**Networking WiTH php**

**Accessing other websites**

You can access other websites the same way you would access a text file:

Fopen(‘https://www.site.com/’, ‘r’);

You must use a trailing slash after a directory:

[www.example.com/dir/](http://www.example.com/dir/) // will work

[www.example.com/dir](http://www.example.com/dir) // will not work

Example

If (isset($\_GET[‘symbol’]) && !empty($\_GET[‘symbol’])) {

$url = sprintf(‘http://quote.yahoo.com/d/quotes.csv?s=%s&f=nl1’, $\_GET[‘symbol’]);

$fp = fopen($url, ‘r’);

$read = fgetcsv($fp);

Fclose($fp);

// check results for improper symbols

If (strcasecmp($read[0], $\_GET[‘symbol’]) != 0) {

// print the results

Echo ‘<div>The latest value for ’ . $read[0] . ‘ is ‘ . $read[1] . ‘.</div>;

} else {

Echo ‘invalid symbol’;

}

}

// show the form

<form action=”get\_quote.php” method=”get”>

<p>Enter a NYSE stock symbol to get the latest price</p>

<input type=”text” name=”symbol” />

</form>

**Working with sockets**

The fopen() function is one way to access web pages, but a more sophisticated method for interacting with another server is to use sockets.

A socket is a channel through which 2 computers can communicate with each other:

$fp = fsockopen($url, $port, $error\_num, $error\_string, $timeout);

// only the first argument is required

Once the file has been successfully opened, you can again use fwrite(), fgets() etc…

\* parse\_url($url) – returns array of url broken down in to its parts, e.g. host, port number, path, query… \*

Example

Function check\_url($url) {

$url\_pieces = parse\_url($url);

$path = (isset($url\_pieces[‘path’])) ?

$url\_pieces[‘path’] :

‘/’;

$port = (isset($url\_pieces[‘port’])) ?

$url\_pieces[‘port’] :

80;

// connect using fsockopen:

If ($fp = fsockopen($url\_pieces[‘host’], $port, $errno, $errstr, 30) {

// send some data

$send = “HEAD $path HTTP.1.1\r\n”;

$send .= “HOST: {$url\_pieces[‘host’]}\r\n”;

$send .= “CONNECTION: Close\r\n\r\n”;

Fwrite($fp, $send);

// read the response

$data = fgets($fp, 128);

// close the connection

Fclose($fp);

// return the response code

List($response, $code) = explode(‘ ‘, $data);

If ($code == 200) {

Return array($code, ‘good’);

} else {

Return array($code, ‘bad’);

}

} else {

Return array($errstr, ‘bad’);

}

}

$urls = [// array of urls];

Foreach($urls as $url) {

$code = check\_url($url);

Echo “<a href=\”$url\”>$url</a><span>$code</span>”;

}

**Performing IP Geolocation**

Every PC must have an IP address to have access to the internet. An ISP assigns a computer an IP address from a pool of valid addresses only they have access to. By knowing a computers IP address, which PHP stores in: $\_SERVER[‘REMOTE\_ADDR’], you can determine the ISP and therefore, the country.

To perform IP geolocation, you must have access to a GEOIP database.

// see example on P339 for example

**Using cURL**

cURL (client URLs) is a command line tool for working with URLs. With it, you can acess websites, FTP files and more.

**Creating web services**

A web service is a generic term for a server resource that provides a function that, unlike a normal web page, is meant to be accessed directly by another computer, not a user.

2 types:

1. Simple (API-based services)
2. Complex (‘*true web services*’)

Complex – web services often transmit data using custom, non-scalar types. This might require a protocol such as Simple Object Access Protocol (SOAP). This means that instead of just transmitting, say, plain text or xml between client and server, the server may send back data in an agreed-upon object format.

Simple – stateless: basic request-response dynamic. The client makes standard requests in the hope that the server understands that request and is able to reply. A popular type of simple service is called REST-ful

A service does not normally create HTML; it outputs data. In both cases, PHP is still just printing the output, but when using a script as a service, the PHP script has to take the extra step of indicating its alternative usage.

Normally a server associates the content-type of a PHP with HTML, to change that, send a content-type header, indicating the type to be exported:

Header(‘content-type: text/plain’);

Header(‘content-type: text/csv’);

Header(‘content-type: text/xml’);

Header(‘Content-Type: application/json’);

Example

If (isset($\_POST[‘format’])) {

Switch ($\_POST[‘format’]) {

Case ‘csv’:

$type = ‘text/csv’;

Break;

Case ‘json’:

$type = ‘application/json’;

Break;

Default:

$type = ‘text/plain’

}

// create response

$data = [];

$data[‘timestamp’] = time();

// add back in the received data

Foreach ($\_POST as $k => $v) {

$data[$k] = $v;

}

// format the data accordingly

If ($type == ‘application/json’) {

$output = json\_encode($data);

} else if ($type == ‘text/csv’) {

// convert to string

$output = ‘’;

Foreach ($data as $v) {

$output .= ‘”’ . $v . ‘”,’;

}

} else if ($type == ‘text/plain’) {

$output = print\_r($data, 1);

}

} else {

$type = ‘text/plain’;

$output = ‘The service has been incorrectly used’;

}

// set the content type header

Header(“content-type: $type”);

Echo $output;

**Cron**

A cron is a service on UNIX servers that allows tasks to be scheduled and executed automatically:

# # # # #

Minutes hours days months dayOfTheWeek(0-6)

e.g.

30 22 \* \* \* wget -q <http://www.example.com>

// at 10:30pm, every day of the month, every month of the year

* Set ranges with hyphens, e.g. 0-3
* Set multiple values for each field, separating with a comma, e.g. 1, 3, 5