HTTP Deep Dive

Ivan Pepelnjak (ip@ipSpace.net)

ipSpace.net AG

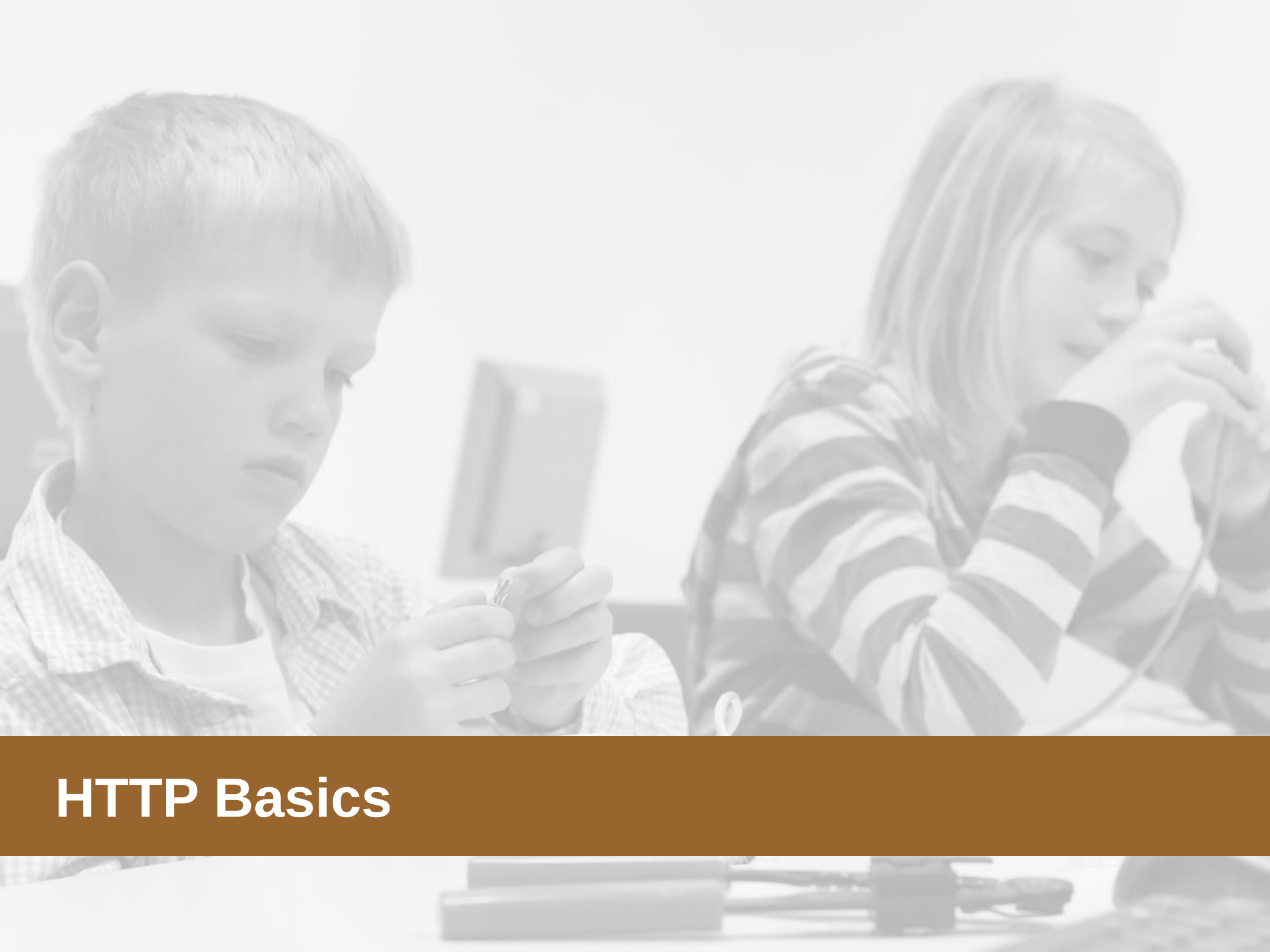
Focus Area: Transport

HTTP principles:

- Request methods
- Header
- Status codes
- Pipelining
- Chunking

Use cases:

- Redirects
- Cookies
- Non-HTML content
- Streaming
- Compression
- Server-side push



HTTP Basics

HTTP Mission Statement

The Hypertext Transfer Protocol (HTTP) is an **application-level protocol** for distributed, collaborative, hypermedia information systems. It is a **generic, stateless**, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its **request methods, error codes and headers**. A feature of HTTP is the **typing and negotiation of data representation**, allowing systems to be built independently of the data being transferred.

Sample HTTP Transaction

```
GET / HTTP/1.1
Host: www.google.si
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:9.0.1) Gecko/20100101
Firefox/9.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive

HTTP/1.1 200 OK
Date: Sun, 26 Feb 2012 16:36:06 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 24237
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
```

HTTP Request/Response Format

- First line: method + URI (request) or status (response)
- Subsequent lines: header fields + values
- Empty line: header/content separator
- Binary content

```
GET / HTTP/1.1
Host: www.google.si
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK
Date: Sun, 26 Feb 2012 16:36:06 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Length: 24237
```

HTTP Request Methods

Typical:

- GET
- POST
- CONNECT

Less common:

- HEAD

RESTful API

- PUT
- DELETE

Troubleshooting:

- OPTIONS
- TRACE

GET and HEAD must be idempotent

Request URI

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

- Hostname/protocol not included in request URI
  Exception: proxy servers

Mandatory headers:

- **Host** - used to implement virtual servers with shared IP addresses

HTTP Status Codes – Overview

```
HTTP/1.1 200 OK
Date: Sun, 26 Feb 2012 16:36:06 GMT
```

1xx Informational

2xx Success

3xx Redirection

4xx Client Error

5xx Server Error

Information Status Codes (Rare)

100 Continue

- Check request headers before sending request body
- Include Expect: 100-continue header in request

102 Processing (WebDAV) (RFC 2518)

- Prevents the client from timing out and assuming the request was lost.

103 Checkpoint

- Used in the Resumable HTTP Requests Proposal

Success Status Codes

200 OK

201 Created

- Used in RESTful API
- Returns URI of newly-created resource in Location header

202 Accepted

203 Non-Authoritative Information (since HTTP/1.1)

204 No Content

- Change metainformation without changing document view
- Also used for statistics/user tracking

205 Reset Content

- Reset form fields

206 Partial Content

- Used with range headers

Redirection Status Codes

300 Multiple Choices (do not use)

301 Moved Permanently

- Resource @ new URI (provided in Location header)
- Always use the new URI

302 Found

- Soft redirect (do not replace old URI with new)

303 See Other

- POST-to-GET redirection

304 Not Modified

- Used in HTTP caching

307 Temporary Redirect

- Like 302, but prohibiting POST-to-GET conversion

Use 3xx codes for web site migration, consistent URLs and SEO

Client-Side Error Codes (Common)

400 Bad Request

401 Unauthorized

- Repeat with HTTP authentication

403 Forbidden

- The resource might be there, but you'll never get it

404 Not Found

- Resource is not found

410 Gone

- Used to purge caches

You should include HTTP content with error code responses

Client-Side Error Codes (Continued)

405 Method Not Allowed (using GET on a POST-only script)

406 Not Acceptable (no match with Accept headers)

407 Proxy Authentication Required

409 Conflict

413 Request Entity Too Large

- Example: Upload of a huge file

414 Request-URI Too Long

415 Unsupported Media Type

416 Requested Range Not Satisfiable

- Used with range requests

417 Expectation Failed

418 I'm a teapot (RFC 2324)

420 Enhance Your Calm – used by Twitter

You can use any code you wish … if you control the client (AJAX)

Server-Side Error Codes

500 Internal Server Error

- Automatically generated on script compilation/execution error

501 Not Implemented

502 Bad Gateway

503 Service Unavailable

- Use when the server is overloaded (example: DB open fails)
- Include Retry-After field in response header

504 Gateway Timeout

505 HTTP Version Not Supported

Catch overload errors and response with 503

What Is In The Response?

What Is In the Response?

```
HTTP/1.1 200 OK
Date: Sun, 26 Feb 2012 16:36:06 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
```

Specifying the content type:

- **Content-Type: media-type** header
- **Media-type** = type/subtype; parameter
- Type = application, audio, image, text, video ...

Sample subtypes:

- Image: gif, png, jpeg ...
- Video: h264, mpeg, mp4, ogg, vnd.*
- Text: plain, html, xml, css,
- Application: javascript, atom+xml, xhtml+xml, pdf, vnd.ms-excel

MIME Type Registry @ http://www.iana.org/assignments/media-types

How Is the Response Encoded?

Character encoding

- Specified in *charset* parameter of *Content-type* header
- Use UTF-8 if at all possible
- Prefer HTTP headers over META tags

```
<meta http-equiv="Content-Type"
 content="text/html; charset=UTF-8">
```

- Use both if you have unusual character set and expect users to save HTML files to disk

```
HTTP/1.1 200 OK
Date: Sun, 26 Feb 2012 16:36:06 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
```

Response language

- Specified in *Content-Language* header and/or equivalent META tag
- Used by some search engines (not by Google Translate)

Content-Specific HTTP Headers

Client-specified conditions

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language

Content description

- Content-Language
- Content-Length
- Content-Type
- Content-Encoding
- Content-MD5

Content Description: Type And Length

```
GET /images/icons/product/chrome-48.png HTTP/1.1
Host: www.google.si
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:9.0.1) Gecko/20100101 Firefox/9.0.1
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer: http://www.google.si/

HTTP/1.1 200 OK
Content-Type: image/png
Last-Modified: Tue, 22 Mar 2011 21:49:32 GMT
Date: Sun, 26 Feb 2012 17:49:54 GMT
Expires: Sun, 26 Feb 2012 17:49:54 GMT
Cache-Control: private, max-age=31536000
X-Content-Type-Options: nosniff
Server: sffe
Content-Length: 1834
X-XSS-Protection: 1; mode=block
```

Content Description: Character Set

```
GET /bin/start HTTP/1.1
Host: cmsdev.ipspace.net
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive

HTTP/1.1 200 OK
Date: Sun, 26 Feb 2012 17:28:29 GMT
Server: Apache/2.2.15 (CentOS)
Expires: Sun, 26 Feb 2012 17:29:00 GMT
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
```

Always specify character set for *text* type in Content-Type

Who Generates Content Headers

Web server when serving static files

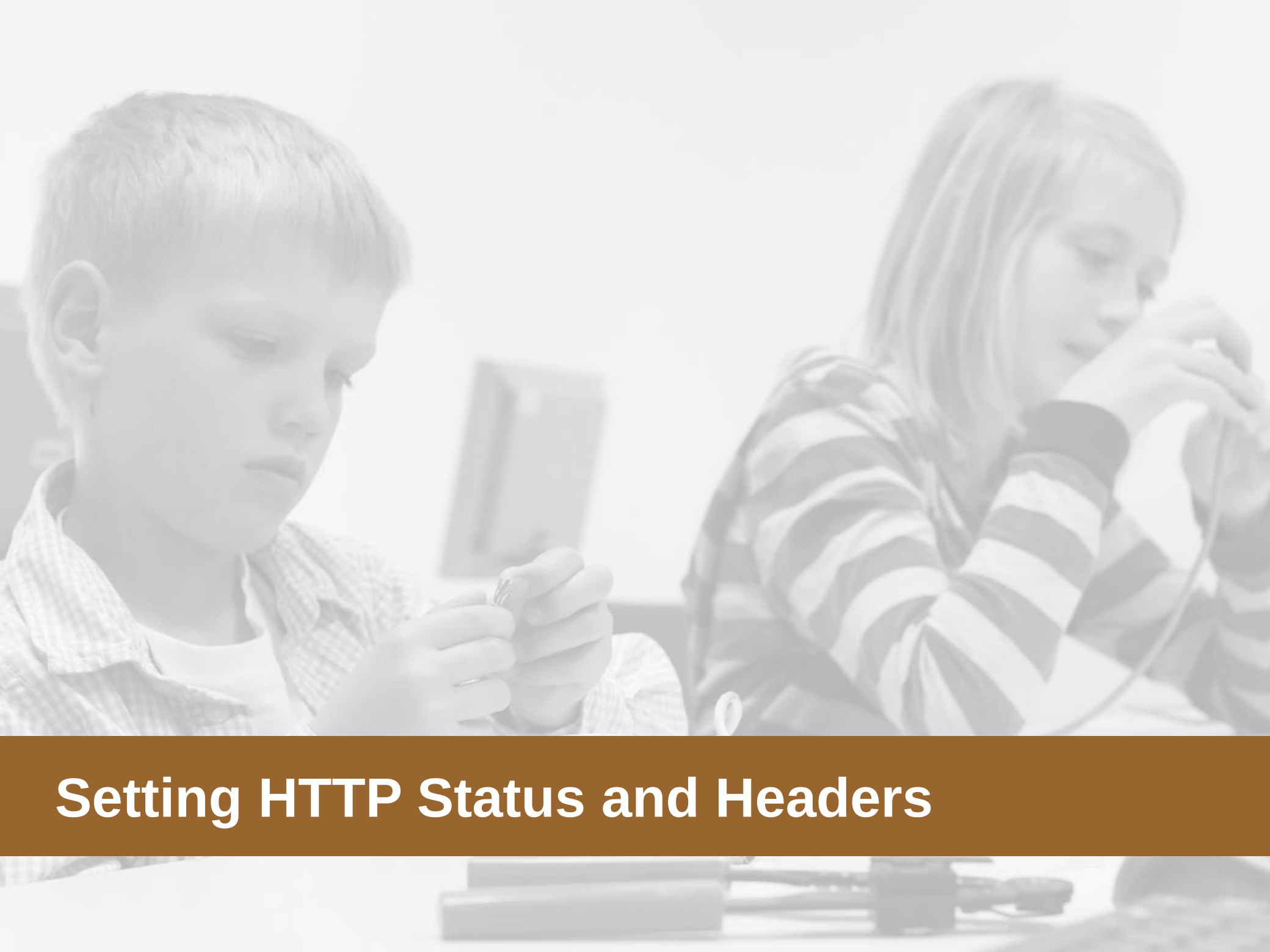
- *Content-type* based on filename extension
- *Content-length* equals file length
- Character encoding and *Content-Language* based on web server configuration

Example: Apache 2.2

- *Content-Language* and charset could be based on filename extension
- Filename might have multiple extensions (.html.sl.csz)
- Path info (part after the actual file name) could also be used (x.html.en would be an English document even when only x.html exists)

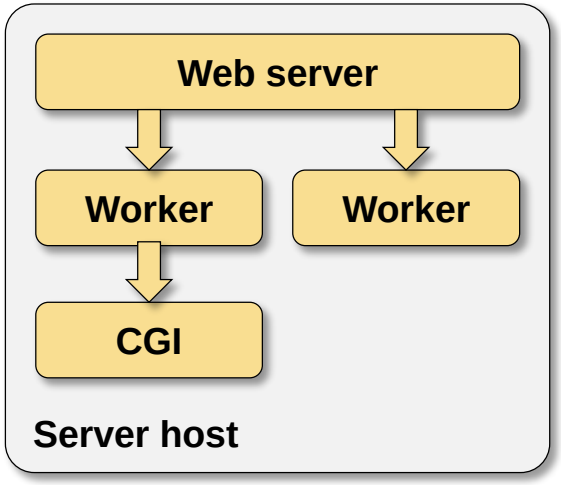
Server-side script when serving dynamic content (HOW???)

See http://httpd.apache.org/docs/2.2/mod/mod_mime.html for details



Setting HTTP Status and Headers

CGI Interface Overview (RFC 3875)



- Web server is communicating with the browser ➔ receiving HTTP request and sending response
- Web server executes requested script in child process

Server-to-script communication

- Meta-variables passed as environment variables
- Request headers passed as HTTP_* or HTTPS_* environment variables
- Request content (ex: form data) passed through STDIN

Script-to-server communication:

- Response sent to STDOUT
- List of headers (one per line) + empty line + response body –or– single-line Location header (redirect)
- Response MUST include Content-Type and Status headers
- Status header used by web server to generate first line of HTTP response

Useful CGI variables

- Client information: REMOTE_HOST, REMOTE_USER, AUTH_TYPE, REMOTE_ADDR
  - ➔ REMOTE_USER and AUTH_TYPE only when the web server authenticates the user
  - ➔ REMOTE_HOST only when reverse DNS lookup works
- Server information: SERVER_NAME, SERVER_PORT, SERVER_PROTOCOL
  - ➔ used to build full URL in responses or emails
- Request/script information
  - ➔ CONTENT_LENGTH = length of request body
  - ➔ CONTENT_TYPE = content-type HTTP header
  - ➔ PATH_INFO = extra path from URI
  - ➔ PATH_TRANSLATED = translated PATH_INFO (mostly useless)
  - ➔ QUERY_STRING = part of URL after the question mark
  - ➔ REQUEST_METHOD = GET, POST, PUT … (you get the idea)
  - ➔ SCRIPT_NAME = script part of the URI

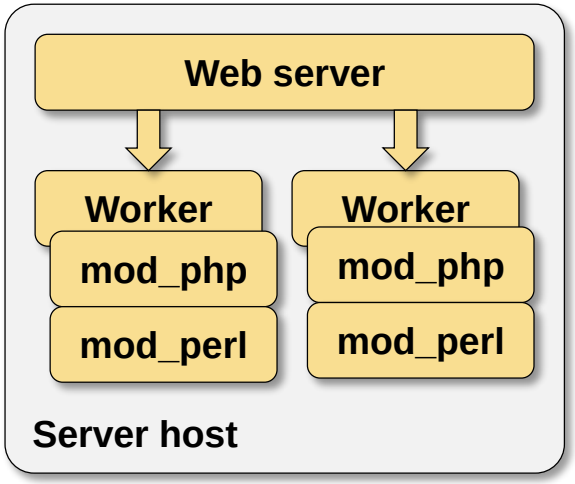
CGI Example

GET http://cgi.bw.org/cgi-t/test.cgi/wazzup?name=value#top

```
HTTP_HOST [cgi.bw.org]
HTTP_USER_AGENT [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_5)]
PATH_INFO [/wazzup]
PATH_TRANSLATED [/var/web/vhosts/cgi.bw.org/wazzup]
QUERY_STRING [name=value]
REMOTE_ADDR [193.110.145.6]
REMOTE_PORT [60566]
REQUEST_METHOD [GET]
REQUEST_URI [/cgi-t/test.cgi/wazzup?name=value]
SCRIPT_FILENAME [/var/web/vhosts/cgi.bw.org/cgi-t/test.cgi]
SCRIPT_NAME [/cgi-t/test.cgi]
SERVER_ADDR [184.106.128.201]
SERVER_NAME [cgi.bw.org]
SERVER_PORT [80]
SERVER_PROTOCOL [HTTP/1.1]
```

Scripting Interface Overview

- Scripting language interpreter loaded into web worker process (loadable library)
- File extension or configuration rules select desired language interpreter
- Most scripting languages provide semi-parsed data to the script:
  - ➔ Request parameters in $_REQUEST or similar
  - ➔ Cookies in $_COOKIE
  - ➔ CGI-like variables in $_SERVER
  - ➔ Environment variables in $_ENV
- Response headers generated with function calls or through Response object
- Response data is generated by the script's PRINT statements



Response headers might be dumped the moment you print the first data

Setting HTTP Headers and Status

CGI script

- Write response headers and status information to STDOUT
- Use libraries whenever available (Perl: CGI)

PHP

```
header("HTTP/1.1 302 Gone with the wind");
header("Location: http://www.example.com/");
```

ASP

```
Response.Status "302 Try something else"
Response.AddHeader "Wazzup”,"Don’t know"
```

Reading language documentation usually helps

Content Compression

Content Encoding: Compression

```
GET / HTTP/1.1
Host: www.google.si
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:8.0.1)
Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive

HTTP/1.1 200 OK
Date: Mon, 27 Feb 2012 09:49:35 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 17349
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
```

Standard methods: gzip and deflate. Very old browsers are buggy

HTTP Compression: Hints

- Use if at all possible
- Configure it for compressible content (exclude JPG/PNG/MPx)
- Configured in web server configuration
  Apache: mod_deflate, mod_gzip (SourceForge)

```
AddOutputFilterByType DEFLATE text/html text/plain text/xml …
```

Optional: Configure compression for server-side scripts

- Configure in individual scripting environment
- Might happen automatically with recent versions of Apache

More @ http://phplens.com/phpeverywhere/tuning-apache-php

Pipelining and Chunking

Persistent Connections and Pipelining

Persistent connections:

- Reduce the number of TCP sessions
- Multiple HTTP requests executed over the same TCP session
- Controlled with *Connection* header (keep-alive or close)
- Default: Connections are persistent

```
KeepAlive On
MaxKeepAliveRequests 100
```

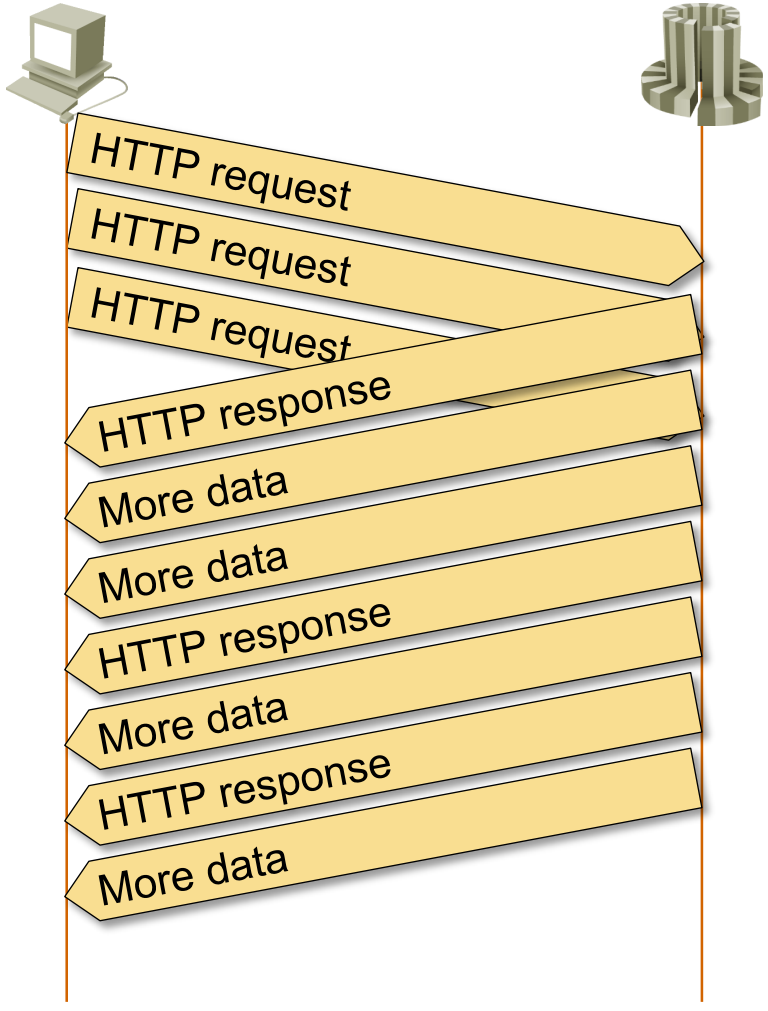
Loved by browsers, hated by people running Apache servers. Why?

Pipelining

- Multiple requests sent “in advance”
- Works only for responses with known content-length
- Usually triggers chunked encoding

Typical script processing (Apache 2.2.17)

- Worker reads up to chunk-size data from script output
- Has the script finished?
  - Yes: generate **Content-length** header and send response
  - No: send chunk, wait for more



HTTP pipelining

Chunked Transfer Encoding

```
GET /Main_Page HTTP/1.1
Host: www.ipspace.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:8.0.1) Gecko/20100101
Firefox/8.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive

HTTP/1.1 200 OK
Date: Mon, 27 Feb 2012 09:50:38 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.2
Content-Language: en
Vary: Accept-Encoding,Cookie
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
```

Chunked Transfer Encoding

```
▷ HTTP/1.1 200 OK\r\n
  Date: Mon, 27 Feb 2012 10:13:14 GMT\r\n
  Server: Apache/2.2.15 (CentOS)\r\n
  X-Powered-By: PHP/5.3.2\r\n
  Content-language: en\r\n
  Vary: Accept-Encoding,Cookie\r\n
  X-Vary-Options: Accept-Encoding;list-contains=gzip,Cookie;string-contains=ioswikiToken;string-contains
  Expires: Thu, 01 Jan 1970 00:00:00 GMT\r\n
  Cache-Control: no-cache, no-store, max-age=0, must-revalidate\r\n
  Pragma: no-cache\r\n
  Keep-Alive: timeout=15, max=100\r\n
  Connection: Keep-Alive\r\n
  Transfer-Encoding: chunked\r\n
  Content-Type: text/html; charset=utf-8\r\n
  \r\n
▽ HTTP chunked response
  ▽ Data chunk (15340 octets)
      Chunk size: 15340 octets
    ▷ Data (15340 bytes)
      Chunk boundary
  ▽ End of chunked encoding
      Chunk size: 0 octets
      Chunk boundary
```

Why Do I Care?

- Content-type: generate non-HTML content with scripts
  - Custom authentication
  - Thumbnails
  - Downloads
  - Dynamic PDFs
- Status codes: proper handling of redirects and errors
- Accept-language: auto-detect visitor's language
- Content-language: used by some search engines (najdi.si)
- Content-Encoding, pipelining: optimize performance

Performance Tuning

- Enable compression (server configuration + scripting language)
- Smaller chunks (if needed and configurable)
- Persistent sessions (check with Wireshark)
- Domain sharding

Questions?