>

Crawl safe!

