

# Cookies and Sessions

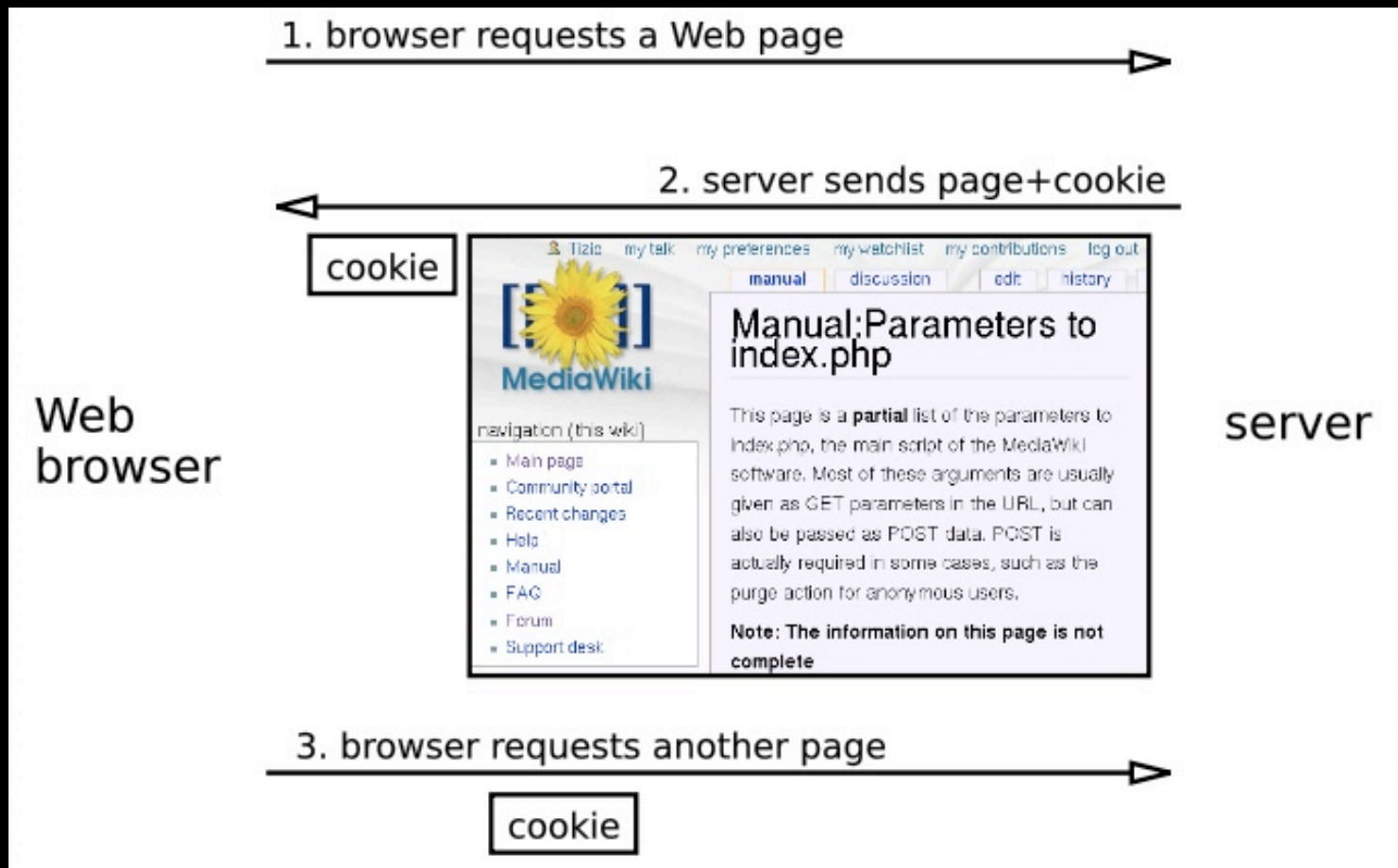
# Multi-User / Multi-Browser

- When a server is interacting with many different browsers at the same time, the server needs to know \*which\* browser a particular request came from.
- Request / Response initially was stateless - all browsers looked identical . This was really bad and did not last very long at all.

# Web Cookies to the Rescue

*Technically, cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site.*

[http://en.wikipedia.org/wiki/HTTP\\_cookie](http://en.wikipedia.org/wiki/HTTP_cookie)



[http://en.wikipedia.org/wiki/HTTP\\_cookie](http://en.wikipedia.org/wiki/HTTP_cookie)

# Cookies In the Browser

- Cookies are marked as to the web addresses they come from. The browser only sends back cookies that were originally set by the same web server.
- Cookies have an expiration date. Some last for years, others are short-term and go away as soon as the browser is closed

## Cookies

---

PHP transparently supports HTTP cookies. Cookies are a mechanism for storing data in the remote browser and thus tracking or identifying return users. You can set cookies using the [setcookie\(\)](#) or [setrawcookie\(\)](#) function. Cookies are part of the HTTP header, so [setcookie\(\)](#) must be called before any output is sent to the browser. This is the same limitation that [header\(\)](#) has. You can use the [output buffering functions](#) to delay the script output until you have decided whether or not to set any cookies or send any headers.

Any cookies sent to you from the client will automatically be included into a `$_COOKIE` auto-global array

// cookie will expire when the browser close

To delete a cookie, use the `setcookie()` function with an expiration date in the past

<http://php.net/manual/en/features.cookies.php>

## Syntax

```
setcookie(name, value, expire);
```

## Parameter Values

Parameter	Description
<i>name</i>	Required. Specifies the name of the cookie
<i>value</i>	Optional. Specifies the value of the cookie
<i>expire</i>	Optional. Specifies when the cookie expires. The value: <code>time()+86400*30</code> , will set the cookie to expire in 30 days. If this parameter is omitted or set to 0, the cookie will expire at the end of the session (when the browser closes). Default is 0

- To modify a cookie, just set (again) the cookie using the `setcookie()` function
- To delete a cookie, use the `setcookie()` function with an expiration date in the past

```
<?php
// Note - cannot have any output before setcookie, even a free space/line
if ( ! isset($_COOKIE['zap']) ) {
    setcookie('zap', '42', time()+3600);
}
print_r($_COOKIE); ?>

<p><a href="cookie.php">Click Me!</a> or press Refresh</p>
```

<http://www.wa4e.com/code/sessions/cookie.php>



Time



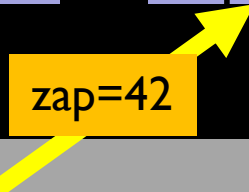
Browser



Browser  
zap=42 🍪



zap=42



Browser  
zap=42 🍪



Browser  
zap=42 🍪



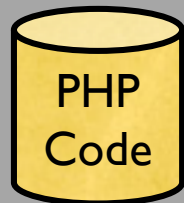
Apache

setcookie()

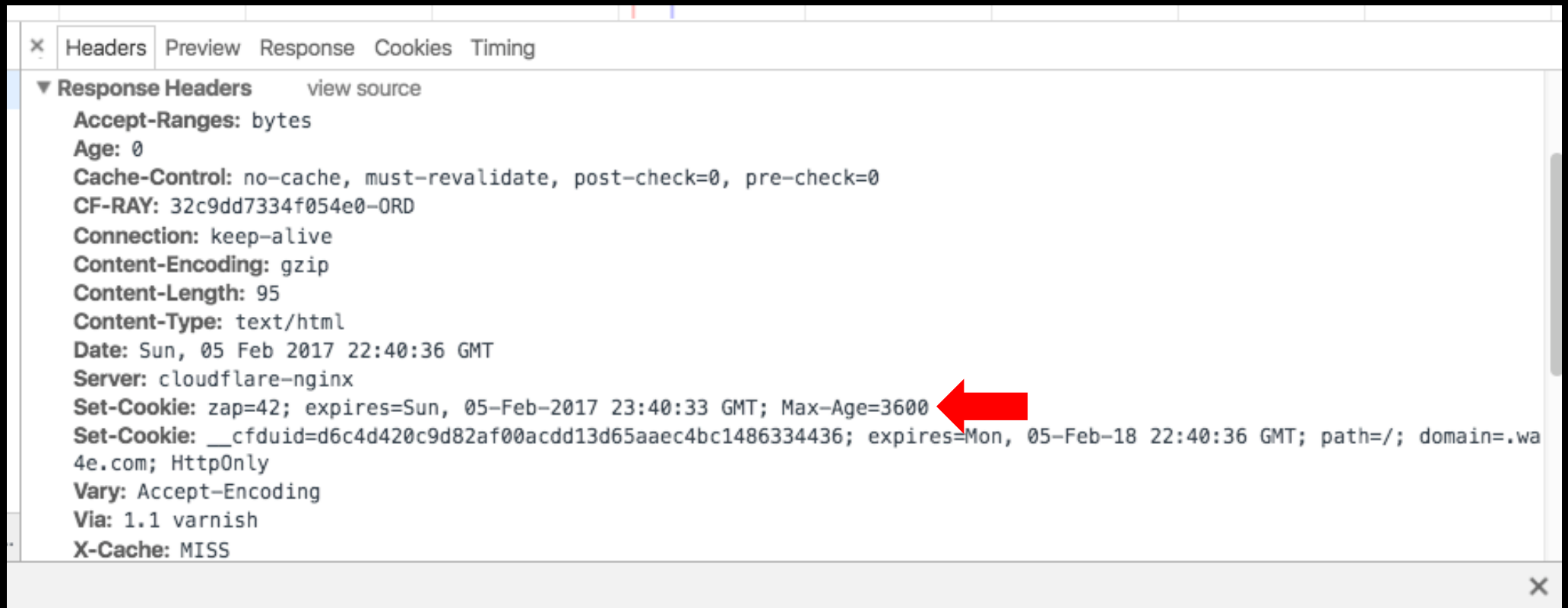
\$\_COOKIE

\$\_COOKIE

\$\_COOKIE



Test request to : <http://www.wa4e.com/code/sessions/cookie.php>



<< Press Refresh >>

×

Headers

Preview

Response

Cookies

Timing

Via: 1.1 varnish

X-Cache: MISS

X-Cache-Svr: d12

X-Powered-By: PHP/5.5.9-1ubuntu4.20

X-Varnish: 2131034848

▼ Request Headers

view source

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

Accept-Encoding: gzip, deflate, sdch

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

Connection: keep-alive

Cookie: zap=42; \_\_cfduid=d6c4d420c9d82af00acdd13d65aaec4bc1486334436

Host: www.wa4e.com

Referer: http://www.wa4e.com/code/sessions/cookie.php

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36

×

# Cookies Limitation

- Users data are not completely controlled by server (can be deleted or modified by client side processes).
- Require sending user data with each request (confidentiality, performance issue).
- Data format restricted to “text” type (can not store native objects or arrays).
- Developer must take care of putting all cookies writing before any HTML code.