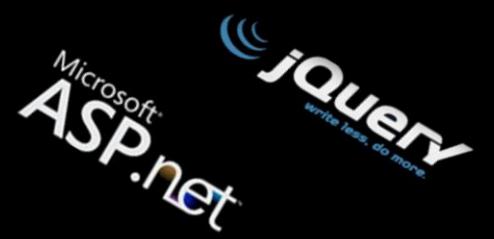


PHP Web Technologies



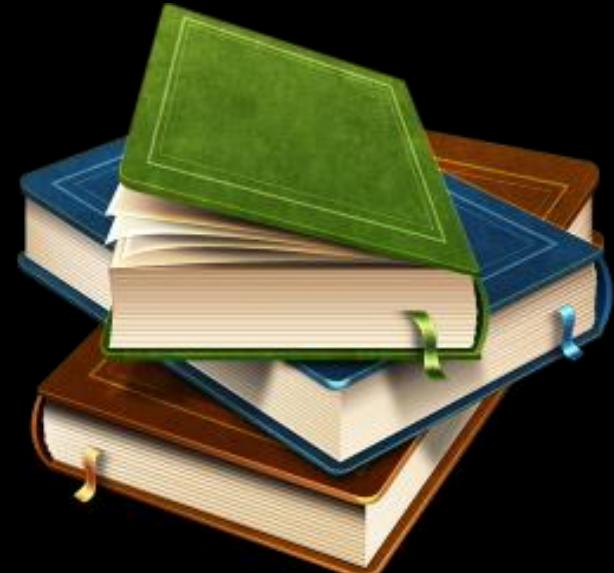
Mario Peshev
Technical Trainer
<http://peshev.net/>
Software University
<http://softuni.bg>



How can we connect web technologies with PHP

Table of Contents

1. WWW and URL
2. HTML, XML, JSON, RSS
3. HTTP Protocol: HTTP Request, HTTP Response
4. HTTP Cookies
5. AJAX Requests
6. Web Developer Tools





WWW and URL

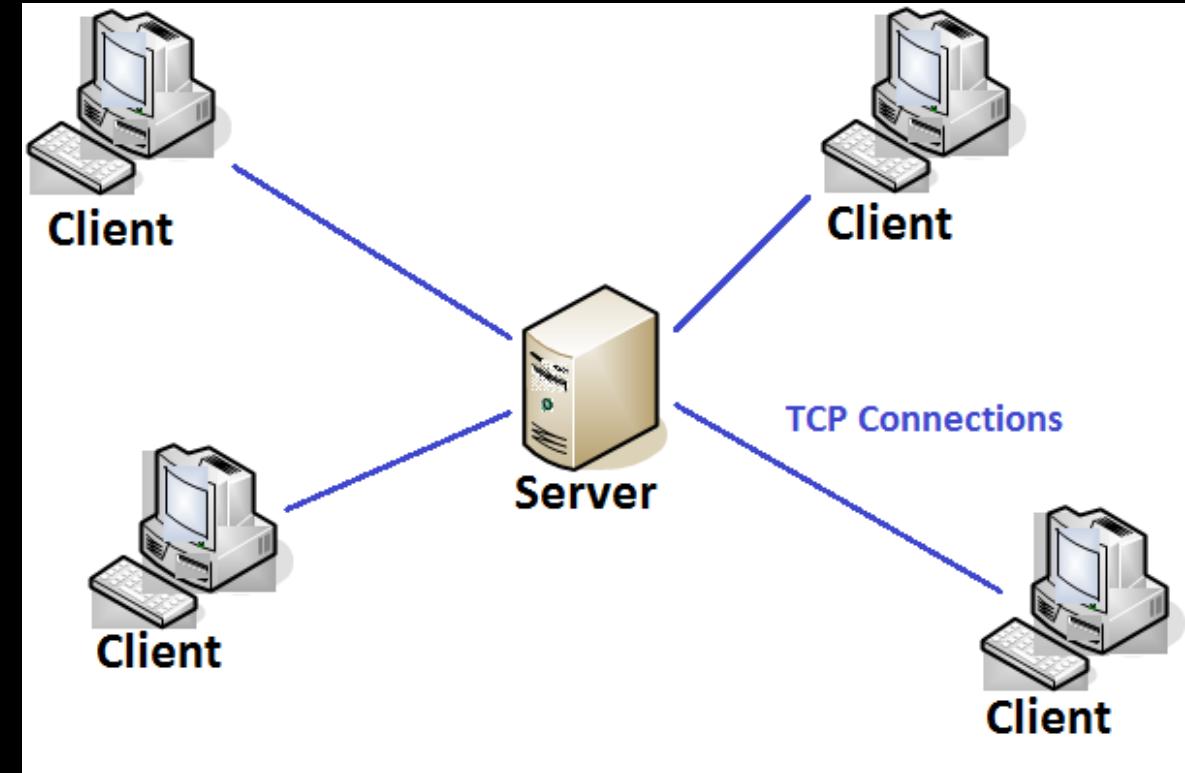
What are these?

What is WWW?

- WWW = World Wide Web = Web
 - Global distributed information system in Internet
 - A service in Internet (like E-mail, DNS, ...)
 - Consists of set of documents (and other resources) located on different Internet servers
 - Accessed through standard protocols like HTTP, HTTPS and FTP by their URL
 - **Web servers** provide Web content
 - **Web browsers** display the Web content

WWW Components

- Structural components
 - Internet – provides data transfer channels over TCP, HTTP and other protocols
 - Clients (Web browsers)
 - display Web content
 - Web servers
 - IIS, Apache, Tomcat, GWS, etc.



WWW Components

- Semantic components
 - Hyper Text Transfer Protocol (**HTTP**)
 - Hyper Text Markup Language (**HTML**)
 - Uniform Resource Identifiers (**URIs**)
 - Uniform Resource Locator (**URL**)
 - Uniform Resource Name(**URN**)



WWW Infrastructure

- Clients use Web browser application to request resources from the Web servers via HTTP
 - Resources have unique URL address
- Servers send the requested resource as a response
 - Or reply with an error message
- Web pages are resources in WWW
 - HTML text, graphics, animations and other files
- Web sites are sets of Web pages in WWW

WWW Infrastructure (2)

- Client's browser renders Web pages returned by the Web servers
 - Pages are in HTML (Hyper Text Markup Language)
 - Browsers shows the text, graphics, sounds, etc.
 - HTML pages contain hyperlinks to other pages
- The entire WWW system runs over standard networking protocols
 - TCP, DNS, HTTP, FTP, etc.
- The HTTP protocol is fundamental for WWW

What is URL?

- Uniform Resource Locator (URL)
 - Unique resource location in WWW, e.g.

```
https://softuni.bg/Search/Results?query/php
```
- It is just a formatted string, consisting of:
 - Protocol for communicating with the server (e.g., http, ftp, https)
 - Name of the server or IP address + optional port
(e.g. www.softuni.bg, mail.bg:8080)
 - Path and name of the resource (e.g. Search/Results)
 - Parameters (optional, e.g. ?query=php)

URL Encoding

- URLs are encoded according RFC 1738:

“... Only alphanumeric [0-9a-zA-Z], the special characters \$-_.+!*'() and reserved characters used for their reserved purposes may be used unencoded within an URL.”

- All other characters are escaped with the formula:

`%[character hex code in ISO-Latin character set]`

- Space can also be encoded as "+"

URL other characters - Examples

- URLs are encoded according RFC 1738 in UTF-8

Character	Decimal code	Hex code	URL display
“ ”	32	20	%20
“{”	123	7B	%7B
“}”	125	7D	%7D
“/”	47	2F	%2F
“\”	92	5C	%5C
“а”	53424	D0B0	%D0%Б0
“я”	53647	D18F	%D1%8F

- Chrome and Firefox translates the URL on copy

URL - Examples

- Some valid URLs:

```
https://www.google.bg/webhp?sourceid=chrome-  
instant&ion=1&espv=2&ie=UTF-8#q=http%20get%20vs%20post
```

```
https://softuni.bg/trainings/fasttracks/details/1033
```

Should be: %D0%BF%D1%80%D0%BE%D0%B3%D1%80
%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5

- Some invalid URLs:

```
http://www.kaldata.com/forums/forum/101-програмиране/
```

Should be: %D1%81%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80

```
http://obekti.point.bg/search/софтуер
```



HTML, XML, JSON, RSS

Comparing the Common Web Data Formats

- Hyper Text Markup Language (HTML)
 - Notation for describing formatted text with images and hyperlinks
 - Interpreted and displayed by the Web browsers
- A Web (HTML) page consists of:
 - HTML file
 - CSS stylesheet file (optional)
 - A bunch of images (optional)
 - Other resources (optional)



- HTML is straight-forward and easy to learn
 - HTML documents are plain text files
 - Easy to add formatting, hyperlinks, bullets, etc.
 - Images can be added as separate files
 - Can be automatically generated by authoring programs
 - Tools to help users creating HTML pages
 - E.g. FrontPage, Dreamweaver, Visual Studio
 - WYSIWYG HTML editors

HTML - Example

```
<!DOCTYPE html>
<html>
  <head><title>HTML Example</title></head>
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>
    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>
    <div align="center"
          style="background:skyblue">
      This is a div</div>
  </body>
</html>
```

- XML is markup-language for encoding documents in machine-readable form
 - Text-based format
 - Consists of tags, attributes and content
 - Provide data and meta-data in the same time



```
<?xml version="1.0"?>
<library>
    <book><title>HTML 5</title><author>Bay Ivan</author></book>
    <book><title>WPF 4</title><author>Microsoft</author></book>
    <book><title>WCF 4</title><author>Kaka Mara</author></book>
    <book><title>UML 2.0</title><author>Bay Ali</author></book>
</library>
```

JSON

- JSON (JavaScript Object Notation)
 - Standard for representing simple data structures and associative arrays
 - Lightweight text-based open standard
 - Derived from the JavaScript language



```
{  
    "firstName": "Svetlin", "lastName": "Nakov", "age": 34,  
    "address": { "streetAddress": "26, Vassil Kunchev str.",  
                "city": "Sofia", "postalCode": "10021" },  
    "phoneNumber": [{ "type": "home", "number": "212 555-1234"},  
                  { "type": "fax", "number": "646 555-4567" }]  
},  
{ "firstName": "Gosho", "lastName": "Goshov", "age": 79 }
```

RSS

- RSS (Really Simple Syndication)
 - Family of Web feed formats for publishing frequently updated works
 - E.g. blog entries, news headlines, videos, etc.
 - Based on XML, with standardized XSD schema
- RSS documents (feeds) are list of items
 - Each containing title, author, publish date, summarized text and metadata
 - Atom protocol aimed to enhance / replace RSS



RSS - Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
    <title>W3Schools Home Page</title>
    <link>http://www.w3schools.com</link>
    <description>Free web building tutorials</description>
    <item>
        <title>RSS Tutorial</title>
        <link>http://www.w3schools.com/rss</link>
        <description>New RSS tutorial on W3Schools</description>
    </item>
    <item>
        <title>XML Tutorial</title>
        <link>http://www.w3schools.com/xml</link>
        <description>New XML tutorial on W3Schools</description>
    </item>
</channel>
</rss>
```

The HTTP Protocol

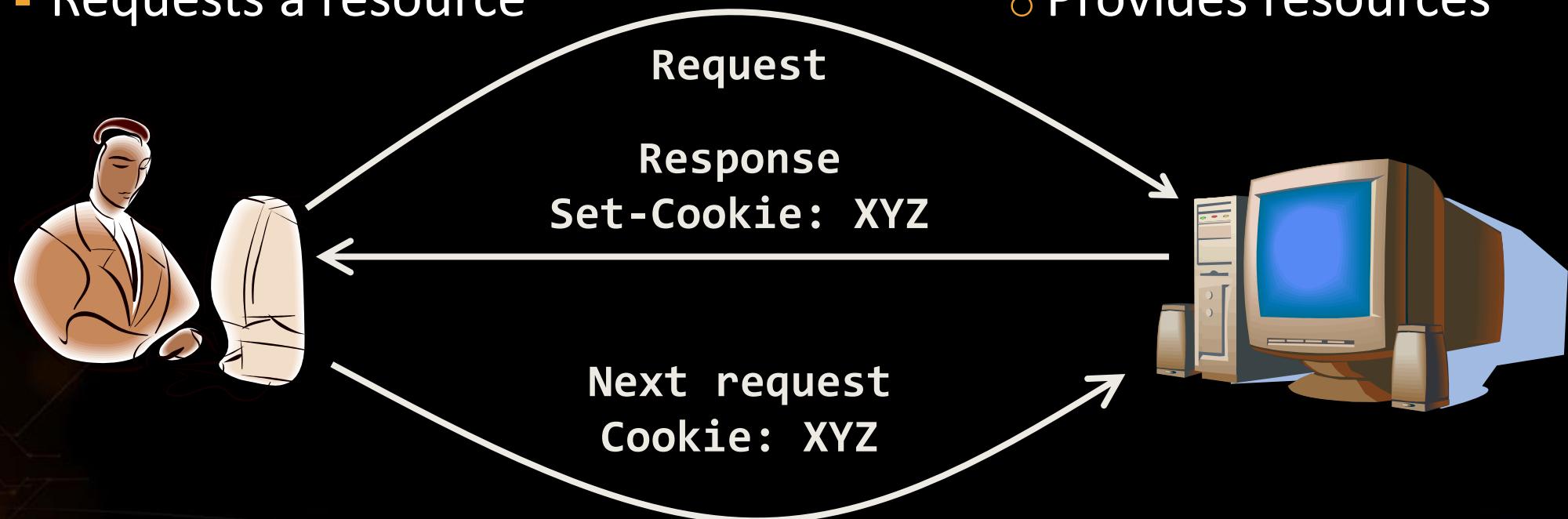
How HTTP Works?



- Hyper Text Transfer Protocol (HTTP)
 - Client-server protocol for transferring Web resources (HTML files, images, styles, etc.)
- Important properties of HTTP
 - Request-response model
 - Text-based format
 - Relies on a unique resource URLs
 - Provides resource metadata (e.g. encoding)
 - Stateless (cookies can overcome this)

HTTP: Request-Response Protocol

- Client program
 - Running on end host
 - E.g. Web browser
 - Requests a resource
- Server program
 - Running at the server
 - E.g. Web server
 - Provides resources



HTTP - Example

■ HTTP request:

```
GET / HTTP/1.1
Host: softuni.bg
User-Agent: Mozilla/5.0
```

The empty line
denotes the end of
the request header

■ HTTP response:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/8.5
Date: Thu, 17 Jul 2014 12:11:44 GMT
Content-Length: 8560

<!DOCTYPE html>
...
```

The empty line
denotes the end of
the response header

HTTP Request Methods

- HTTP request methods:
 - GET
 - Return the specified resource, run a program at the server or just download file
 - HEAD
 - Return the meta-data associated with a resource (headers only)
 - POST
 - Update a resource, provide input data for processing at the server

HTTP Response Codes

- HTTP response code classes
 - 1xx: informational (e.g., “100 Continue”)
 - 2xx: success (e.g., “200 OK”)
 - 3xx: redirection (e.g., “304 Not Modified”, "302 Found")
 - 4xx: client error (e.g., “404 Not Found”)
 - 5xx: server error (e.g., “503 Service Unavailable”)
- "302 Found" is used for redirecting the Web browser to another URL

HTTP Request Message

- Request message sent by a client consists of
 - Request line – request method (GET, POST, HEAD), resource URI, and protocol version
 - Request headers – additional parameters
 - Body – optional data
 - E.g. posted form data, files, etc.

```
<request method> <resource> HTTP/<version>
<headers>
<empty line>
<body>
```

HTTP GET Request - Example

GET / HTTP/1.1

Host: softuni.bg

Connection: keep-alive

Cache-Control: max-age=0

Accept:

text/html,application/xhtml+xml,application/xml;
q=0.9,image/webp,*/*;q=0.8

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64)

AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/36.0.1985.103 Safari/537.36

Accept-Encoding: gzip,deflate,sdch

Accept-Language: bg,en-US;q=0.8,en;q=0.6

HTTP request line

HTTP headers

The request body
is empty

HTTP POST Request - Example

POST /account/login HTTP/1.1

HTTP request line

Host: softuni.bg

Connection: keep-alive

Content-Length: 200

Cache-Control: max-age=0

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Origin: https://softuni.bg

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.103 Safari/537.36

Content-Type: application/x-www-form-urlencoded

Referer: https://softuni.bg/account/authenticate

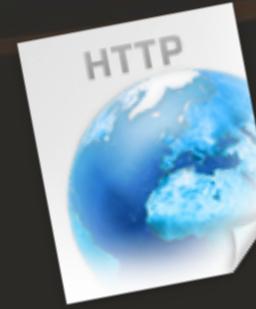
HTTP headers

__RequestVerificationToken=D9303irA706r-tT2hHxLhiX8hpfo7bnc0YcvGqGRRmP-PpSh-fgAtl9fJ04ybeNzyQ2EVRp95kX_J35hF10w9G7RpJfyDeqi20Neu8yAmqc1&LoginUserName=Gosho&LoginPassword=gosho1sTheBest&LoginRememberMe=false

The request body contains the submitted form data

Conditional HTTP GET Request - Example

```
GET /Scripts/lazyPainter.js HTTP/1.1
Host: softuni.bg
Connection: keep-alive
Cache-Control: max-age=0
Accept: */*
If-None-Match: "0c5f1b54095cf1:0"
If-Modified-Since: Tue, 01 Jul 2014 15:25:38 GMT
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/36.0.1985.103 Safari/537.36
Referer: https://softuni.bg/
```



- Fetches the resource only if it has been changed at the server
 - Server replies with “304 Not Modified” if the resource has not been changed
 - Or “200 OK” with the latest version otherwise

HTTP Response Message

- Response message sent by the server
 - Status line – protocol version, status code, status phrase
 - Response headers – provide meta data
 - Body – the contents of the response (the requested resource)

```
HTTP/<version> <status code> <status text>
<headers>
<empty line>
<response body - the requested resource>
```

HTTP Response - Example

HTTP/1.1 200 OK

Cache-Control: private

Content-Type: text/html; charset=utf-8

Content-Encoding: gzip

Vary: Accept-Encoding

Server: Microsoft-IIS/8.5

X-Frame-Options: SAMEORIGIN

X-Frame-Options: SAMEORIGIN

Date: Thu, 17 Jul 2014 12:11:44 GMT

Content-Length: 8560

<!DOCTYPE html>

...

HTTP response
status line

HTTP response
headers

HTTP response
body

HTTP Response Error - Example

HTTP/1.1 404 Not Found

Server: nginx/1.2.4

Date: Thu, 17 Jul 2014 12:49:11 GMT

Content-Type: text/html; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

Vary: Accept-Encoding

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0

Pragma: no-cache

X-FRAME-OPTIONS: SAMEORIGIN

Content-Encoding: gzip

<h2>Error</h2>

HTTP response
status line

HTTP response
headers

HTTP response
body

Browser Redirection - Example

- HTTP browser redirection example
 - HTTP GET requesting a moved URL:

```
GET / HTTP/1.1
Host: www.thefacebook.com
Connection: keep-alive
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.103 Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: bg,en-US;q=0.8,en;q=0.6
```

Browser Redirection - Example

- HTTP browser redirection example
 - The HTTP response says the browser should request another URL:

```
HTTP/1.1 301 Moved Permanently
Location: http://www.facebook.com/
Pragma: no-cache
Cache-Control: private, no-cache, no-store, must-revalidate
Expires: Sat, 01 Jan 2000 00:00:00 GMT
Content-Type: text/html; charset=utf-8
X-FB-Debug: psDcX9Hme6xz95MICUIL4hBx38gp+
JK0HV3fiEhdqNBaycrgcsoNl3lQ8sZT0wuBcV0K/16+L0xy/VZMUOoqxQ==
Date: Thu, 17 Jul 2014 13:17:46 GMT
Connection: keep-alive
Content-Length: 0
```

Content-Type and Disposition

- The Content-Type header at the server specifies how the output should be processed
- Examples:

UTF-8 encoded HTML page.
Will be shown in the browser.

Content-Type: text/html; charset=utf-8

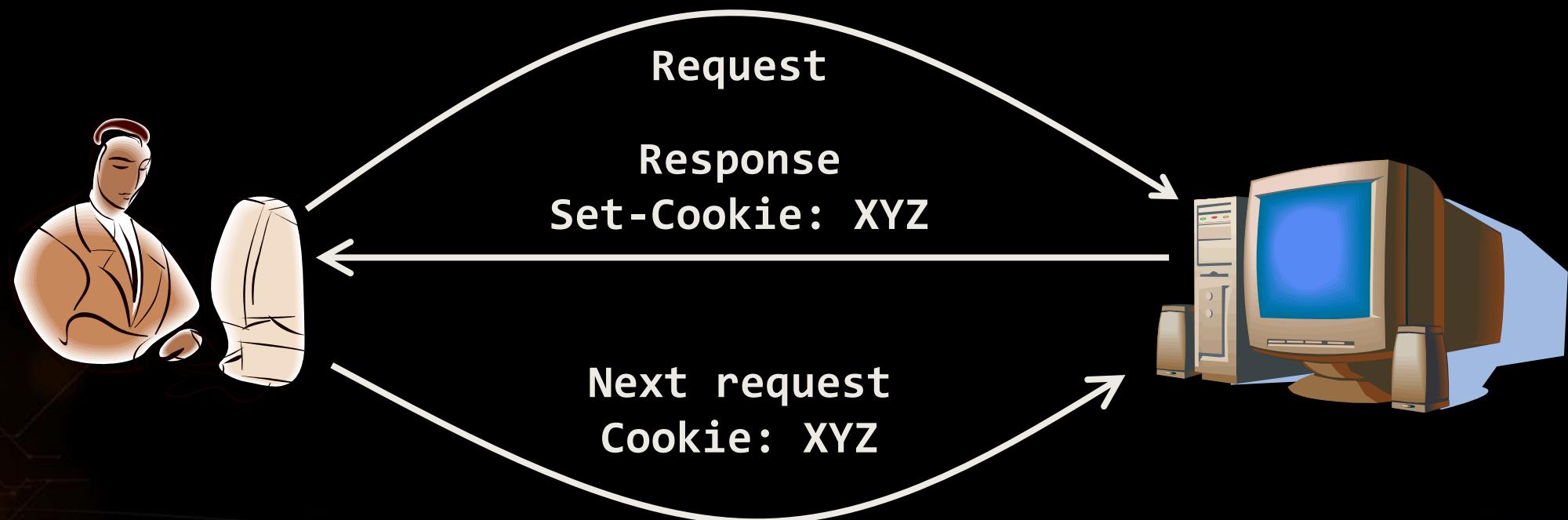
Content-Type: application/pdf
Content-Disposition: attachment;
filename="Financial-Report-April-2010.pdf"

This will download a PDF file named
Financial-Report-April-2010.pdf

HTTP Cookies

Cookie

- Cookies are pieces of data, stored by the client on behalf of the server
- Included in all future HTTP requests to the server



HTTP Cookies - Example

- The client requests some URL:

```
GET / HTTP/1.1
Host: www.google.bg
```

- The server sets a cookie in the HTTP response:

```
HTTP/1.1 200 OK
Set-Cookie:
"PREF=ID=fb2c28f98c46c0de:FF=0:TM=1405603577:LM=1405603577:S=t5-
0GoiYm3dWNH7J; expires=Sat, 16-Jul-2016 13:26:17 GMT; path=/;
domain=.google.bg NID=67=stap-pci4zxItxiLp6S3IKimA1KYTfLg5z9xu
D4AelWcxGfc3i0oressSGNFsIF6V70VRYs7bPsc5F5X76shUvCknYVEJUkteSC3Qg9Wx_mmv
vMUIprBfNiwBwnONQp_H; expires=Fri, 16-Jan-2015 13:26:17 GMT; path=/;
domain=.google.bg; HttpOnly"
```

HTTP Cookies - Example

- In further requests to google.bg the Web browser sends the cookie in the HTTP header:

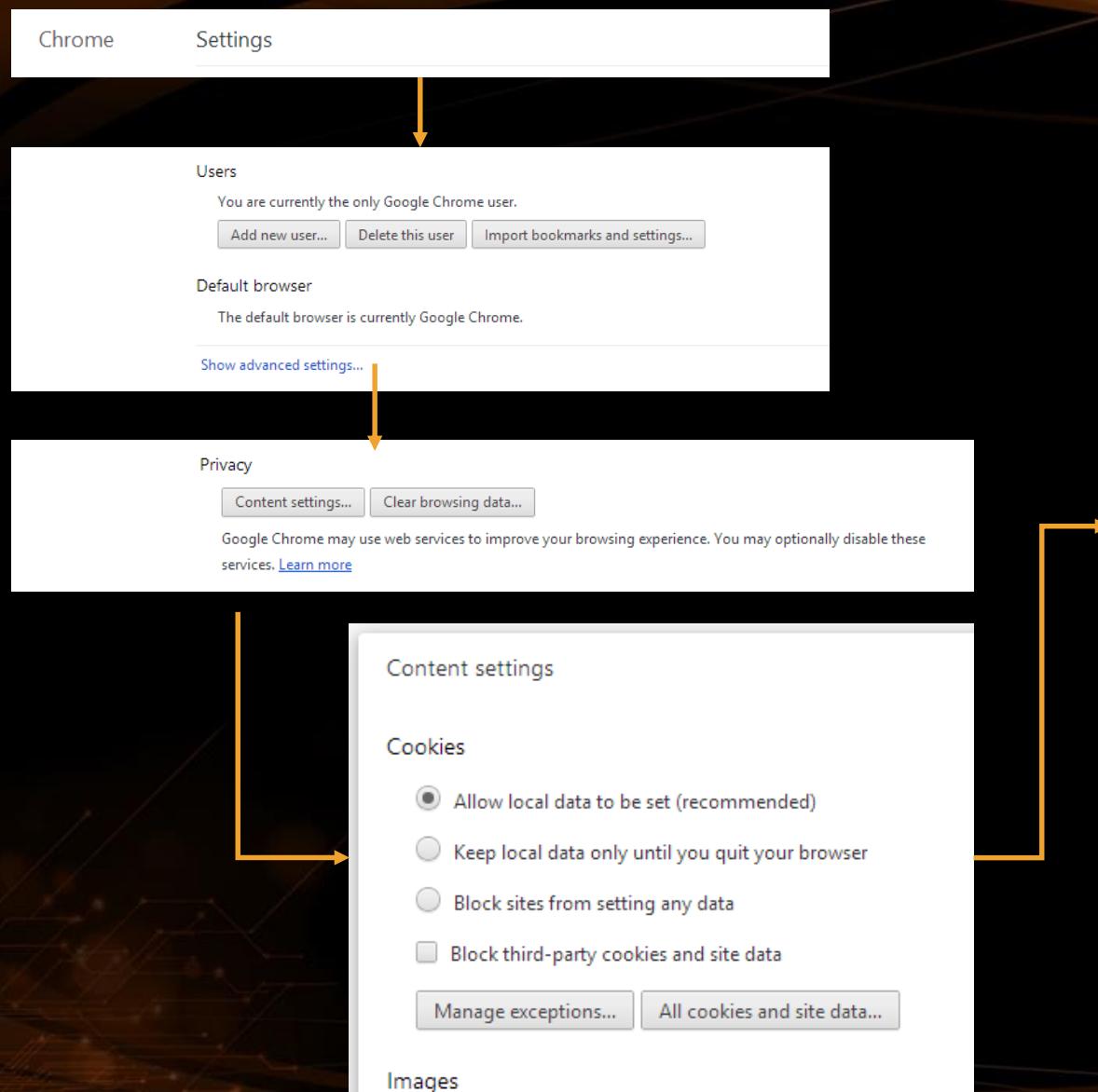
```
GET / HTTP/1.1
```

```
Host: www.google.bg
```

```
Cookie:
```

```
PREF=ID=fb2c28f98c46c0de;U=3c5c60a95dfc8680;FF=0;TM=1405603577;  
LM=1405603611;S=lemd1UUEBAf0ApYB; NID=67=stap-  
pci4zxItxiLp6S3IKimAlKYTfLg5z9xuD4Ae1WcxGfc3i0oresSGNFsIF6V70VR  
Ys7bPsC5F5X76shUvCknYVEJUkteSC3Qg9Wx_mmvvMUIprBfNiwBwnONQp_H;  
OGPC=5-2;
```

View HTTP Cookies in the Web Browser



The screenshot shows the Google Chrome Settings interface. A yellow arrow points from the top navigation bar down to the 'Privacy' section. Another yellow arrow points from the 'Privacy' section down to the 'Content settings...' button. A third yellow arrow points from the 'Content settings...' button down to the 'Cookies' section of the 'Content settings' dialog.

Chrome Settings

Users
You are currently the only Google Chrome user.
[Add new user...](#) [Delete this user](#) [Import bookmarks and settings...](#)

Default browser
The default browser is currently Google Chrome.
[Show advanced settings...](#)

Privacy
[Content settings...](#) [Clear browsing data...](#)
Google Chrome may use web services to improve your browsing experience. You may optionally disable these services. [Learn more](#)

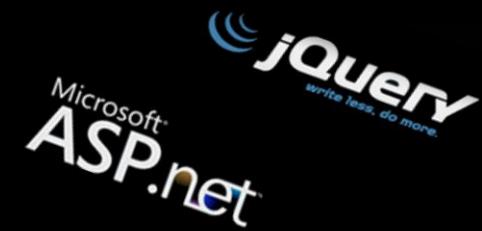
Content settings
Cookies
 Allow local data to be set (recommended)
 Keep local data only until you quit your browser
 Block sites from setting any data
 Block third-party cookies and site data
[Manage exceptions...](#) [All cookies and site data...](#)

Images

Cookies and site data	
Site	Locally stored data
www.1001freefonts.com	Local storage
172.16.0.1	1 cookie
23andme.com	8 cookies
www.23andme.com	6 cookies, Local storage
24hoursofhappy.com	Local storage
254a.com	1 cookie
2mdn.net	Channel ID
2o7.net	3 cookies
112.2o7.net	2 cookies
audible.112.2o7.net	1 cookie
oracle.112.2o7.net	1 cookie
sonyglobal.112.2o7.net	1 cookie
eventbrite.122.2o7.net	1 cookie
ad.360yield.com	3 cookies
ads.3bay.bg	6 cookies
5min.com	1 cookie

AJAX

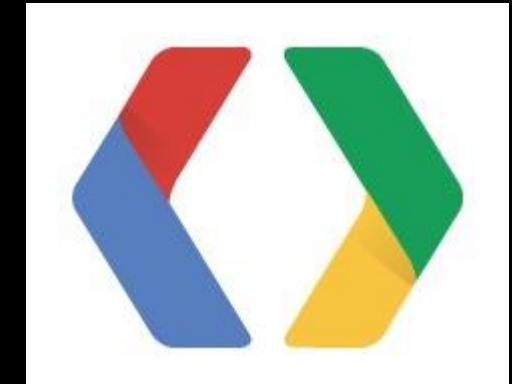
Asynchronous JavaScript and XML



- AJAX is acronym of Asynchronous JavaScript and XML
 - Technique for background loading of dynamic content and data from the server side
 - Allows dynamic client-side changes
- Two styles of AJAX
 - Partial page rendering – loading of HTML fragment and showing it in a <div>
 - JSON service – loading JSON object and client-side processing it with JavaScript / jQuery

Web Developer Tools (1)

- Chrome DevTools
 - Web authoring and debugging tools built into Google Chrome
 - The easiest and most commonly used tool for Web Developers
 - Monitors network activity, source code and connections between files
 - Can add JavaScript break points for inspection
 - Provides inbuilt console



Web Developer Tools (2)

- Firebug plug-in for Firefox
 - One of the best tools for Web developers
 - The ultimate tool for monitoring, editing and debugging HTTP, HTML, CSS, JavaScript, etc.
 - Free, open-source – www.getfirebug.com
- Fiddler – HTTP proxy
 - Intercepts the HTTP traffic
 - Analyzes the HTTP conversation
 - Free tool – www.fiddler2.com



Web Developer Tools (3)

- Wireshark packet analyzer
 - Low-level packet sniffer
 - Intercepts the entire IP network traffic
 - Can reconstruct the HTTP conversation
 - Can intercept any (unencrypted) protocol
 - IP, ICMP, TCP, UDP, HTTP, DNS, SMTP, POP3
 - Can intercept passwords sent in clear-text
 - Free, open-source project – www.wireshark.org



Summary

- WWW and URL
- HTML, XML, JSON, RSS
- HTTP Protocol:
 - HTTP Request, HTTP Response
- HTTP Cookies
- AJAX Requests
- Web Developer Tools



Questions?



License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license



- Attribution: this work may contain portions from
 - "Fundamentals of Computer Programming with C#" book by Svetlin Nakov & Co. under CC-BY-SA license
 - "C# Part I" course by Telerik Academy under CC-BY-NC-SA license

Free Trainings @ Software University

- Software University Foundation – softuni.org
- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University @ YouTube
 - youtube.com/SoftwareUniversity
- Software University Forums – forum.softuni.bg

