Web Developers' HTTP Toolbox

Lorna Mitchell, Nexmo

https://github.com/lornajane/fun-with-http

Why HTTP? Why Toolbox?

We build the web, it's made of HTTP.

Let's look at the tools we need.

Getting to Know HTTP

Let's start with some theory

Getting to Know HTTP

HTTP HyperText Transport Protocol

Request

```
Host          example.com
Path          /index.php
Method        POST
Headers       Accept: text/html, User-Agent: curl, Content-Type: app
Body          fruit=apple&noun=sparkle
```

Response

```
Status        200
Headers       Content-Type: text/html
Body          <h1>Hello!</h1>
```

HTTP Verbs

A verb is a "doing word" ... in HTTP as well as in English.

- GET
- POST
- DELETE
- PUT
- PATCH
- HEAD

HTTP Headers

Request headers: `Host`, `Accept`

Response headers: `ETag`, `Location`

Entity headers: `Content-Type`

HTTP Status Codes

Code	Meaning
200	OK
201	Created
204	No Content
302	Found
307	Moved Temporarily

HTTP Status Codes

Code	Meaning
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Server Error

HTTP and PHP

PHP can be client, server, or both

Speaking HTTP

Speaking HTTP

To speak HTTP, you will need a client. Suggestions include:

- curl https://curl.haxx.se/
- Postman https://www.getpostman.com/
- HTTPie https://httpie.org/
- http-console https://github.com/cloudhead/http-console
- Paw (Mac only) https://paw.cloud/client
- Insomnia https://insomnia.rest/
- probably your IDE?

Exercise 1: Get the Sample App

Data Formats

Form encoded:

```
Content-Length: 40
Content-Type: application/x-www-form-urlencoded

message=This is a message&name=lornajane
```

JSON data:

```
Content-Type:application/json
Content-Length: 52

{"message":"This is a message", "name":"lornajane"}
```

Working with JSON

Use the tools!

In PHP: `json_encode()` and `json_decode()`. No string matching and definitely no regex.

At the commandline:

- `jq` https://stedolan.github.io/jq/
- `fx` https://github.com/antonmedv/fx

Exercise 2: Set up and use your HTTP client

Making Requests with PHP

Making Requests with PHP

PHP can also be an HTTP client.

To make HTTP requests:

- use PHP streams
- use Guzzle http://docs.guzzlephp.org/en/stable/
- your favourite framework may also have specific features

HTTP With Streams

Make an HTTP request with PHP, no dependencies.

```php
$url = "https://httpbin.org/post";
$data = ["name" => "lornajane", "message" => "Hi there"];

$context = stream_context_create(
    ["http" => [
        "method" => "POST",
        "content" => http_build_query($data)
    ]
]);

$response = file_get_contents($url, false, $context);
```

HTTP With Guzzle

Use guzzlehttp\guzzle from Composer

```php
require "vendor/autoload.php";

$url = "https://httpbin.org/post";
$data = ["name" => "lornajane", "message" => "Hi there"];

$client = new \GuzzleHttp\Client();
$response = $client->request("POST", $url, [
    "form_params" => $data
]);
```

HTTPBin

https://httpbin.org is a site where you can make requests and get information about the responses.

- `/post` used above shows me the incoming post data and how it was parsed, other verbs also available
- `/response/*` offers a selection of canned responses, such as json data, `robots.txt` file
- `/status/*` returns the specified status code - great for testing error cases

RequestBin

Open source project you can run locally (Docker) or deploy (Heroku) to receive and diagnose any HTTP request.

https://github.com/runscope/requestbin (but try my `lornajane` fork for some fixes)

Useful for testing outgoing API requests, webhooks, anything really!

Exercise 3: PHP as an HTTP Client

Ngrok for Local Testing

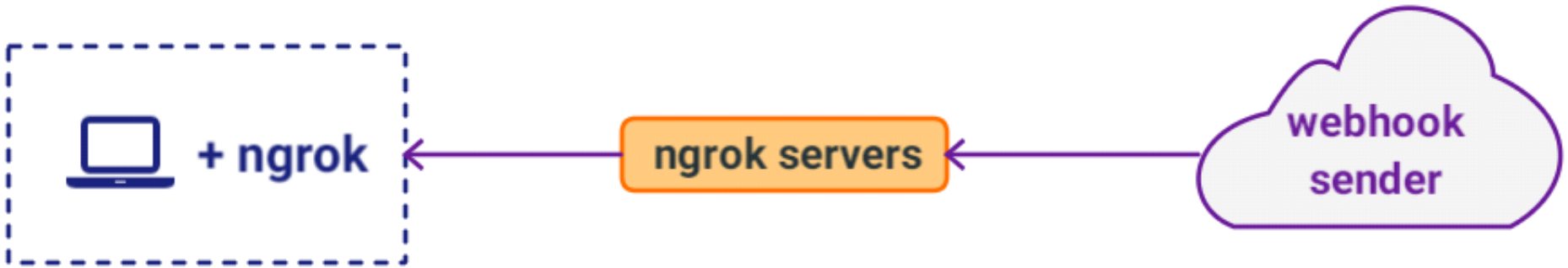
https://ngrok.com/ - secure tunnel to your dev platform

Use this tool to:

- access your local platform from anywhere
- inspect the request and response
- replay requests and see the responses

Ngrok for Testing Webhooks

Start the tunnel on your laptop: receive a public URL



Exercise 4: Try Ngrok!

Local Proxy Tools

Local Proxy Tools

Pass your traffic through a proxy to inspect (or change) it.

Some good examples:

- Charles Proxy (paid product, free trial available) https://www.charlesproxy.com/
- Mitmproxy (open source) https://mitmproxy.org/
- Wireshark is not a proxy but is also useful https://www.wireshark.org/

Transport Layer Security

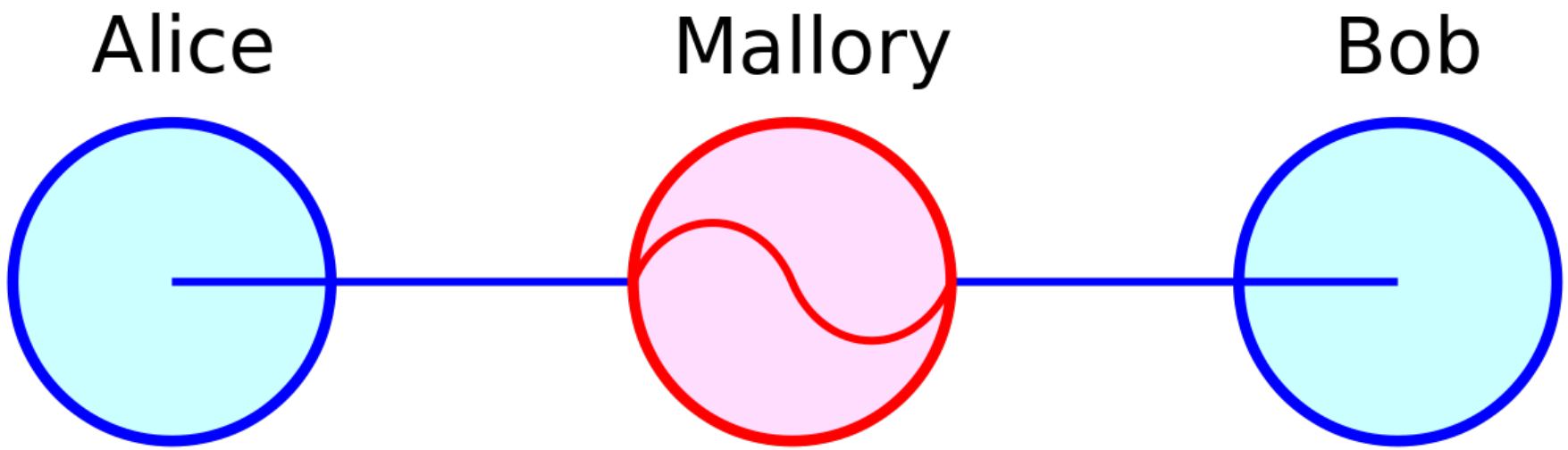
How secure HTTP traffic works:

1. Client meets server, they agree a secret and protocol to use.
2. Server can be identified since its public key or cert is available.
3. All traffic is encrypted and includes integrity checks to eliminate tampering.

This makes TLS traffic difficult to inspect!

Man in the Middle Attack

When a client mistakenly thinks it is talking to a server, but it is actually an impostor!



Performing MitM

Your devices will not TLS to a host that isn't verified by a Certificate Authority.

To debug TLS traffic:

- The proxy tools each have their own cert
- Install the cert into your browser/device
- Ignore security warnings
- Magic! You can inspect/alter even TLS traffic

Exercise 5: Superpower Debugging

Resources

- Feedback please: https://joind.in - this is a new session format for PHP UK and for me
- Website: https://lornajane.net (slides linked here)
- PHP Web Services has an HTTP chapter

For more resources on a specific tool: just ask/tweet!