

AN INTRODUCTION TO NETCENTRIC DEVELOPMENT

HTTP, REST API, Tools

COS216
AVINASH SINGH
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PRETORIA

IP CLASSES

Class	Address range	Supports
Class A	1.0.0.1 to 126.255.255.254	Supports 16 million hosts on each of 127 networks.
Class B	128.1.0.1 to 191.255.255.254	Supports 65,000 hosts on each of 16,000 networks.
Class C	192.0.1.1 to 223.255.254.254	Supports 254 hosts on each of 2 million networks.
Class D	224.0.0.0 to 239.255.255.255	Reserved for multicast groups.
Class E	240.0.0.0 to 254.255.255.254	Reserved for future use, or research and development purposes.

IP CLASSES

IP ADDRESS

A

CLASS

B

C

D



WHY 127.0.0.1?

- 127 is the last network number in a class A network with a subnet mask of 255.0.0.0.
- 127.0.0.1 is the first assignable address in the subnet.
- 127.0.0.0 cannot be used because that would be the wire number (network address).

HYPertext Transfer Protocol



Hypertext Transfer
Protocol (HTTP)



Most commonly used
protocol on the web



Application protocol
for transmitting
hypermedia

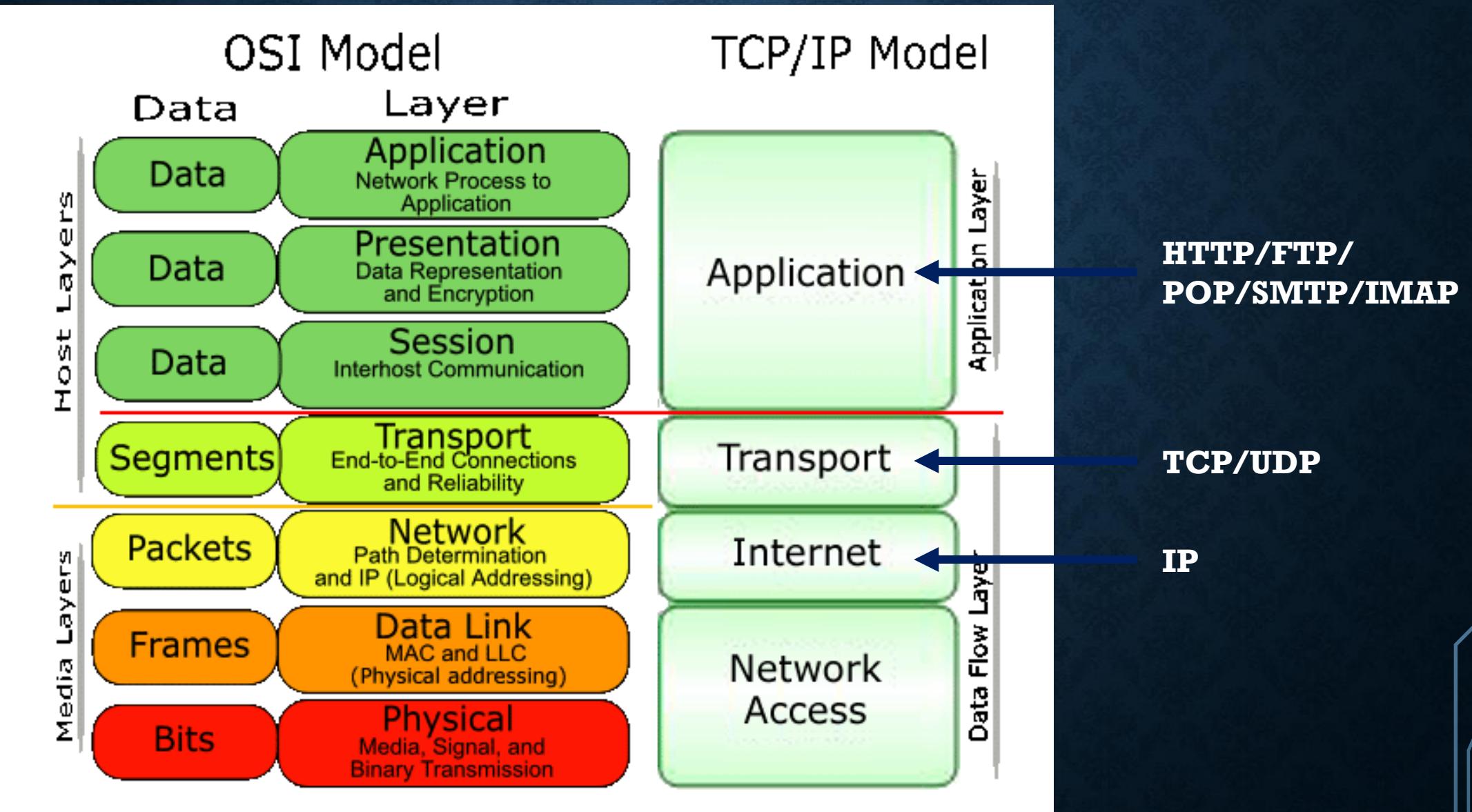


Initiated by Tim
Berners-Lee

HYPertext Transfer Protocol

- HTTP/0.9
 - Created in 1991
- HTTP/1.0
 - Created in 1993
- HTTP/1.1
 - Created in 1997
 - Still most widely used version
- HTTP/2.0
 - Standardized in 2015
- HTTP/3
 - Uses the QUIC protocol

HYPertext Transfer Protocol



HYPertext Transfer Protocol

- HTTP assumes a reliable transmission protocol (TCP or UDP)
- HTTP is not concerned with lower network layers, it assumes they work
- HTTP is stateless
- Stateless protocols do not save states/variables/memory between requests
- HTTP Secure (HTTPS) runs over a secure/encrypted connection
- Uses SSL or TLS which utilizes RSA encryption

HTTP TERMS

- **RFC:** The IETF standard (request for comment)
- **Request Has Body:** Whether or not you send a body of data to the server
- **Response Has Body:** Whether or not you receive a body of data from the server
- **Safe:** By convention safe because it does not change data or state on the server
- **Idempotent:** multiple identical requests should have the same effect as a single request
- **Cacheable:** A response that can be saved, and retrieved and used at a later stage

HTTP METHODS

HTTP Method	RFC	Request Has Body	Response Has Body	Safe	Idempotent	Cacheable
GET	RFC 7231	Optional	Yes	Yes	Yes	Yes
HEAD	RFC 7231	No	No	Yes	Yes	Yes
POST	RFC 7231	Yes	Yes	No	No	Yes
PUT	RFC 7231	Yes	Yes	No	Yes	No
DELETE	RFC 7231	No	Yes	No	Yes	No
CONNECT	RFC 7231	Yes	Yes	No	No	No
OPTIONS	RFC 7231	Optional	Yes	Yes	Yes	No
TRACE	RFC 7231	No	Yes	Yes	Yes	No
PATCH	RFC 5789	Yes	Yes	No	No	No

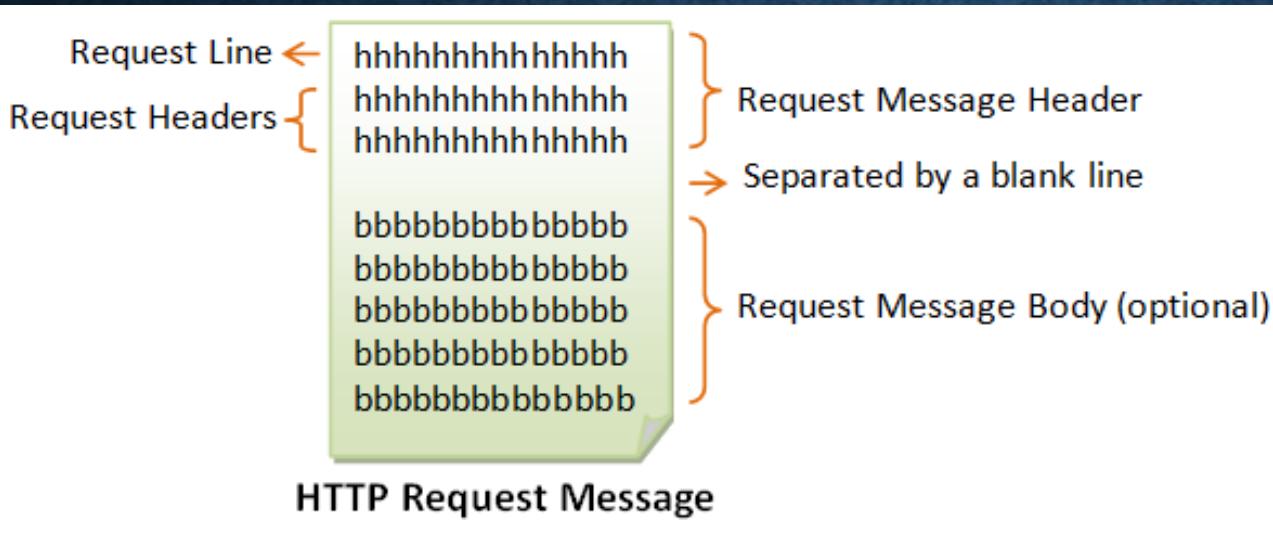
HTTP REQUEST HEADERS

- Headers can be added by the client to a HTTP request
- Provides additional client information to the server
- Can be used by the server to reject requests, provide customized responses, or log the information for statics
- These headers are automatically set by the browser
- Many headers exists, only the most important ones are listed here

HTTP REQUEST HEADERS

Header Name	Description	Example
Accept	Media type(s) that is(/are) acceptable for the response. See Content negotiation	Accept: text/plain
Accept-Charset	Character sets that are acceptable	Accept-Charset: utf-8
Accept-Encoding	List of acceptable encodings. See HTTP compression	Accept-Encoding: gzip, deflate
Accept-Language	List of acceptable human languages for response. See Content negotiation	Accept-Language: en-US
Cookie	An HTTP cookie previously sent by the server with Set-Cookie (below)	Cookie: \$Version=1; Skin=new;
Content-Length	The length of the request body in octets (8-bit bytes)	Content-Length: 348
Content-Type	The Media type of the body of the request (used with POST and PUT requests)	Content-Type: application/x-www-form-urlencoded
Date	The date and time that the message was originated (in "HTTP-date" format as defined by RFC 7231 Date/Time Formats)	Date: Tue, 15 Nov 1994 08:12:31 GMT
Host	The domain name of the server (for virtual hosting), and the TCP port number on which the server is listening. The port number may be omitted if the port is the standard port for the service requested.	Host: en.wikipedia.org

HTTP REQUEST EXAMPLE



A detailed example of an HTTP Request Message is shown:

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35
bookId=12345&author=Tan+Ah+Teck
```

The Request Line is the first line: **GET /doc/test.html HTTP/1.1**. The Request Headers follow, ending with a blank line: **Host: www.test101.com**, **Accept: image/gif, image/jpeg, */***, **Accept-Language: en-us**, **Accept-Encoding: gzip, deflate**, **User-Agent: Mozilla/4.0**. A blank line separates the headers from the body. The Request Message Body is the last part: **Content-Length: 35** and **bookId=12345&author=Tan+Ah+Teck**.

Request Line →
Request Headers {
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35
}
} Request Message Header
→ A blank line separates header & body
} Request Message Body

HTTP RESPONSE HEADERS

- The server returns HTTP headers with its response
- Provides information about the server
- Describes the data that is being returned
- Used by the browser to decide how to display the data
- Many headers exists, only the most important ones are listed here



HTTP RESPONSE HEADERS

Header Name	Description	Example
Access-Control-Allow-Origin	Specifying which web sites can participate in cross-origin resource sharing	Access-Control-Allow-Origin: *
Accept-Ranges	What partial content range types this server supports via byte serving	Accept-Ranges: bytes
Content-Encoding	The type of encoding used on the data. See HTTP compression.	Content-Encoding: gzip
Content-Language	The natural language or languages of the intended audience for the enclosed content[38]	Content-Language: en-US
Content-Length	The length of the response body in octets (8-bit bytes)	Content-Length: 348
Content-Range	Where in a full body message this partial message belongs	Content-Range: bytes 21010-47021/47022
Content-Type	The MIME type of this content	Content-Type: text/html; charset=utf-8
Set-Cookie	An HTTP cookie	Set-Cookie: UserID=JohnDoe; Max-Age=3600; Version=1
Server	A name for the server	Server: Apache/2.4.1 (Unix)

HTTP RESPONSE EXAMPLE



The example shows a detailed view of an HTTP/1.1 200 OK response. It includes the status line, various response headers (Date, Server, Last-Modified, ETag, Accept-Ranges, Content-Length, Connection, Content-Type), a blank line separating the header from the body, and the response message body itself, which contains an HTML heading.

HTTP/1.1 200 OK

Date: Sun, 08 Feb xxxx 01:11:12 GMT

Server: Apache/1.3.29 (Win32)

Last-Modified: Sat, 07 Feb xxxx

ETag: "0-23-4024c3a5"

Accept-Ranges: bytes

Content-Length: 35

Connection: close

Content-Type: text/html

<h1>My Home page</h1>

Status Line

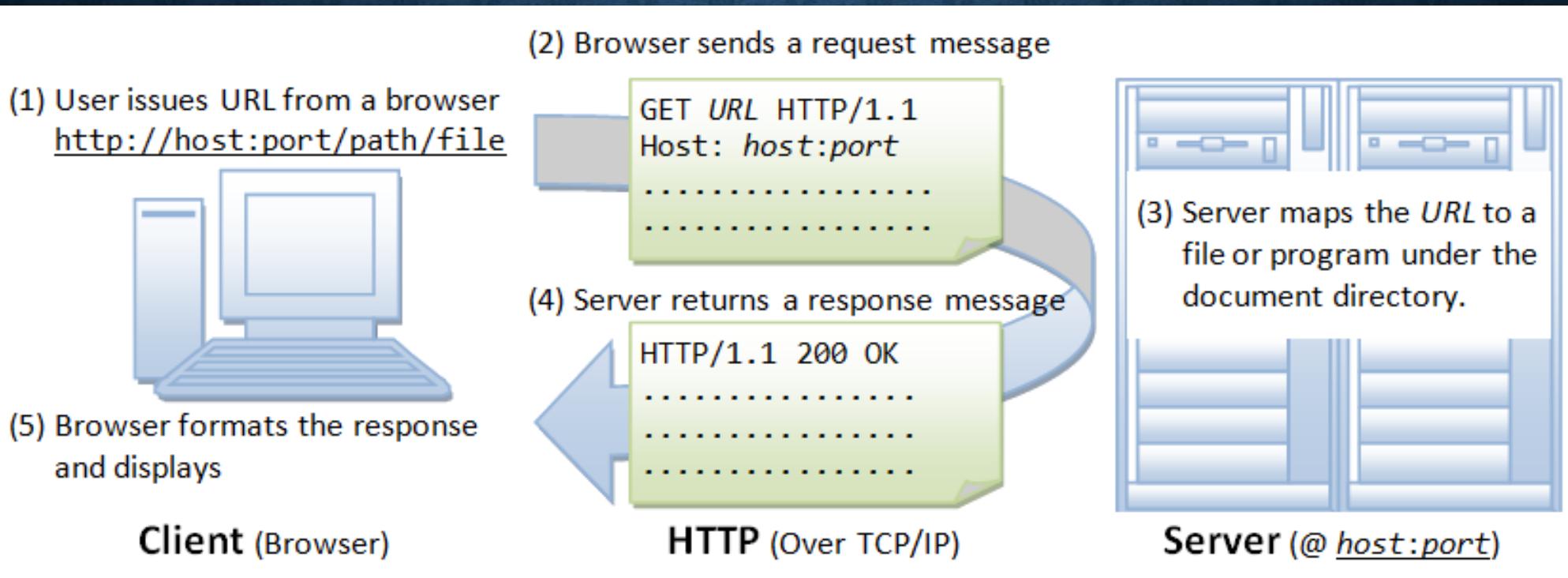
Response Headers

Response Message Header

A blank line separates header & body

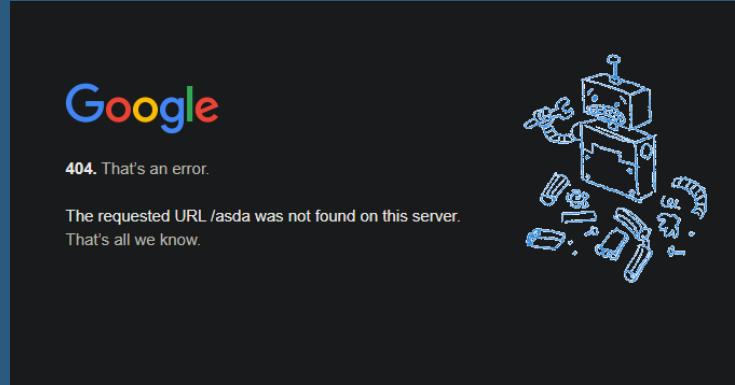
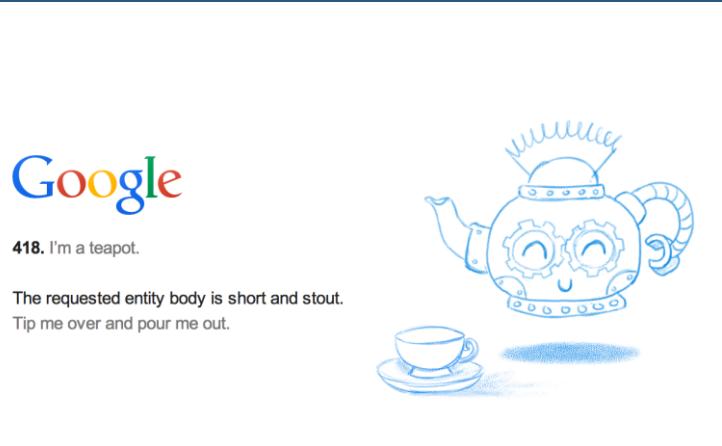
Response Message Body

HTTP REQUESTS



INTERNET MIME TYPES

Type	Description
application/octet-stream	Any kind of binary data
application/x-www-form-urlencoded	Form Data
text/plain	Plain Text
text/html	HTML
application/xml	XML
application/json	JSON
application/javascript	JavaScript
application/pdf	Adobe Portable Document Format (PDF)
application/zip	ZIP archive
font/ttf	TrueType Font
image/jpeg	JPEG Images
video/mpeg	MPEG Video
audio/mpeg or audio/mp3	MP3 Audio
Many more ...	



HTTP STATUS CODES

HTTP STATUS CODES

Code	Description
1XX	Informational Responses
2XX	Success
3XX	Redirection
4XX	Client Errors
5XX	Server errors
XXX	Other Unofficial Codes

HTTP STATUS CODES

Code	Description
200	OK
301	Moved Permanently
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
408	Request Timeout
500	Internal Server Error
502	Bad Gateway
503	Service Unavailable
418	I'm a teapot
Many more ...	

HTTP PARAMETERS

- Customize the request
 - Request specific values from the server
 - Send data to server to update values on server/database
- Separated by an ampersand (&)
- Format: parameterName=parameterValue
- Example: `username=satoshi&password=Complex$1Pass`

HTTP GET PARAMETERS

- Send parameters as part of the request URL
- The URL and its parameters can be bookmarked for quick access
- Not secure
 - Parameters are send as raw text
 - Does not benefit from HTTPS (SSL/TLS) since parameters are not encrypted
 - Do not use for passwords or any sensitive data
- Example:
 - Request URL: `http://moviesite.com/api?type=movie&title=avatar&year=2009`
 - Request Body: `None`

HTTP POST PARAMETERS

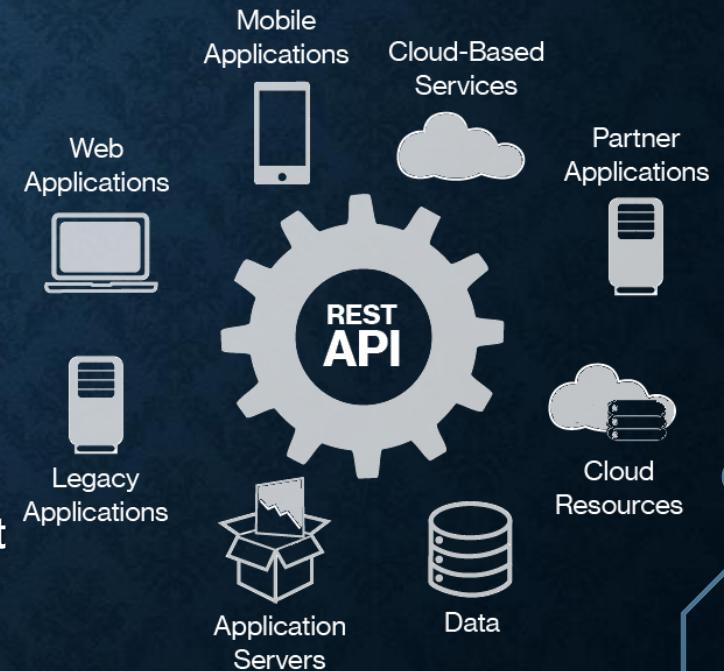
- Send parameters as part of the request body
- The URL can be bookmarked, but without its parameters
- Secure
 - Parameters are send as part of the body, not as part of the URL
 - Does benefit from HTTPS (SSL/TLS) since the entire body is encrypted
 - Use for passwords or any sensitive data
- Example:
 - Request URL: `http://moviesite.com/api`
 - Request Body: `action=login&username=satoshi&password=Complex$1Pass`

URLENCODE

- Characters in a URL have to be encoded:
 - For some non-ASCII characters
 - Characters reserved for URLs (eg: / : @ .)
- For instance, sending an URL as a GET parameter:
 - Wrong: `http://api.mysite.com/submit?url=http://cs.up.ac.za`
 - Correct: `http://api.mysite.com/submit?url=http%3A%2F%2Fcs.up.ac.za`
- Work out the URL encoding yourself or use a tool:
 - <https://www.urlencoder.org>

REST APIs

- Public or private stateless APIs running mostly over HTTP
- Retrieve information from or change data on a server
- Send a HTTP request to the server and retrieve the data
 - Traditionally APIs returned XML data
 - Almost all APIs return JSON data these days
- Examples:
 - Twitter & Facebook: Retrieve your friend's latest post/tweet
 - Trakt.tv: Retrieve movie and TV information
 - CoinMarketCap.com: Retrieve the latest crypto currency prices





<https://www.youtube.com/watch?v=7YcW25PHnAA>

HTTP APIs – COINMARKETCAP.COM

- Request: <https://api.coinmarketcap.com/v1/ticker/>

- Response:

```
[  
  {  
    "id": "bitcoin",  
    "name": "Bitcoin",  
    "symbol": "BTC",  
    "rank": "1",  
    "price_usd": "8558.77",  
    "price_btc": "1.0",  
    "24h_volume_usd": "10752600000.0",  
    "market_cap_usd": "144242234626",  
    "available_supply": "16853150.0",  
    "total_supply": "16853150.0",  
    "max_supply": "21000000.0",  
    "percent_change_1h": "1.1",  
    "last_updated": "1518094763"  
  },  
  ...  
]
```

HTTP APIs – TRAKT.TV

- Request: <https://api.trakt.tv/movies/tron-legacy-2010>

- Response:

```
{  
    "title": "TRON: Legacy",  
    "year": 2010,  
    "ids": {  
        "trakt": 343,  
        "slug": "tron-legacy-2010",  
        "imdb": "tt1104001",  
        "tmdb": 20526  
    },  
    "released": "2010-12-16",  
    "runtime": 125,  
    "updated_at": "2014-07-23T03:21:46.000Z",  
    "trailer": null,  
    "homepage": "http://disney.go.com/tron/",  
    "rating": 8,  
    "language": "en",  
    "genres": ["action"],  
    "certification": "PG-13"  
}
```

HTTP REQUEST TOOLS

- HTTP requests can be send:
 - Directly inside a browser
 - Inside your JS or PHP scripts
 - Using external tools:
 - Online website tools (eg: <https://www.hurl.it>,)
 - Browser addons (eg: Firefox RESTClient, Talend API Tester)
 - Command-line tools (eg: cURL in Linux)
 - GUI tools (eg: Postman, Insomnia, ThunderClient)

HTTP REQUEST TOOLS – TALEND API TESTER

The screenshot shows the TALEND API Tester interface. At the top, there are tabs for 'CLIENT', 'REQUESTS', and 'SCENARIOS'. The 'REQUESTS' tab is selected. The main area has a 'METHOD' dropdown set to 'GET' with a red arrow pointing to it. Below it is a 'SCHEME // HOST [":" PORT] [PATH ["?" QUERY]]' input field containing 'https://api.coinmarketcap.com/v1/ticker/' with a red arrow pointing to it. To the right of the URL is a 'Send' button with a red arrow pointing to it. In the middle left, there's a 'HEADERS' section with a '+ Add header' button and a red arrow pointing to it. Below that is a 'BODY' section with a 'Form' dropdown. On the right, there's a 'Response' section showing a green bar with '200' and a red arrow pointing to it. The response body contains JSON data with a red arrow pointing to it.

Set HTTP method

API URL

Add HTTP query parameters

Add HTTP headers

Returned HTTP status code

Returned HTTP response

Returned HTTP body data

Send request

HTTP REQUEST TOOLS – TALEND API TESTER

The screenshot shows the Talend API Tester application interface. The top navigation bar includes tabs for 'API TESTER' (selected), 'REQUESTS', 'SCENARIOS', and 'REPOSITORY'. A sidebar on the left shows 'MY DRIVE' with 'NO SAVED DATA'. The main workspace displays a 'DRAFT' request configuration.

METHOD: POST
SCHEME ::// HOST [":" PORT] [PATH ["?" QUERY]]: https://www.google.com/recaptcha/api/siteverify?

QUERY PARAMETERS:

- secret = 6LeUc8oUAAAAAGP
- response = 03AOLTBLRUYq19yq

HEADERS (Form):

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

BODY (Form):

- response = 03AOLTBLRUYq19yq
- secret = 6LeUc8oUAAAAAGP

Response:
Not available, a request has not been sent yet.

Bottom navigation bar: HISTORY, ASSERTIONS, HTTP, DESCRIPTION.

HTTP REQUEST TOOLS – THUNDERCLIENT

The screenshot shows the ThunderClient interface within the VSCode extension. At the top, there's a navigation bar with tabs for 'New Request', 'Activity' (which is selected), 'Collections', and 'Env'. Below the navigation bar, a search bar contains 'https://www.thunderclient.com/welcome' with a dropdown menu showing 'GET' and 'Headers 2'. To the right of the search bar is a 'Send' button. Underneath the search bar, there are tabs for 'Query' (selected), 'Headers 2', 'Auth', 'Body', 'Tests', 'Pre Run', and 'New'. The main area is titled 'Query Parameters' and contains a table with one row: a checkbox labeled 'parameter' and a text input labeled 'value'. Below this table, the response status is displayed as 'Status: 200 OK Size: 419 Bytes Time: 2.00 s'. At the bottom, there are tabs for 'Response' (selected), 'Headers 10', 'Cookies', 'Results', and 'Docs'. The 'Response' tab displays the JSON content of the API response.

```
1 {  
2   "message": "Welcome to Thunder Client",  
3   "about": "Lightweight Rest API Client for VSCode",  
4   "createdBy": "Ranga Vadhineni",  
5   "launched": 2021,  
6   "features": {  
7     "git": "Save data to Git Workspace",  
8     "themes": "Supports VSCode Themes",  
9     "data": "Collections & Environment Variables",  
10    "testing": "Scriptless Testing",  
11    "local": "Local Storage & Works Offline"  
12  },  
13  "supports": {  
14  }  
15}
```

ONLINE RESOURCES – W3SCHOOL.COM

- Everything web-related (HTML, JS, CSS, PHP, MYSQL, much more ...)

The screenshot shows the homepage of w3schools.com. The header features the site's logo "w3schools.com" and the tagline "THE WORLD'S LARGEST WEB DEVELOPER SITE". Below the header, there are three main navigation categories: "TUTORIALS", "REFERENCES", and "EXAMPLES". The "TUTORIALS" section is currently selected and expanded, displaying a list of topics: HTML and CSS, JavaScript, and Server Side. Each topic has a corresponding "Learn" link. The main content area is titled "HTML" and describes it as "The language for building web pages". It includes two buttons: "LEARN HTML" and "HTML REFERENCE". To the right, there is a "HTML Example" code block with syntax highlighting for HTML tags like <!DOCTYPE html>, <html>, <title>, <body>, <h1>, <p>, and </>. A "Try it Yourself" button is located at the bottom of this example. At the bottom of the page, there are sections for "CSS Example" and "CSS Examples".

CSS
C22

ONLINE RESOURCES – PHP.NET

More documentation
on PHP

The screenshot shows the "Documentation" section of the official PHP website. The top navigation bar includes links for "php", "Downloads", "Documentation" (which is highlighted), "Get Involved", and "Help". A search bar is also present. The main content area has a heading "Documentation" and a sub-section titled "Documentation for PHP 4". It notes that while PHP 4 documentation is removed from the main manual, an archived version is available. Below this, there's a table comparing "Formats" (View Online or Downloads) with "Destinations" (languages like English, Brazilian Portuguese, Chinese, French, German, Japanese, etc.). A "More documentation" section at the bottom lists links for documentation HOWTO, PHP-GTK documentation, and PECL documentation.

Documentation

The PHP Manual is available online in a selection of languages. Please pick a language from the list below.

More information about php.net URL shortcuts by visiting our [URL howto page](#).

Note, that many languages are just under translation, and the untranslated parts are still in English. Also some translated parts might be outdated. The translation teams are open to contributions.

Documentation for PHP 4 has been removed from the manual, but there is archived version still available. For more informations, please read [Documentation for PHP 4](#).

Formats	Destinations
View Online	English , Brazilian Portuguese , Chinese (Simplified) , French , German , Japanese , Romanian , Russian , Spanish , Turkish
Downloads	For downloadable formats, please visit our documentation downloads page.

More documentation

- If you are interested in how the documentation is edited and translated, you should read the [Documentation HOWTO](#).
- [PHP-GTK related documentation](#) is hosted on the PHP-GTK website.
- Documentation of PECL and the various packades can be found on a separate server.

ONLINE RESOURCES – STACKOVERFLOW.COM

- Questions,
code snippets, and more

The screenshot shows a question titled "Capture HTML Canvas as gif/jpg/png/pdf?" on the Stack Overflow homepage. The question has 582 votes and is tagged with javascript, html5, canvas, export, and png. It was asked 8 years, 8 months ago and viewed 431,623 times. The question asks if it's possible to capture or print what's displayed in an html canvas as an image or pdf? A user named ZJR provided an answer with a link to a pythonic solution. Another user named sam provided an answer with a link to examples of HTMLCanvasElement.toDataURL(). The answers section shows 10 answers, with the most recent one being active.

Get personalized job matches now

stackoverflow Questions Developer Jobs Tags Users Search... Log In Sign Up Ask Question

Capture HTML Canvas as gif/jpg/png/pdf?

Is it possible to capture or print what's displayed in an html canvas as an image or pdf?

582 I'd like to generate an image via canvas, and be able to generate a png from that image.

javascript html5 canvas export png

share improve this question edited Mar 29 '16 at 13:54 asked May 29 '09 at 0:28 ZJR 6,477 ● 2 ● 22 ● 32 Parand 34.5k ● 35 ● 131 ● 178

Here's a pythonic solution: stackoverflow.com/questions/19395649/... in addition to answer stackoverflow.com/a/3514404/529442 – Evgeny Philippov May 23 '16 at 16:20

HTMLCanvasElement.toDataURL#Examples – sam Oct 8 '16 at 22:10

add a comment

10 Answers active oldest votes

Oops. Original answer was specific to a similar question. This has been revised:

600 var canvas = document.getElementById("mycanvas"); var img = canvas.toDataURL("image/png");

with the value in IMG you can write it out as a new Image like so:

900

BENSON NEEDS AN EDIT. nodejsup limits a file size to 10MB in a single file

Pluralsight & Stack Overflow: Helping the World's Developers Learn New Skills

Gotthardt Healthgroup AG

We have great benefits!

Flexible working hours

Work in an excellent team

Nice atmosphere, no dresscode

RNV-Jobticket for public transport & office in the heart of Heidelberg

+ 6 more benefits

Learn more

QUESTIONS?

