

Interacting with External APIs



“Today you are You,
that is truer than true.
There is no one alive
who is Youer than
You.” - Dr. Seuss

Ben Lobaugh
@benlobaugh
ben@lobaugh.net
<http://ben.lobaugh.net/>



Bella
#bellacam on Twitter



Mariah

Session Overview

- What is an API
- HTTP concepts
- The PHP way
- WordPress magic
- Demo
- Advanced: Caching

What is an API?

API stands for Application Programming Interface

Simply put it is a way for external applications to tap into the data and functionality of another applications

HTTP Concepts

- HTTP stands for HyperText Transport Protocol
- Every website you visit uses HTTP to transfer data between the server (website) and client (browser)
- HTTP has several standard methods (or verbs) that describe actions

HTTP Concepts: Methods

- GET: Used to retrieve data from a resource.
Example: Viewing a website
- POST: Used to send data to a resource.
Example: Contact form submission
- HEAD: Used to retrieve info about a resource

HTTP Concepts: Response Codes

Status Code	Description
2xx	Request was successful
3xx	Request was redirected to another URL
4xx	Request failed due to client error. Usually invalid authentication or missing data
5xx	Request failed so a server error. Commonly missing or misconfigured configuration files

Status Code	Description
200	OK - Request was successful
301	Resource was moved permanently
302	Resource was moved temporarily
403	Forbidden - Usually due to an invalid authentication
404	Resource not found
500	Internal server error
503	Service unavailable

API calls the PHP way

PHP Example

```
function curlFile($url) {  
    $options = array(  
        CURLOPT_RETURNTRANSFER => true,           // return web page  
        CURLOPT_HEADER         => false,         // don't return headers  
        CURLOPT_FOLLOWLOCATION  => true,           // follow redirects  
        CURLOPT_ENCODING       => "",           // handle all encodings  
        CURLOPT_USERAGENT      => "blob curler 1.2", // who am i  
        CURLOPT_AUTOREFERER    => true,         // set referer on redirect  
        CURLOPT_CONNECTTIMEOUT => 120,         // timeout on connect  
        CURLOPT_TIMEOUT        => 120,         // timeout on response  
        CURLOPT_MAXREDIRS      => 10,         // stop after 10 redirects  
    );  
  
    $ch = curl_init( $url );  
    curl_setopt_array( $ch, $options );  
    $content = curl_exec( $ch );  
    $err = curl_errno( $ch );  
    $errmsg = curl_error( $ch );  
    $header = curl_getinfo( $ch );  
    curl_close( $ch );  
  
    $header['errno'] = $err;  
    $header['errmsg'] = $errmsg;  
    $header['content'] = $content;  
    return $header;  
}
```

Can WordPress do it better/faster/easier?

Introducing the HTTP API

WordPress provides a built-in HTTP API

The HTTP API contains simple functions to help you interact with external resources.

The HTTP API determines the best method for the remote call.

WordPress Example

```
$data = wp_remote_get( 'TheURL' );
```

Overview of HTTP API functions

Calling functions

- `wp_remote_get($url, $args);`
- `wp_remote_post($url, $args);`
- `wp_remote_head($url, $args);`
- `wp_remote_request($url, $args);`

Calling functions

```
$args = array(  
    'method'      => 'POST',  
    'timeout'     => 45,  
    'redirection' => 5,  
    'httpversion' => '1.0',  
    'blocking'    => true,  
    'headers'     => array(),  
    'body'        => array(),  
    'cookies'     => array()  
);
```

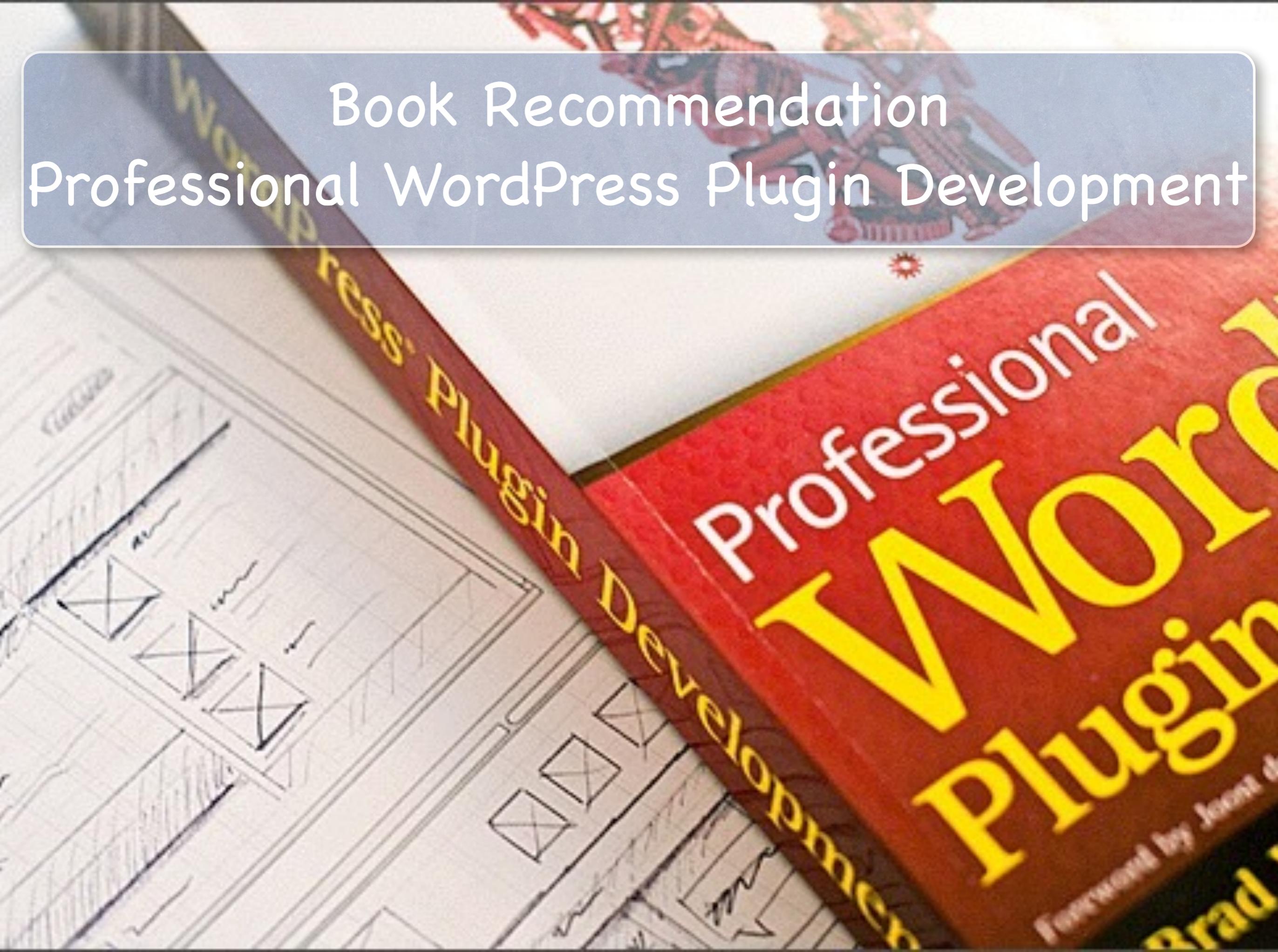
Response Functions

- `wp_remote_retrieve_body($response);`
- `wp_remote_retrieve_header($response);`
- `wp_remote_retrieve_headers($response);`
- `wp_remote_retrieve_response_message($rsp);`

CONSULT THE CODEX!

- The Codex is your friend
- Always check it before creating something new
- Using Core functions guarantees your code will always "just work"
- Core code has been looked at by many eyes

Book Recommendation
Professional WordPress Plugin Development

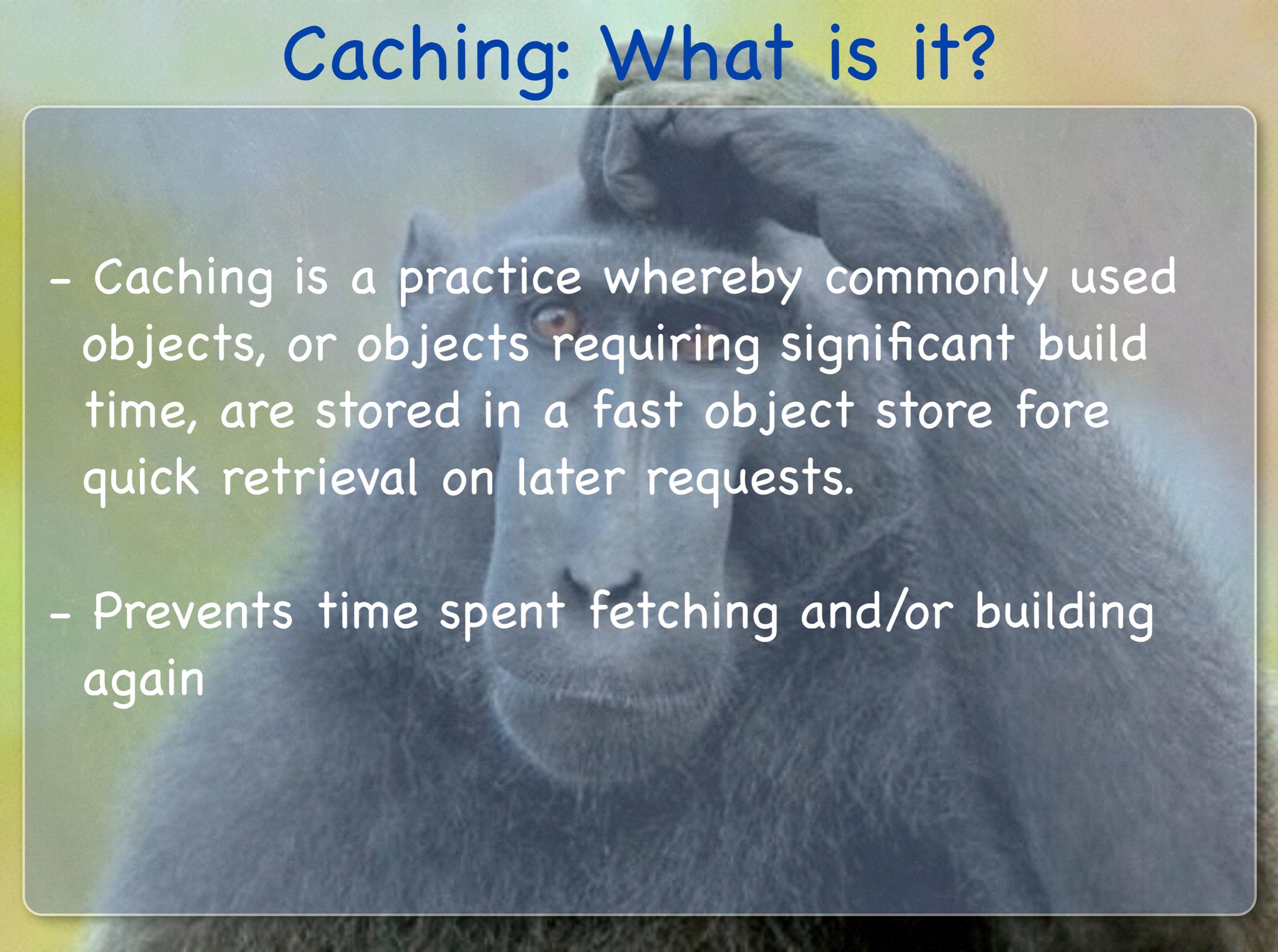


Demo Code: Retrieve list of post from ben.lobaugh.net



Getting Advanced with Caching

Caching: What is it?

A close-up photograph of a blue cat's face. The cat has its right paw pressed against its forehead, with its fingers slightly curled. The cat's eyes are looking directly at the camera with a thoughtful or curious expression. The background is a soft, out-of-focus light blue and green.

- Caching is a practice whereby commonly used objects, or objects requiring significant build time, are stored in a fast object store for quick retrieval on later requests.
- Prevents time spent fetching and/or building again

Caching: When to use it?

- **Always** cache - except for...
- Do not cache real time data
- Do not cache when an API recommends not to
- Especially cache when retrieving large amounts of data and/or doing long computations

Introducing Transients

- Similar in use to the Options API
- Convenient way to store and use cached objects
- Will “just work” with caching plugins
- Live for a specified amount of time before being updated

Cache an object

- Done with `set_transient($name, $value, $expiration)`
- `$name` - Name of transient to store
- `$value` - Value of transient
- `$expiration` - Number of seconds until transient expires and is refreshed

```
$response = wp_remote_get( 'https://api.github.com/users/lobaugh' );  
  
set_transient( 'lobaugh_github_userinfo', $response, 60*60 );
```

Retrieve a cached object

- Uses `get_transient()`, simply pass the transient name, however usage is more complex

```
$github_userinfo = get_transient( 'blobaugh_github_userinfo' );

if( $false === $github_userinfo ) {
    // Transient expired, refresh the data
    $response = wp_remote_get( 'https://api.github.com/users/blobaugh' );
    set_transient( 'blobaugh_github_userinfo', $response, 60*60 );
}

// Use $github_userinfo as you will
```

Bonus: Basic Authentication

```
$args = array(  
    'headers' => array(  
        'Authorization' => 'Basic ' . base64_encode( YOUR_USERNAME . ':' .  
        YOUR_PASSWORD )  
    )  
);  
wp_remote_request( $url, $args );
```

<http://lud.icro.us/wordpress-http-api-basicauth>

Questions?



*Obligatory Cat Picture