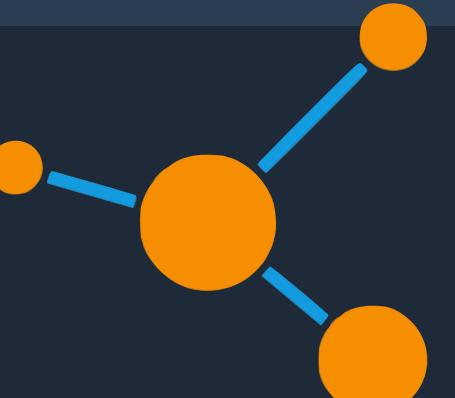


# The WordPress API

Web applications for the REST of us!



@nagmay

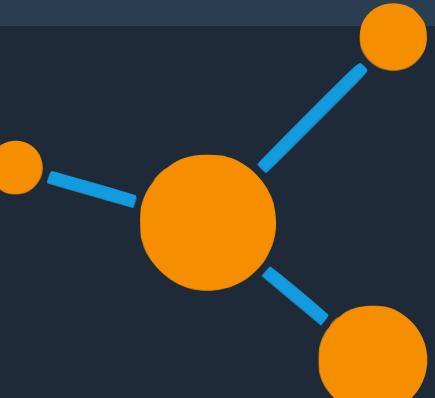
# What the what?

Overview and history

Pulling data

— Intermission: Let's talk about beer —

Pushing data

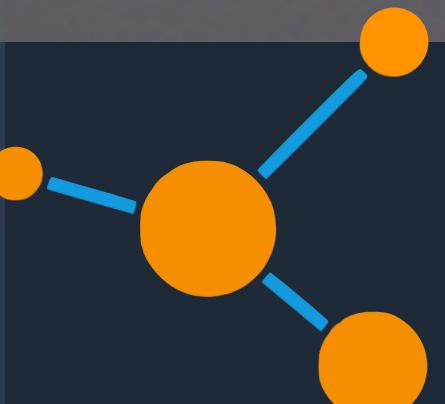


The WordPress API

@nagmay

**Gabriel Nagmay**  
[gabriel@nagmay.com](mailto:gabriel@nagmay.com)

**Web Analyst  
Programmer  
Developer**



**The WordPress API**



**@nagmay**

**Gabriel Nagmay**  
[gabriel@nagmay.com](mailto:gabriel@nagmay.com)

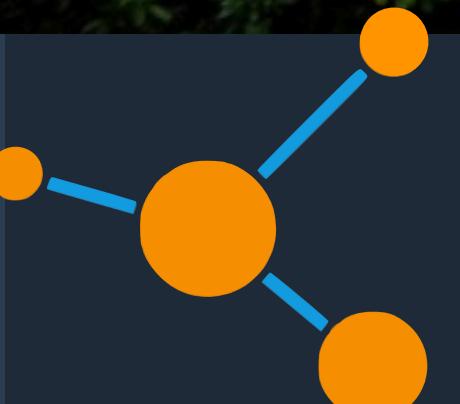
Bee Keeper  
Coffee Lover  
Home Brewer



The WordPress API



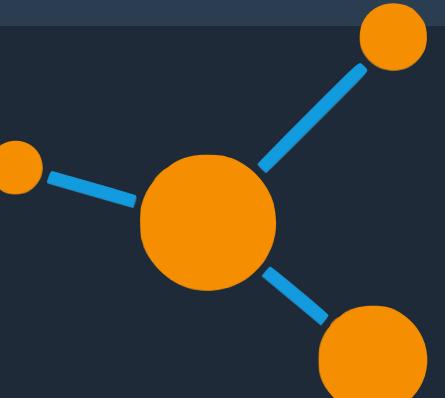
@nagmay



# The WordPress API

@nagmay

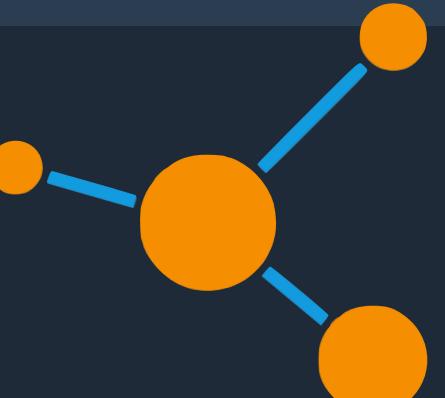
# Application Protocol Interface



The WordPress API

@nagmay

# REpresentational State Transfer



The WordPress API

@nagmay

# HTTP Methods

POST .../posts



Create

GET .../posts



Retrieve

PUT .../posts

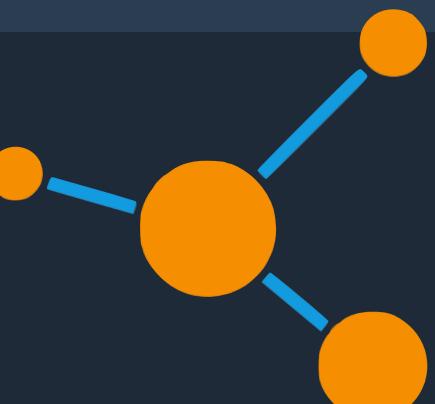


Update

DELETE .../posts



Delete



The WordPress API

@nagmay

# JavaScript

Coldfusion

Python

# Ruby

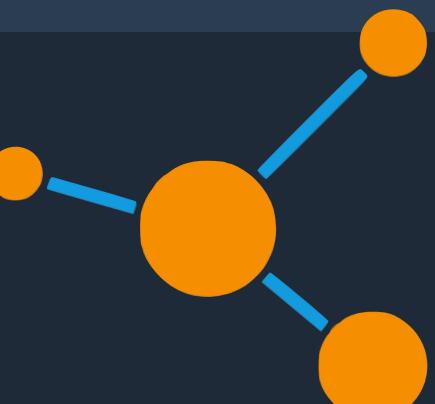
# PHP

Java

# C#

Bash

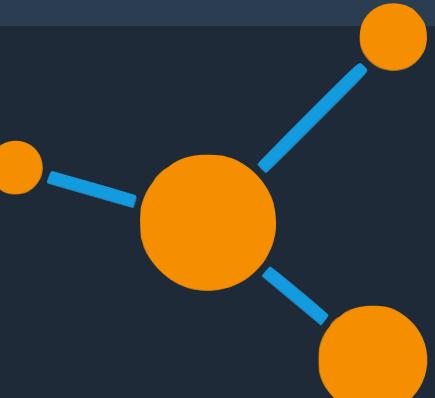
The WordPress API



@nagmay

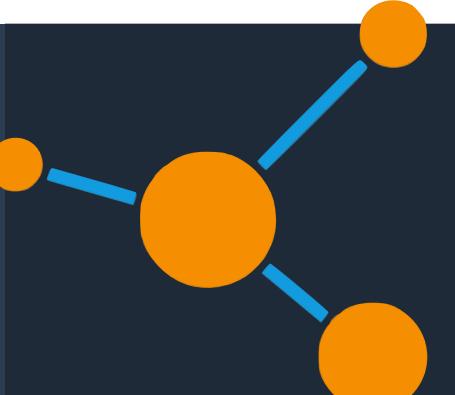
# WP-API History

- 2013 GSOC project by Ryan McCue
- 2014 v1.0 released as WP-API plugin
- 2015 v2.0 developed  
Infrastructure added to WP4.4
- 2016 Endpoints added to WP4.7



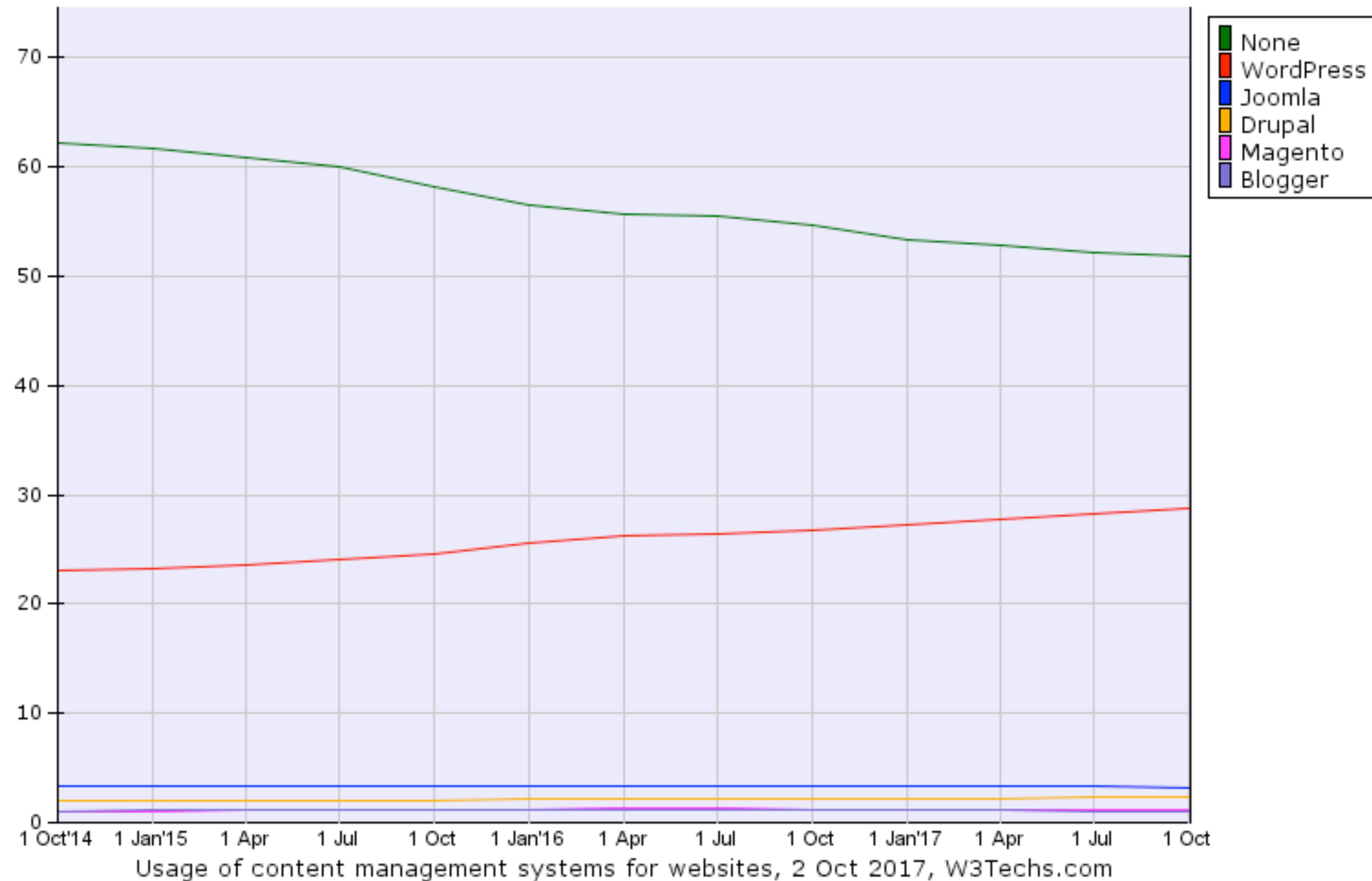


# WORDPRESS



The WordPress API

@nagmay



The WordPress API

@nagmay

# 28.7% of the web

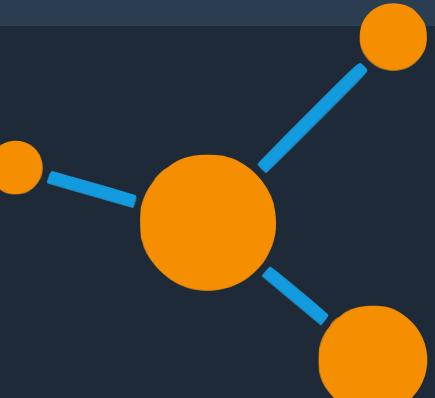
Now available to you via an API



- WordPress
- Other CMS
- None

The WordPress API

@nagmay



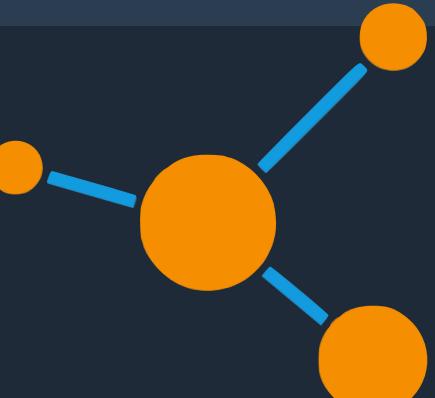
# Tools

Web browser

Any language that can talk HTTP methods

Recommended

- WP REST API Handbook
- Postman: [getpostman.com](https://getpostman.com)



news.pcc.edu/wp-json/

<https://news.pcc.edu/wp-json/>**JSON Raw Data Headers**

Save Copy

```
name: "News"
description: ""
url: "https://news.pcc.edu"
home: "https://news.pcc.edu"
gmt_offset: -7
timezone_string: "America/Los_Angeles"
namespaces:
  0: "oembed/1.0"
  1: "akismet/v1"
  2: "jetpack/v4"
  3: "wp-super-cache/v1"
  4: "wp/v2"
authentication: []
routes: {...}
_links: {...}
```

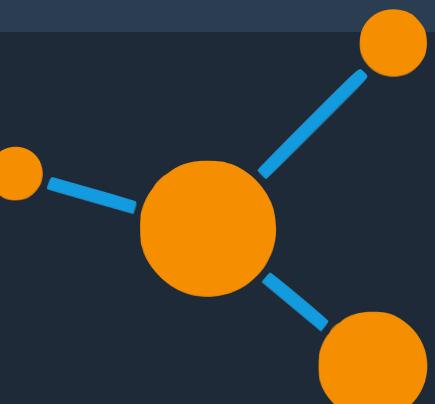


# The WordPress API

@nagmay

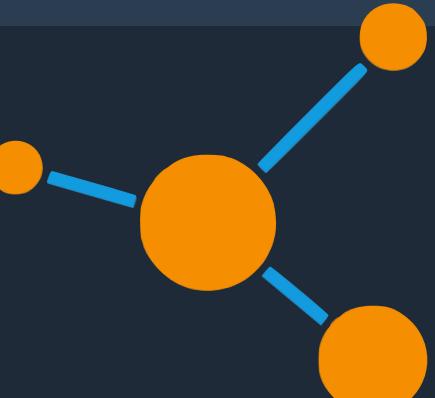
# Routes

namespace:	"wp/v2"
routes:	
▶ /wp/v2:	Object
▶ /wp/v2/posts:	Object
▶ /wp/v2/posts/(?P<id>[\d]+):	Object
▶ /wp/v2/posts/(?P<parent>[\d]+)/revisions:	Object
▶ /wp/v2/posts/(?P<parent>[\d]+)/revisions/(?P<id>[\d]+):	Object
▶ /wp/v2/pages:	Object
▶ /wp/v2/pages/(?P<id>[\d]+):	Object
▶ /wp/v2/pages/(?P<parent>[\d]+)/revisions:	Object
▶ /wp/v2/pages/(?P<parent>[\d]+)/revisions/(?P<id>[\d]+):	Object
▶ /wp/v2/media:	Object
▶ /wp/v2/media/(?P<id>[\d]+):	Object
▶ /wp/v2/feedback:	Object



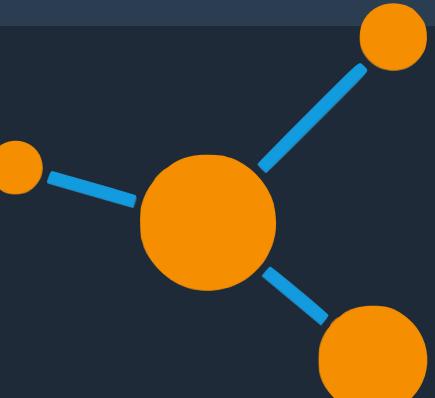
# Methods

```
namespace: "wp/v2"
routes:
  ▶ /wp/v2: Object
    ▶ /wp/v2/posts:
      namespace: "wp/v2"
      methods:
        0: "GET"
        1: "POST"
      endpoints:
        ▶ 0: Object
        ▶ 1: Object
      _links:
        self:
          ▶ /wp/v2/posts/(?P<id>[\d]+): Object
          ▶ /wp/v2/posts/(?P<id>[\d]+)/comment-count: Object
```



# Endpoints

Method	+	Route	Result
POST		wp/v2/posts/123	Create a post
GET		wp/v2/posts/123	Retrieve a post
PUT		wp/v2/posts/123	Update a post
DELETE		wp/v2/posts/123	Delete a post



# Examples

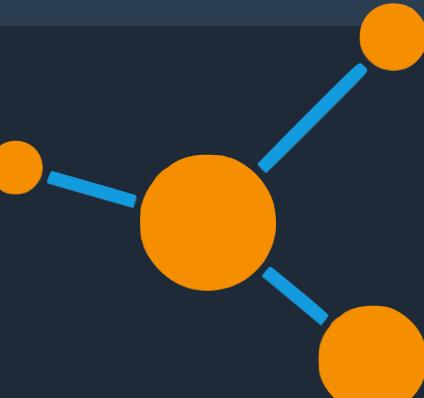
[github.com/nagmay/WRK7](https://github.com/nagmay/WRK7)

“WTF? Why did you choose JS for your examples”

- *someone in this workshop?*



The WordPress API



@nagmay



# Developer Resources

Browse: [Home](#) / [REST API Handbook](#) / REST API Handbook

## CHAPTERS

### 1 REST API Handbook

### 2 Reference

- Posts
- Post Revisions
- Categories
- Tags
- Pages
- Comments
- Taxonomies
- Media
- Users
- Post Types
- Post Statuses

# REST API Handbook

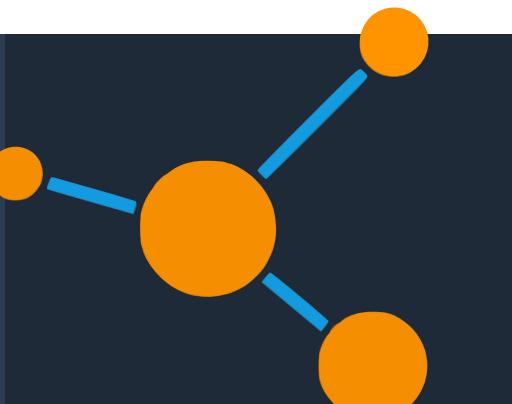
The WordPress REST API provides API endpoints for WordPress data types that allow developers to interact with sites remotely by sending and receiving [JSON](#) (JavaScript Object Notation) objects. JSON is an open standard data format that is lightweight and human-readable, and looks like Objects do in JavaScript; hence the name. When you send content to or make a request to the API, the response will be returned in JSON. This enables developers to create, read and update WordPress content from client-side JavaScript or from external applications, even those written in languages beyond PHP.

## TOPICS

### Why use the WordPress REST API

### Key Concepts

- [Routes & Endpoints](#)
- [Requests](#)
- [Responses](#)
- [Schema](#)
- [Controller Classes](#)



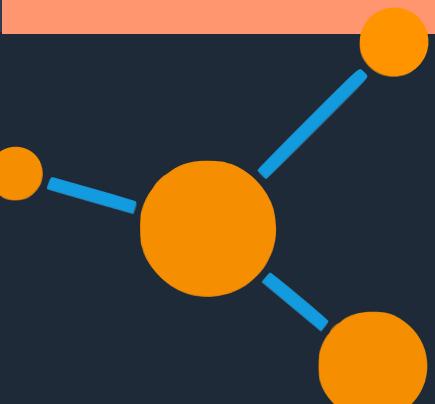
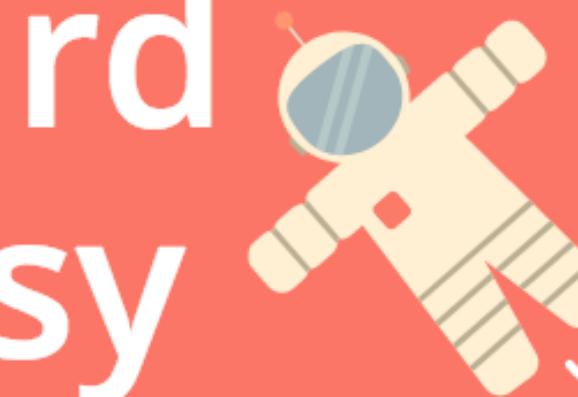
# The WordPress API

@nagmay



# POSTMAN

Developing APIs is hard  
Postman makes it easy



The WordPress API

@nagmay

# Pulling Data

POST .../posts



Create

GET .../posts



Retrieve

PUT .../posts

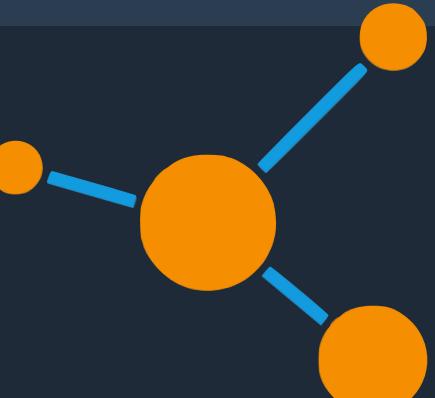


Update

DELETE .../posts



Delete



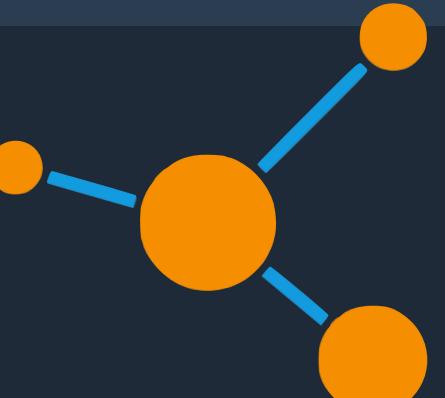
The WordPress API

@nagmay

# 01: GET JSON

```
$.ajax({
  method: "GET",
  url: url,
  dataType: "json"
}).done(function( data ) {
  // do something
});
```

```
$.getJSON(url, function (data) {
  // do something
});
```



# 02: GET Posts

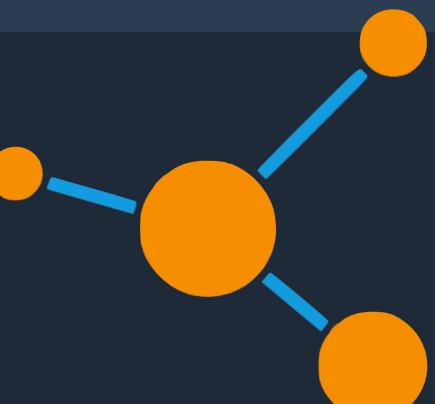
```
  ▼ title:  
    ▼ rendered: "Home to nursing and medical imaging, HT  
      comprehensive renovation"  
  
data[i].title.rendered
```

### Response Headers

---

**Access-Control-Allow-Headers** Authorization, Content-Type  
**Access-Control-Expose-Headers** X-WP-Total, X-WP-TotalPages

```
numPosts = request.getResponseHeader('x-wp-total');
```



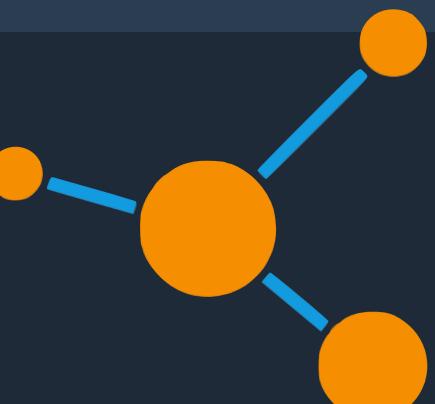
# 03: Posts Arguments

`http://news.pcc.edu/wp-json/wp/v2/posts/?page=1&search=biology`

http://news.pcc.edu GET some

Refine:

page	<input type="text" value="1"/>
per_page	<input type="text" value="#"/>
search	<input type="text" value="biology"/>
after	<input type="text" value="0000-00-00T00:00:00"/>
author *	<input type="text" value="user_id #"/>



# 04: Posts by year?

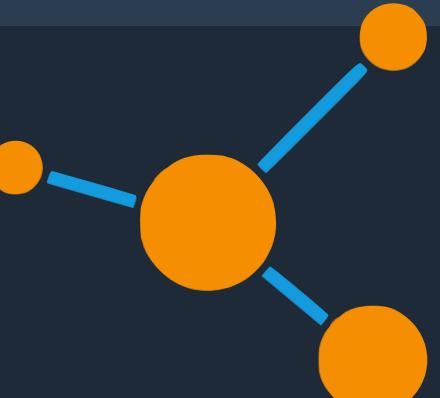
```
var url = $('#site').val()
    + '/wp-json/wp/v2/posts/?per_page=1&after='
    + year + '-01-01T00:00:00&before='
    + (year+1) + '-01-01T00:00:00';

// Ajax
$.getJSON(url, function (data, status, request) { // if success
    numPosts = request.getResponseHeader('x-wp-total');

    // Keep going?
    year -= 1;

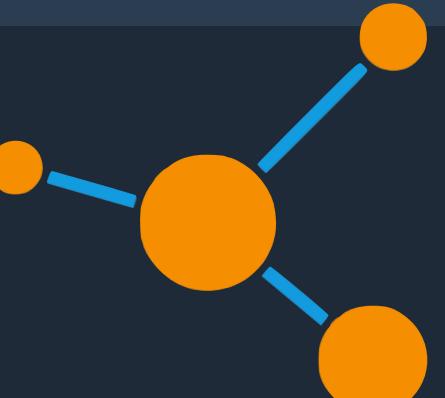
    ...
}

});
```



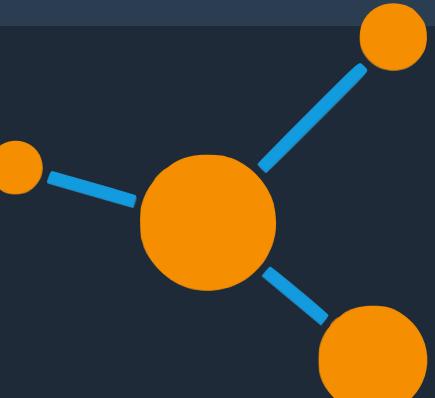
# 05: Posts by year? Smarter!

```
// Ajax 1: How old is this site?  
var url = $('#site').val()  
    + '/wp-json/wp/v2/posts?per_page=1&order=asc'; // Get  
$.getJSON(url, function (data, status, request) { // if success:  
  
    // Find the year of the first post  
    until = data[0].date;  
    until = until.substring(0,4);  
  
    ...  
});  
  
// Ajax 2: Get number of posts for each year  
...
```



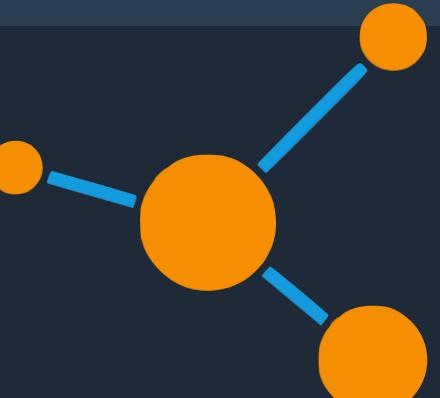
# 06: Comments by year

```
// Ajax 1: How old is this site?  
var url = $('#site').val()  
    + '/wp-json/wp/v2/comments?per_page=1&order=asc';  
$.getJSON(url, function (data, status, request) {  
  
    // Find the year of the first post  
    until = data[0].date;  
    until = until.substring(0,4);  
  
    ...  
});  
  
// Ajax 2: Get number of comments for each year  
...
```



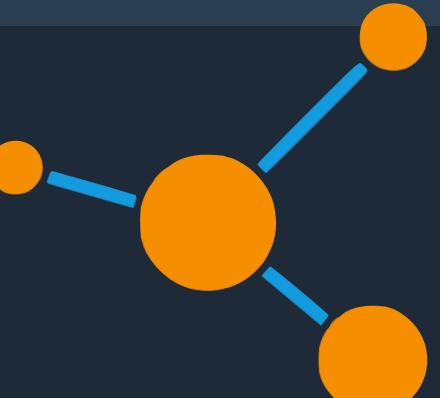
# 07: Media by year

```
// Ajax 1: How old is this site?  
var url = $('#site').val()  
    + '/wp-json/wp/v2/media?per_page=1&order=asc';  
$.getJSON(url, function (data, status, request) {  
  
    // Find the year of the first post  
    until = data[0].date;  
    until = until.substring(0,4);  
  
    ...  
});  
  
// Ajax 2: Get number of media uploads for each year  
...
```



# 08: Custom post types

```
/* === Custom Post Type: 'pcc_quotes' === */
add_action( 'init', 'pcc_register_custom' );
function pcc_register_custom() {
    // custom post types
    $post_args = array(
        'description' => '',
        'public' => true,
        'show_in_rest' => true,
        ...
    );
    register_post_type( "pcc_quotes", $post_args );
```

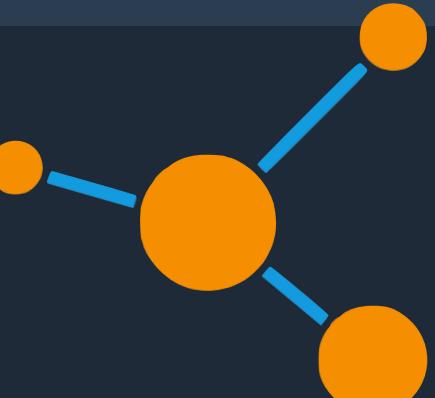


# 09: Users

```
GET http://news.pcc.edu/wp-json/wp/v2/users/?page=1
{"readyState":4,"responseText":"{\\"code\\":\\"rest_user_cannot_view\\",\\"message\\":\\"Sorry, you are not allowed to list users.\\"},\\"data\\":{\\"status\\":
```

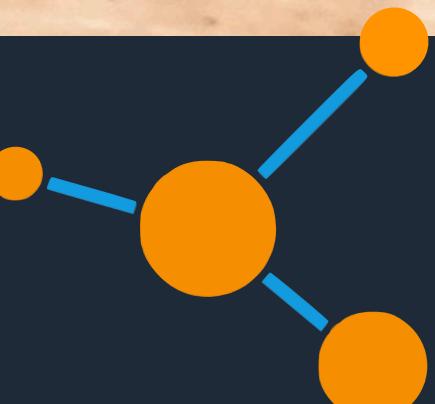
Translation:

“Come back with a warrant”



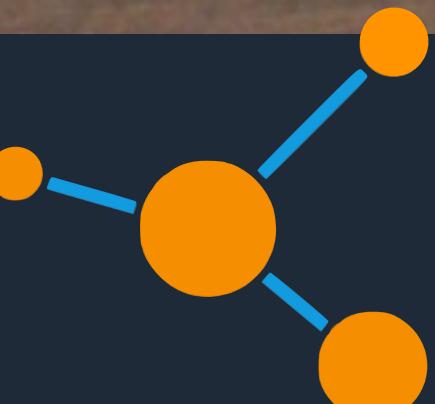


Let's talk  
about beer



The WordPress API

@nagmay



## The WordPress API

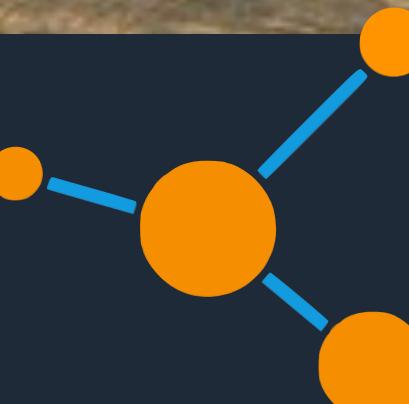


@nagmay



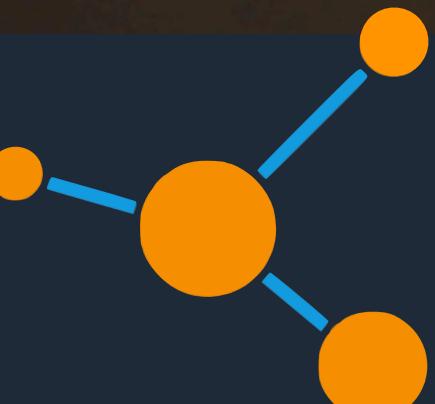
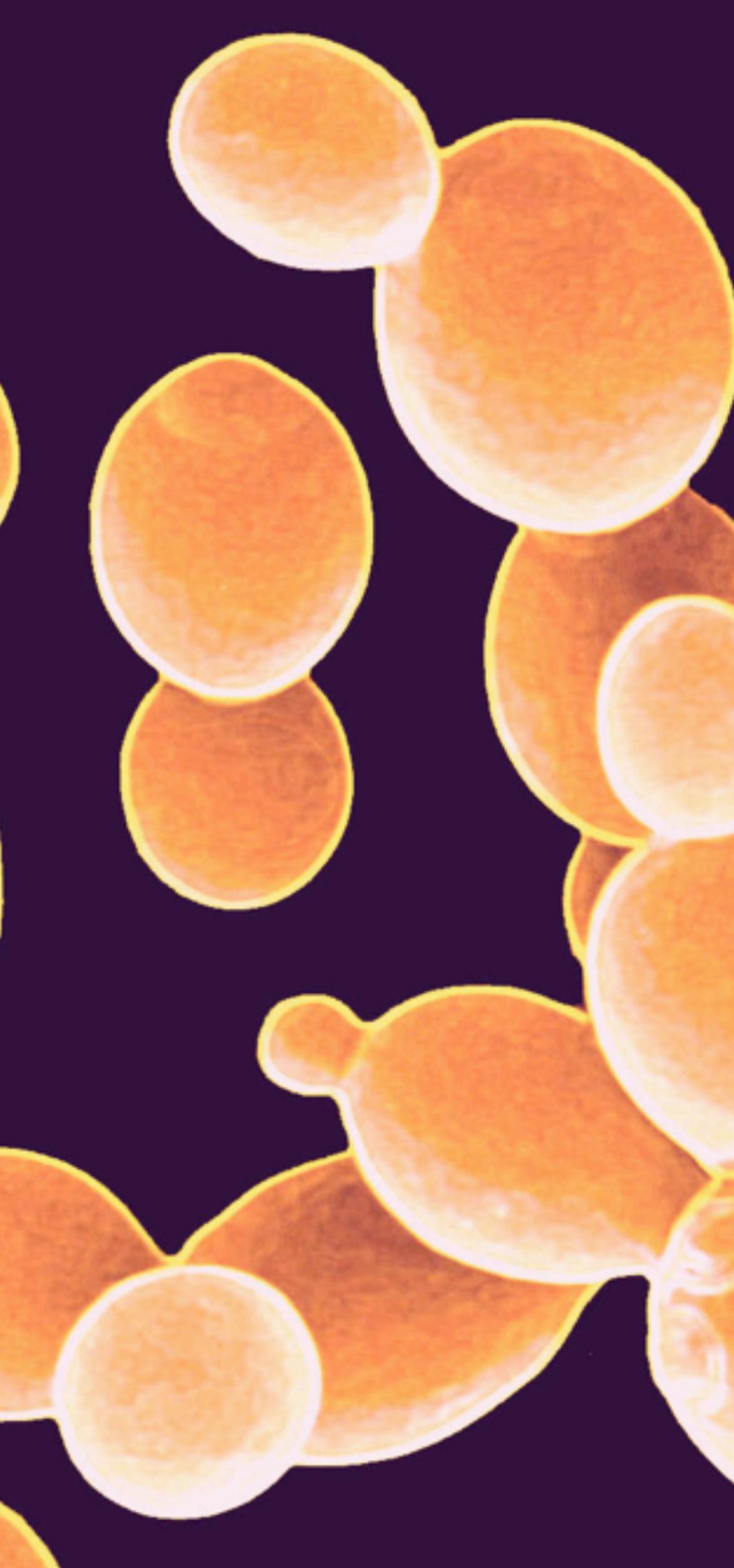
The WordPress API

@nagmay



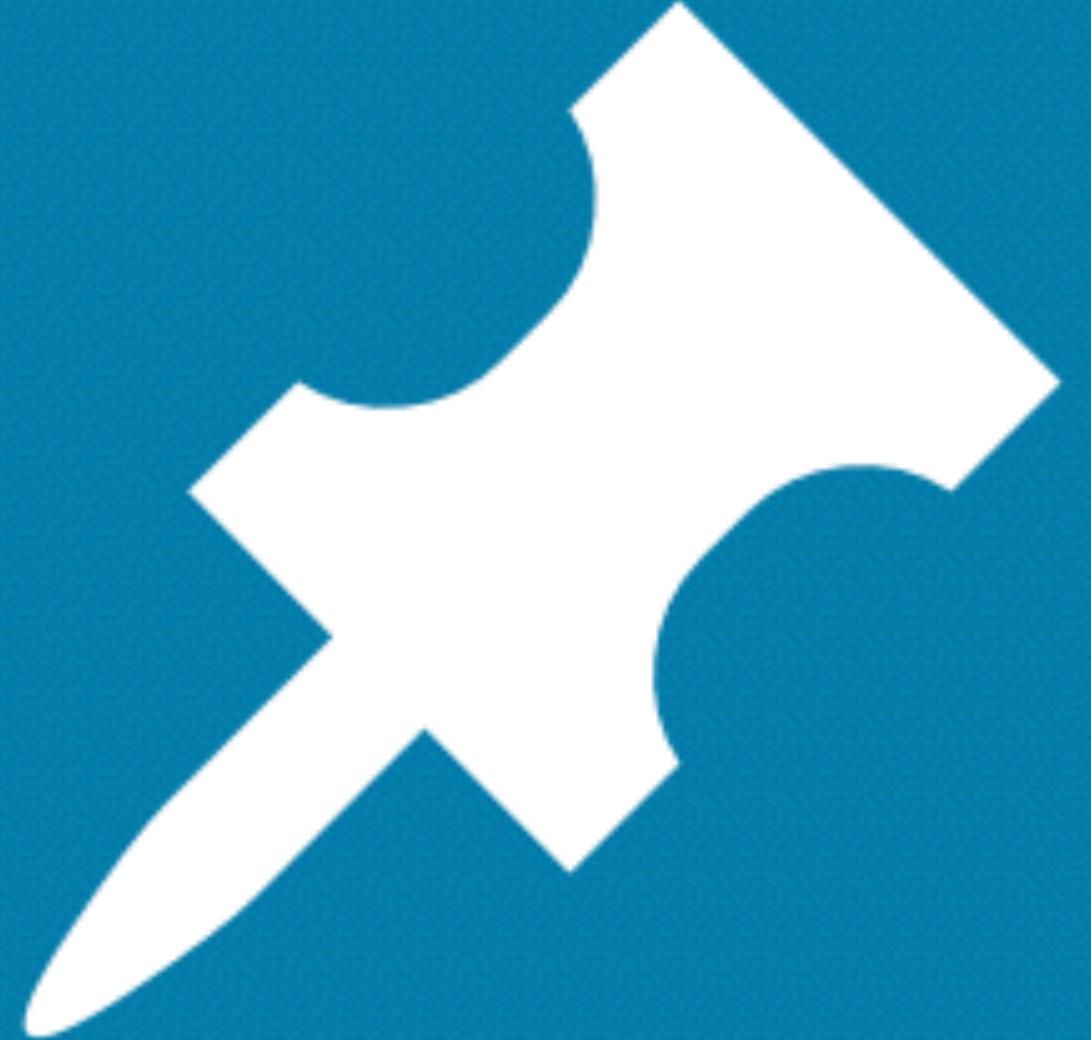
The WordPress API

@nagmay



The WordPress API

@nagmay



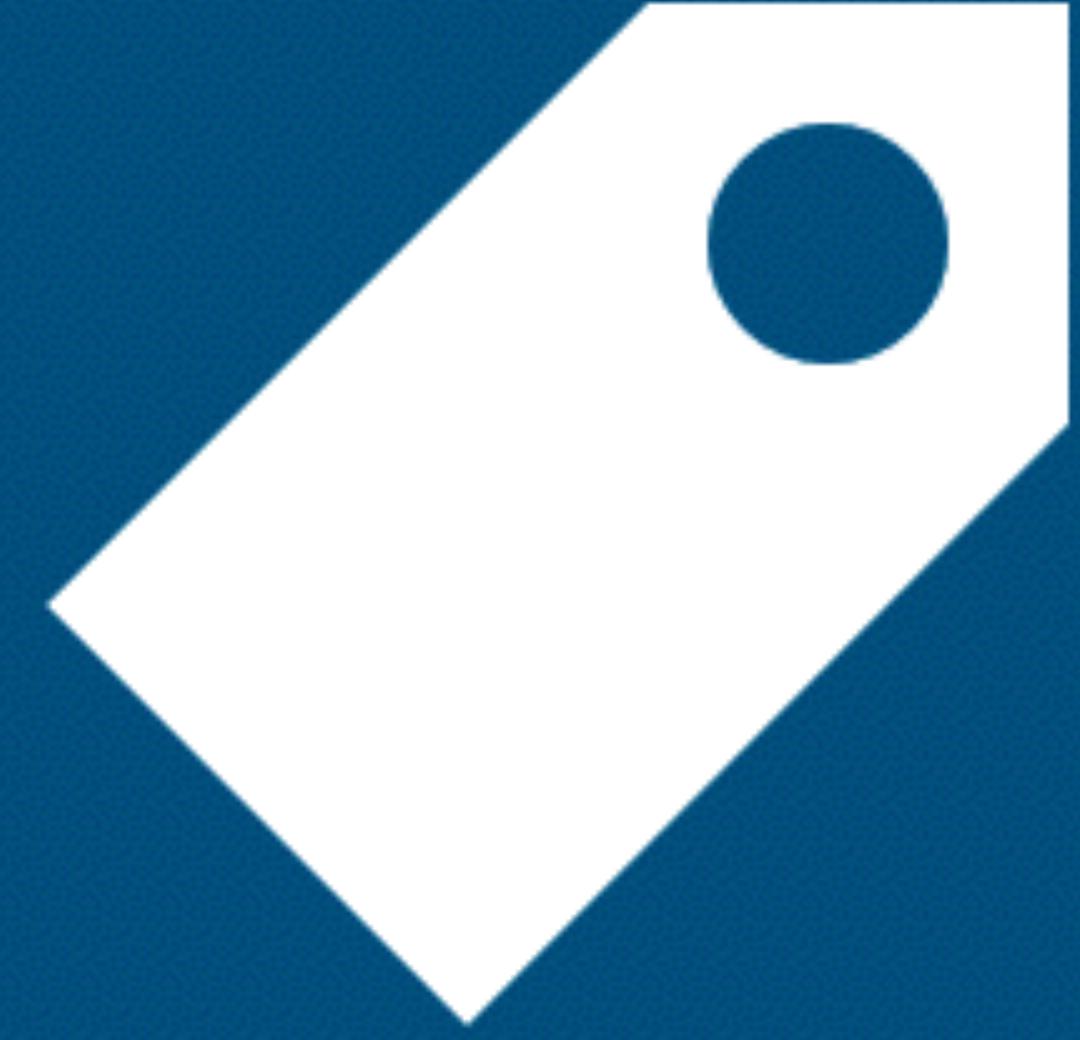
Posts



Pages



Media



Tags



The WordPress API

@nagmay

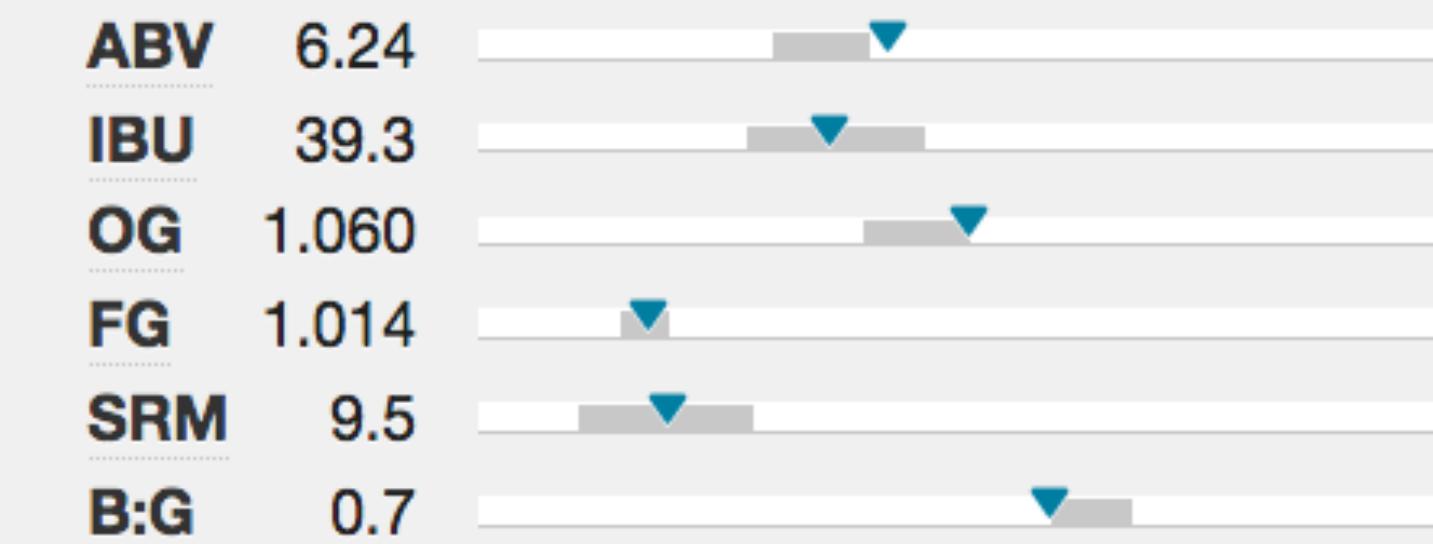
# ALL GRAIN BEER



HTM-Ale

10A: American Pale Ale

5 gal    60 min    70 %



	Grain	Amount	Potential	SRM	SG +	SRM +	
Grain	Pale Malt (2 Row) US	10 lb	36	2	1.050	3.9	x
Hops	Caramel/Crystal Malt – 20L	1.5 lb	35	20	1.007	5.1	x
Yeast	Cara-Pils/Dextrine	0.5 lb	33	2	1.002	0.5	x
Water							
	+ Add some grain						

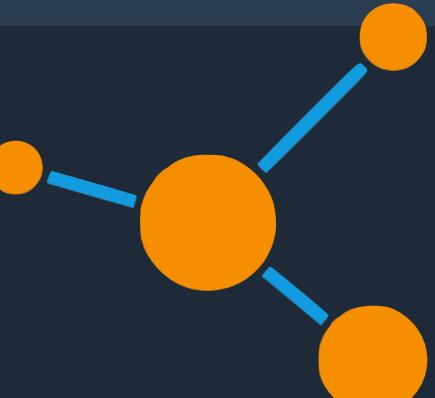
The WordPress API

@nagmay

# 10: Random Beer App

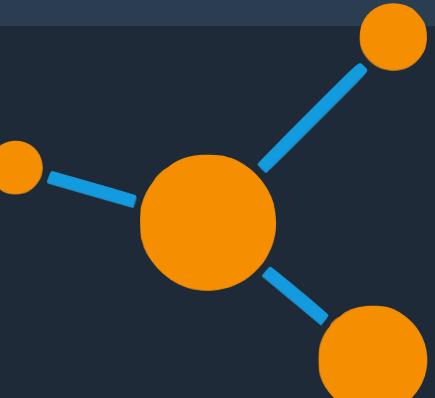
Can you build a simple app using only:

- Site address: <http://allgrain.beer>
- And what we know so far?



# 10: Meta

```
/* =====
 * REST: Custom Fields
 * =====
add_action('init', 'agb_register_meta');
function agb_register_meta(){
    $args = array(
        'type' => 'string',
        'show_in_rest' => true,
        'object_subtype' => 'agb_recipe'
    );
    register_meta( 'post', 'agb_recipe_abv', $args );
    register_meta( 'post', 'agb_recipe_boil', $args );
    register_meta( 'post', 'agb_recipe_efficiency', $args );
    ...
}
```



# Pushing Data

POST .../posts



Create

GET .../posts



Retrieve

PUT .../posts

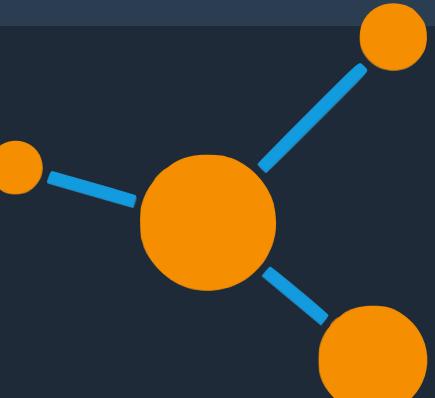


Update

DELETE .../posts



Delete



The WordPress API

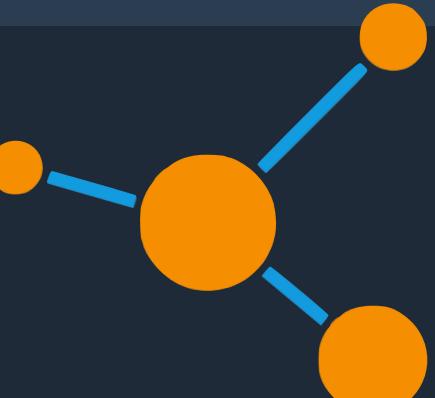
@nagmay

# Authentication

Cookies &Nonce

OAuth plugin

Basic Auth plugin



The WordPress API

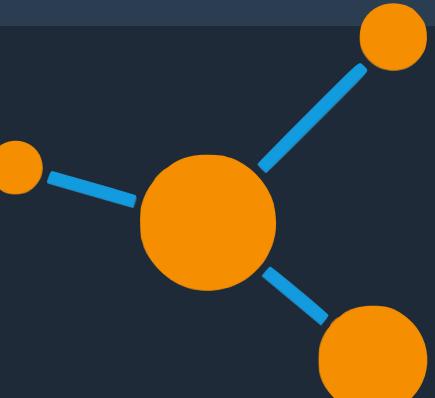
@nagmay

# Cookies &Nonce

Super simple :)

Only available inside of WordPress :(

```
$.ajax( {  
    url: wpApiSettings.root + 'wp/v2/posts/1',  
    method: 'POST',  
    beforeSend: function ( xhr ) {  
        xhr.setRequestHeader( 'X-WP-Nonce', wpApiSettings.nonce );  
    },  
    data:{  
        'title' : 'Hello Moon'  
    } ).done( function ( response ) {  
        console.log( response );  
    } );
```



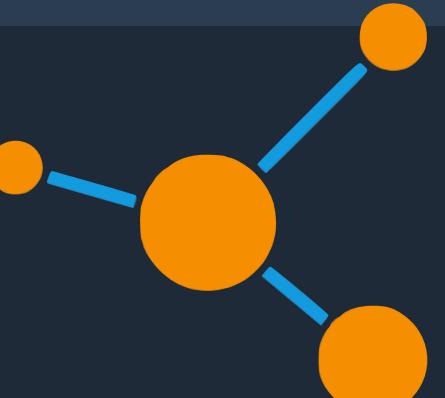
The WordPress API

@nagmay

# OAuth: Server

## WordPress REST API – OAuth 1.0a Server

```
name:          "All Grain Beer"  
description:   "build, create, and manage homebrew recipes"  
url:           "http://allgrain.beer"  
home:          "http://allgrain.beer"  
gmt_offset:    -7  
timezone_string: "America/Los_Angeles"  
▶ namespaces:  [...]  
▼ authentication:  
  ▶ oauth1:  
    request:      "http://allgrain.beer/oauth1/request"  
    authorize:    "http://allgrain.beer/oauth1/authorize"  
    access:       "http://allgrain.beer/oauth1/access"  
    version:      "0.1"  
  ▶ routes:      {...}
```



# Basic Auth plugin

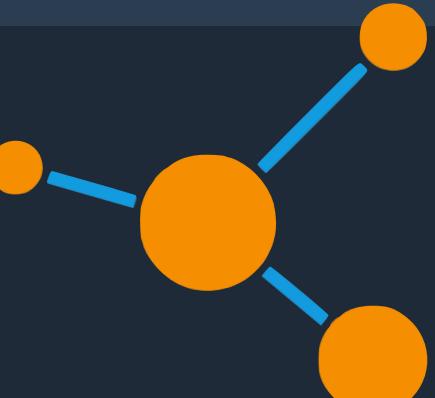
[github.com/WP-API/Basic-Auth](https://github.com/WP-API/Basic-Auth)

**WARNING:** Requires sending your username and password with every request

```
# BEGIN Basic Auth

<IfModule mod_rewrite.c>
RewriteRule ^index\.php$ - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization},L]
</IfModule>

# END Basic Auth
```

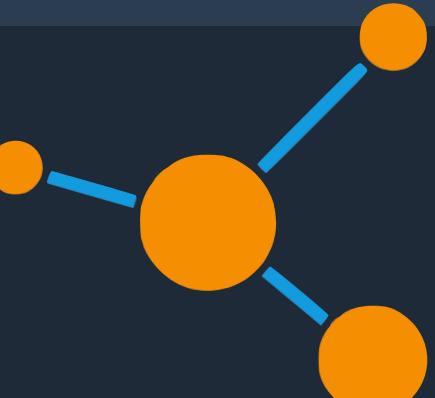


# 11: Get Posts (authorized!)

```
// The auth credentials
user = $('#user').val()
pass = $('#pass').val()

// encode 'username:password' in base 64
creds = btoa( user + ':' + pass );

// Ajax, well ... duh!
$.ajax({
    url: url,
    method: 'GET',
    crossDomain: true,
    beforeSend: function ( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'Basic ' + creds );
    }
})
```

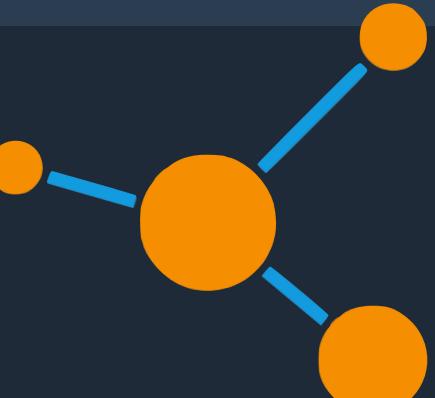


# 12: Create Posts

```
// The auth credentials
user = $('#user').val()
pass = $('#pass').val()

// encode 'username:password' in base 64
creds = btoa( user + ':' + pass );

// Ajax, well ... duh!
$.ajax({
    url: url,
    method: 'POST', // << hey, something new!
    crossDomain: true,
    beforeSend: function ( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'Basic ' + creds );
    }
})
```

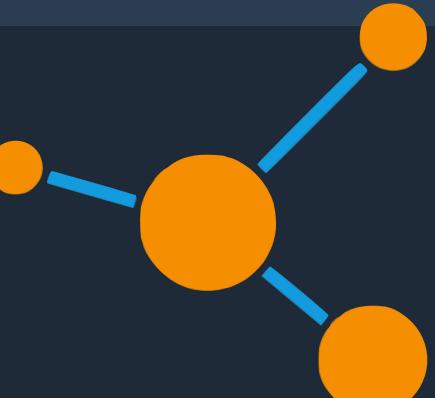


# 13: Update Posts

```
// The auth credentials
user = $('#user').val()
pass = $('#pass').val()

// encode 'username:password' in base 64
creds = btoa( user + ':' + pass );

// Ajax, well ... duh!
$.ajax({
    url: url,
    method: 'PUT', // Could also use POST
    crossDomain: true,
    beforeSend: function ( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'Basic ' + creds );
    }
})
```

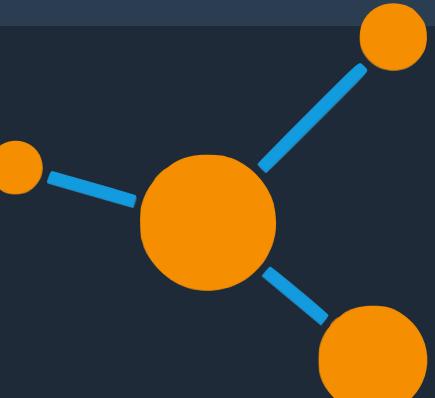


# 14: Delete Posts

```
// The auth credentials
user = $('#user').val()
pass = $('#pass').val()

// encode 'username:password' in base 64
creds = btoa( user + ':' + pass );

// Ajax, well ... duh!
$.ajax({
    url: url,
    method: 'DELETE', // Goodbye!
    crossDomain: true,
    beforeSend: function ( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'Basic ' + creds );
    }
})
```

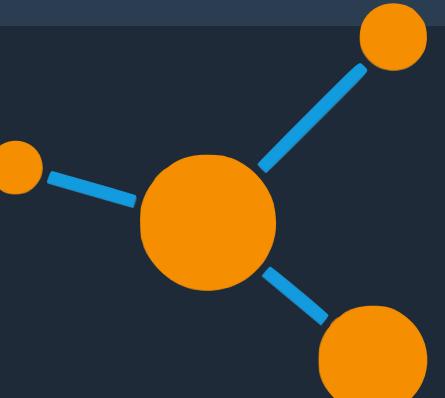


# 15: Users

```
// The auth credentials
user = $('#user').val()
pass = $('#pass').val()

// encode 'username:password' in base 64
creds = btoa( user + ':' + pass );

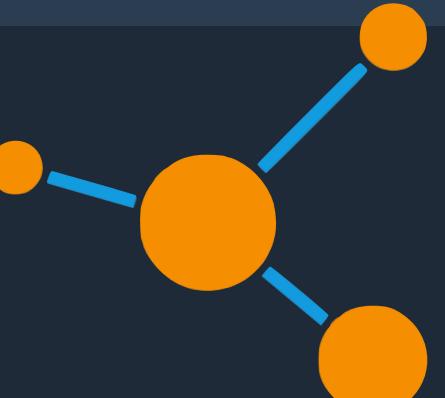
// Ajax, well ... duh!
$.ajax({
    url: url,
    method: 'GET',
    crossDomain: true,
    beforeSend: function ( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'Basic ' + creds );
    }
})
```



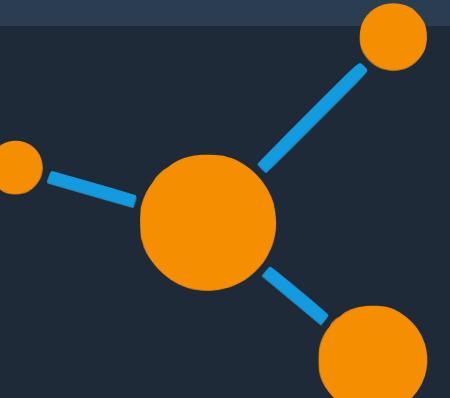
# 16: Media

```
var formData = new FormData();
formData.append( 'file', file );
formData.append( 'title', title );
formData.append( 'alt_text', alt_text );

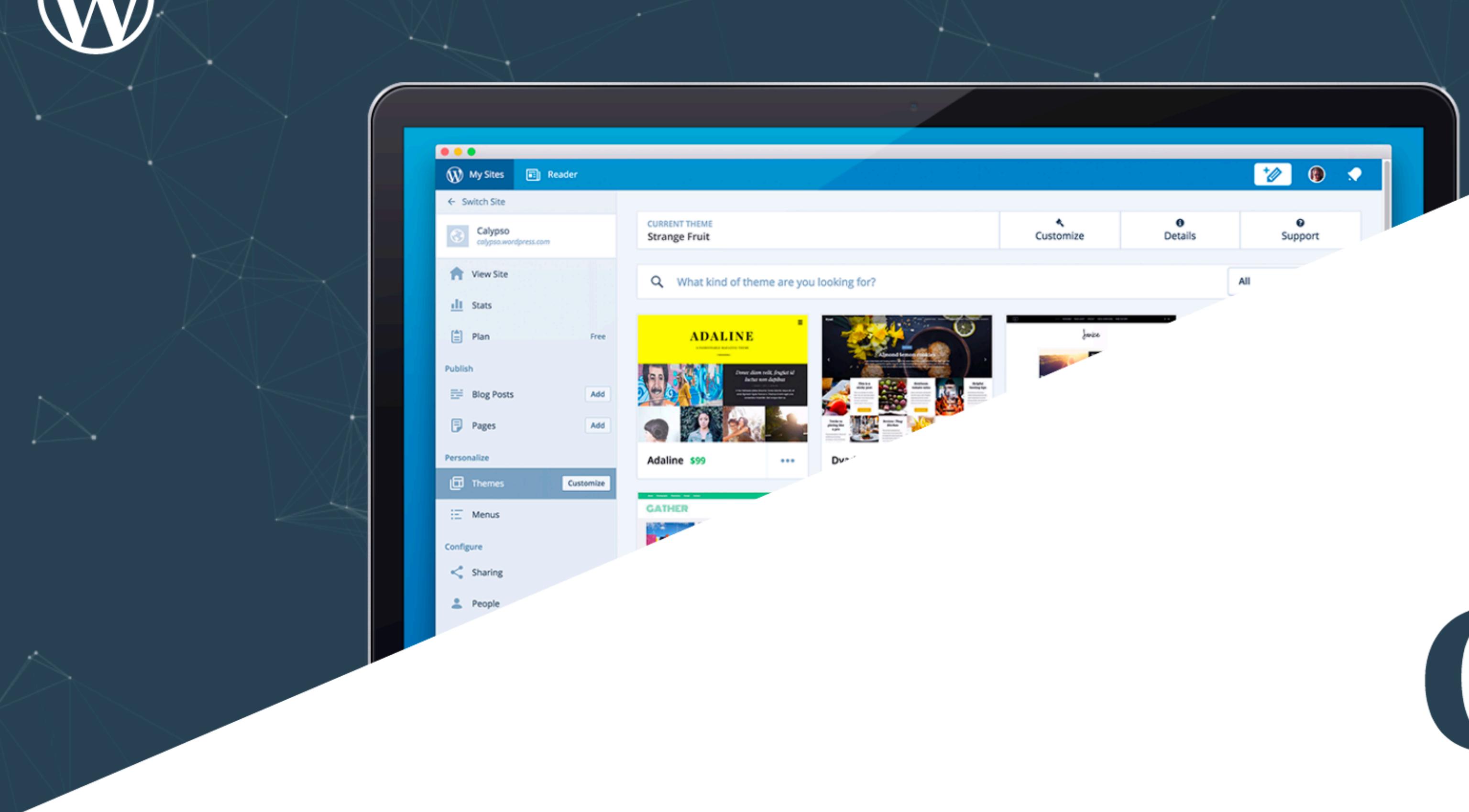
// Ajax, well ... duh!
$.ajax({
    url: url,
    method: 'POST',
    crossDomain: true,
    processData: false, // don't process into urlencoded content type
    contentType: false, // don't send the default content type header
    data: formData, // << here's all your form data!
    beforeSend: function ( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'Basic ' + creds );
    }
})
```



# Real world examples?



@nagmay



*Introducing*

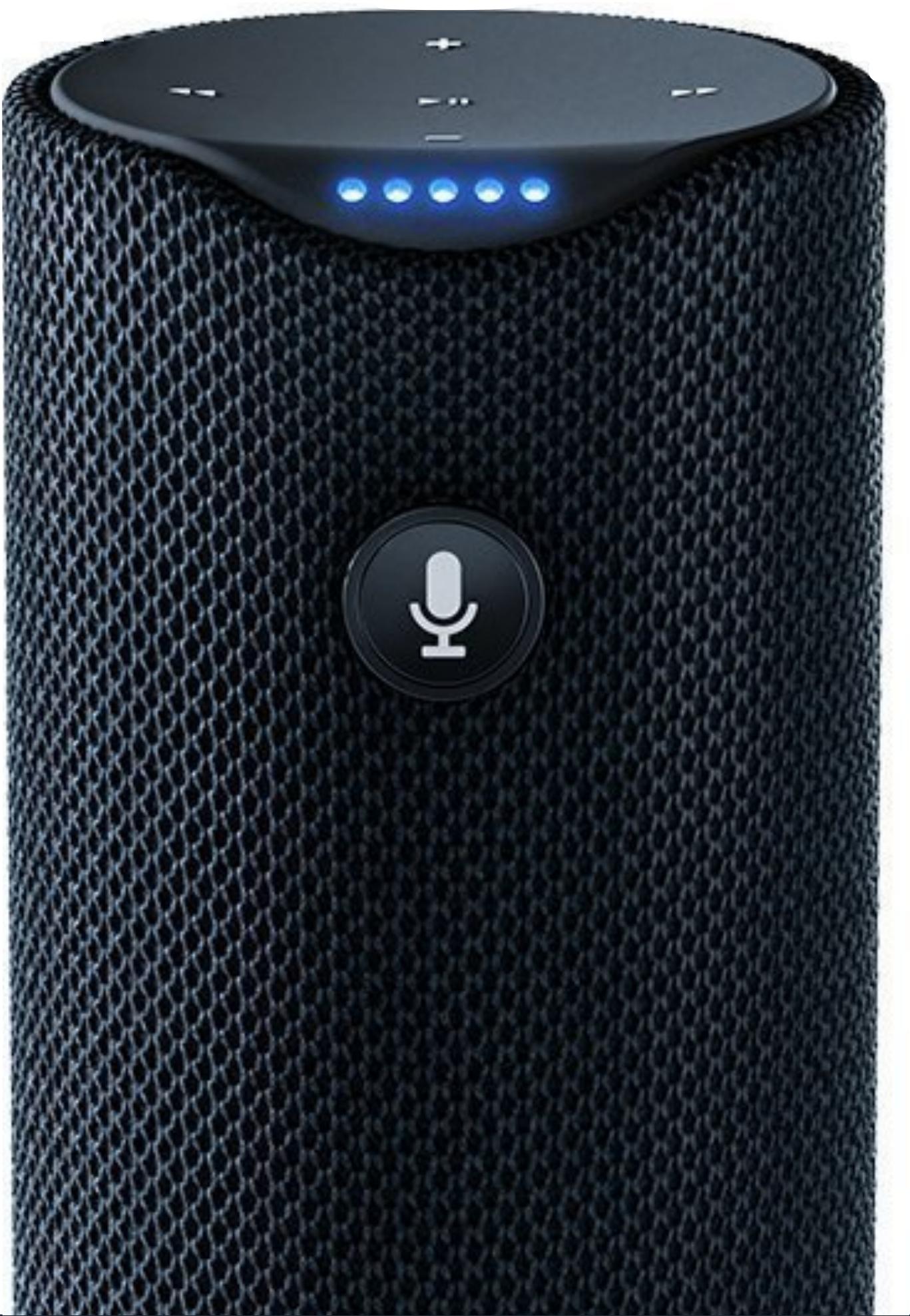
# Calypso

A single interface to manage all your WordPress.com or Jetpack-enabled sites, built with the latest web technologies and used by millions of people — and now it's open source.

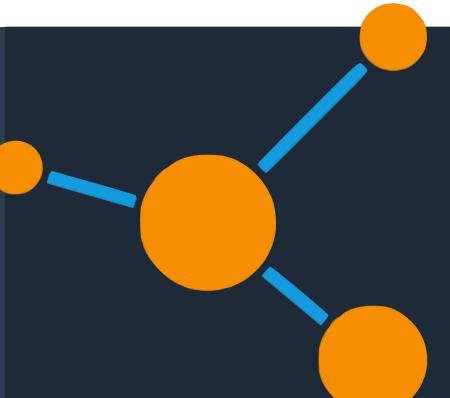
For OSX 10.8+, Windows 7+, and Linux.

The WordPress API

@nagmay



The WordPress API



@nagmay

**Alert notifications**

**Current status**  
PCC is open and observing normal operating hours. During an emergency important notifications will be posted here.

**Where is the PCC website?**  
You may have reached this page because the website is not currently available. Staff are working to correct the problem. Please try visiting the website again later.

**Report an emergency**  
Call: 911 or 971-722-4444  
To report non-emergency events, Public Safety can be reached at 971-722-4902.

**Distance Ed Alerts**  
PCC Distance Ed (@pccdistance) posted a message: FreeGeek and PCC Southeast Campus is doing a personal computer recycling event on Oct. 18th from 11am-2pm. [pcc.edu/about/events/...](http://pcc.edu/about/events/) Oct 4, 2017

**Shuttle Alerts**  
PCC Shuttle (@pcsshuttle) posted a message: Green line 1st run from cascade is cancelled. We apologize for the inconvenience. Resume 7:10 departure. Oct 6, 2017

**News**

**Posted in 2017**

- Caring for HT**  
Sylvania's four-decade-old building, home to high-demand health programs, is in need of major upgrades Posted October 2, 2017
- College Open Panther Pantries**  
Rock Creek Campus students are asking faculty and staff to contribute to their newly branded "Panther Pantry," formerly known at the Rock Creek Canteen Posted September 28, 2017
- Mitsui to Discuss PCC's Future**  
President to discuss fostering greater accessibility, equitable student success and the 2017 bond measure Posted September 27, 2017
- First Humboldt Market on Oct. 3**  
The Humboldt neighborhood will converge on the Cascade Campus for the first-ever Humboldt Market, featuring food, fresh produce, wares from local vendors, live music, and more Posted September 25, 2017
- The Community's Center**  
Portland Metro Workforce Training Center is a community gem, but buildings in need of polish Posted September 25, 2017
- Future Connect Gets Passing Grades**  
Comprehensive scholarship program for first-generation, low-income college students is a huge success Posted September 19, 2017
- Music Series Coming to Rock Creek**  
The concerts are presented in conjunction with the Tuesday evening "Music Appreciation" class and are free and open to the public Posted September 18, 2017
- Childcare for All**  
Without access to daycare services at PCC, Sylvia Koch said she'd have a hard time going to college Posted September 18, 2017
- Bernards Wins Math Award**  
Jessica Bernards has won a prestigious national honor given to people who teach mathematics at a community college Posted September 15, 2017
- DREAMer Center a go at PCC**  
Meyer Memorial Trust's \$50,000 grant will help launch the DREAM Center at Rock Creek Campus Posted September 14, 2017

**Everybody PANIC.**

**Portland Community College**

**Caring for HT**  
Sylvania's four-decade-old building, home to high-demand health programs, is in need of major upgrades

**Read the story >**

**Earn Your Degree or Certificate**  
to kick-start your career or four-year degree – apply now!

**Get Job Ready**  
in a year or less.

**Enrich Your Life**  
through non-credit Community Education classes.

**Develop Your Business or Career**  
at the CLIMB Center for Advancement.

**Prepare for College**  
with high school completion, GED, and English skill courses.

**Study in the USA**  
with intensive English classes and over 100 areas of study.

**College news and events**

- Register to vote**  
Election day is November 7. Oregonians have until October 17 to register to vote.
- News**
  - Caring for HT**  
Oct 2
  - Mitsui to Discuss PCC's Future**  
Sep 27
- Events**
  - Fireside chats**  
Cascade: October 9, Newberg Center: October 11, Southeast: October 12, CLIMB: October 12, PMWTC: October 13
  - SY - Job Fair**  
October 11

**Spottlights**

**OCTOBER 2017 CYBER SECURITY Awareness Month**

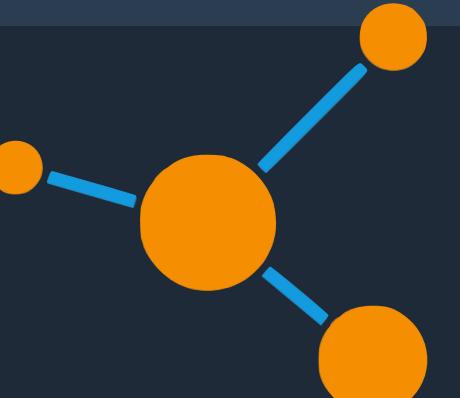
**Portland Community College Information Technology**

The WordPress API

@nagmay

# Thanks Everyone

## Questions?



@nagmay

# WPCAMPUS

WHERE WORDPRESS MEETS HIGHER EDUCATION

## WPCampus Online 2018 (*virtual conference*)

Tuesday, January 30, 2018

Our call for speakers is open until Tuesday, November 7, 2017.

Want to experience WordPress sessions, expertise, and networking with users across the world without the expense and hassle of travel? This one-day, session-filled event is for you!

Visit [online.wpcampus.org](http://online.wpcampus.org) for more information.

[wpcampus.org](http://wpcampus.org)

@wpcampusorg

#WPCampus

## WPCampus 2018 (*in-person conference*)

Summer 2018

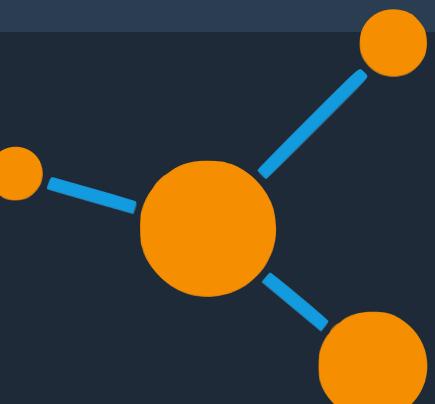
The hosting application will close Tuesday, October 31, 2017.

The main event for WPCampus is our annual in-person conference. We hold this event on college campuses and we'd love to bring our community to you.

Visit [wpcampus.org](http://wpcampus.org) to learn more about the event and how to host.

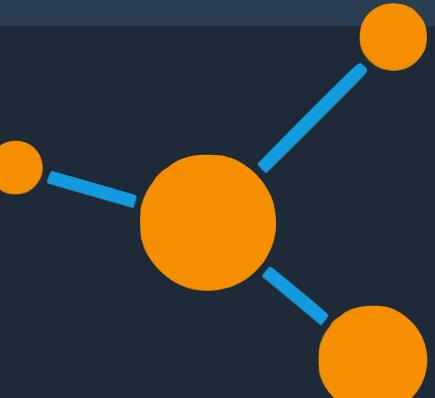
# Extra credit topics?

- Custom end points?
- Server side considerations
  - CURL / wget
  - Execution time / Caching?
- Inside WordPress?
- backbone.js and underscore.js
- The old way?
- AllGrain/ data.js
- blogs.pcc.edu/  
accessibility/  
placemarks-data



# Custom Endpoints

```
add_action( 'rest_api_init', function () {
    register_rest_route(
        'myplugin/v1',           // namespace
        '/stuff/(?P<id>\d+)',   // route
        array(                   // args (allowed methods, callback)
            'methods' => 'GET',
            'callback' => 'my_awesome_func',
        )
    );
});
```



# Custom Endpoints

```
function my_awesome_func( $data ) {
    // bad call
    if ( empty( $posts ) ) {
        return new WP_Error( 'no_stuff',
            'Stuff was not found',
            array( 'status' => 404 ) );
    }

    // get the data to return
    $posts = get_posts( array(
        'stff' => $data['id'],
    ) );
    return $posts[0]->post_title;
}
```

