The WordPress API

Web applications for the REST of us!

@nagmay

WPCAMPUS

WHERE WORDPRESS MEETS HIGHER EDUCATION

wpcampus.org
@wpcampusorg
#WPCampus

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 for more information.

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 to learn more about the event and how to host.

What the what?

- Overview and history
- Pulling data

Intermission: Let's talk about beer

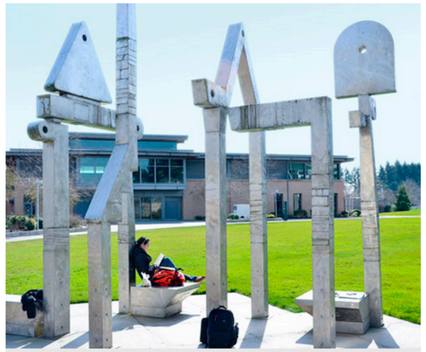
- Pushing data

Gabriel Nagmay
gabriel@nagmay.com

Web Analyst
Coffee Lover
Home Brewer

Cascade

A small campus atmosphere in a diverse urban neighborhood in the heart of Portland.



Rock Creek

A spacious, 260-acre campus in the high-tech corridor near Hillsboro.

Southeast

PCC's newest campus, in the growing and diverse central eastside of Portland.

Sylvania

PCC's largest campus, on a wooded hillside between Tigard and Lake Oswego.

Centers

Centers are smaller facilities offering job training, specialized programs, and transfer courses. These centers, and many other locations throughout the community, make up PCC's Extended Learning Campus.

- CLIMB Center for Advancement
- Downtown Center
- Newberg Center
- Swan Island Trades Center
- Columbia County
- Hillsboro Center
- Portland Metropolitan (PMWTC)
- Willow Creek Center

Application Protocol Interface

REpresentational
State
Transfer

HTTP Methods

POST .../posts	→	Create
GET .../posts	→	Retrieve
PUT .../posts	→	Update
DELETE .../posts	→	Delete

JavaScript

Coldfusion

Python

Ruby

PHP

Java

C#

Bash

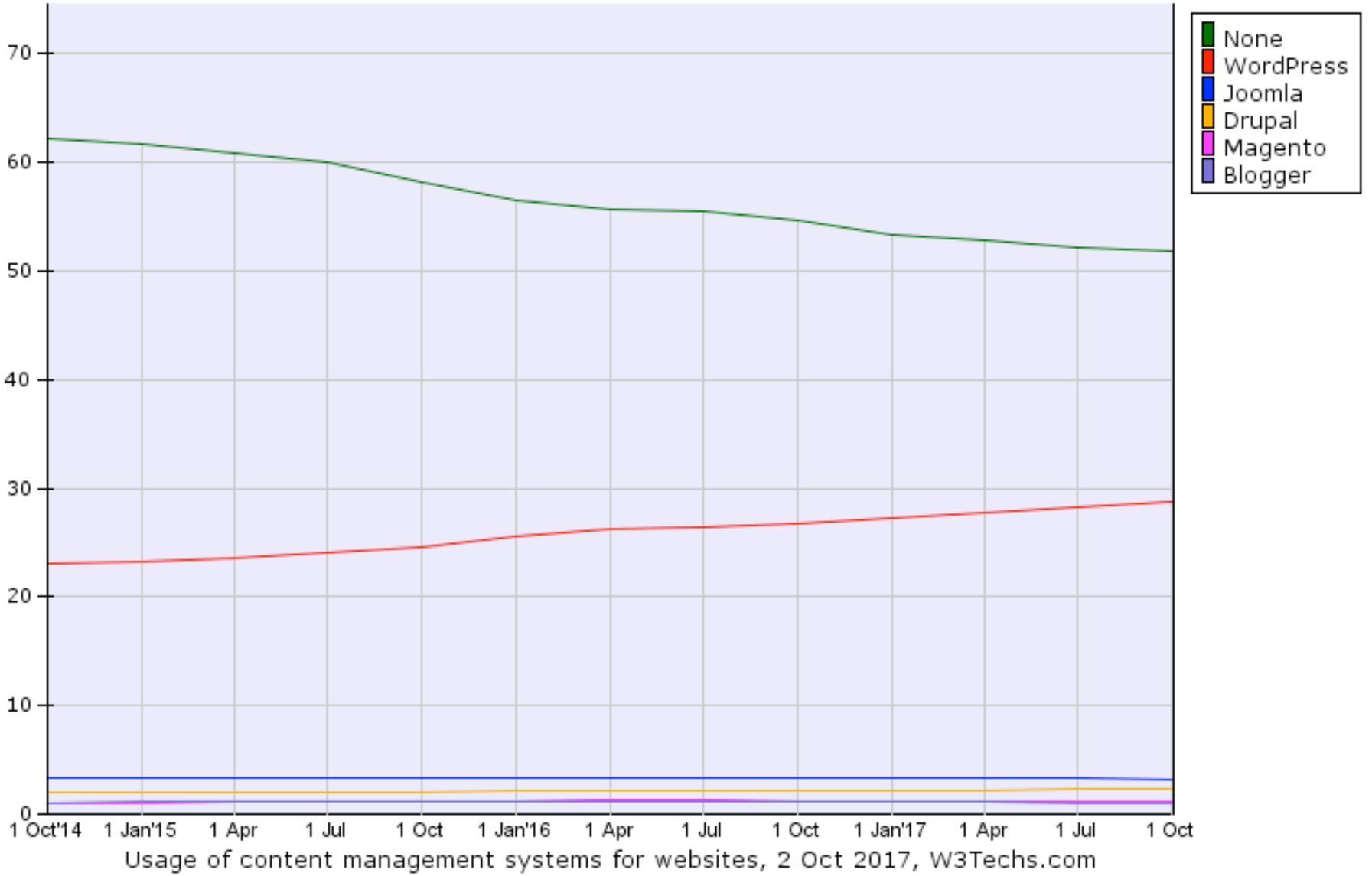
WP-API History

2013 GSoC project by Ryan McCue

2014 v1.0 released as WP-API plugin

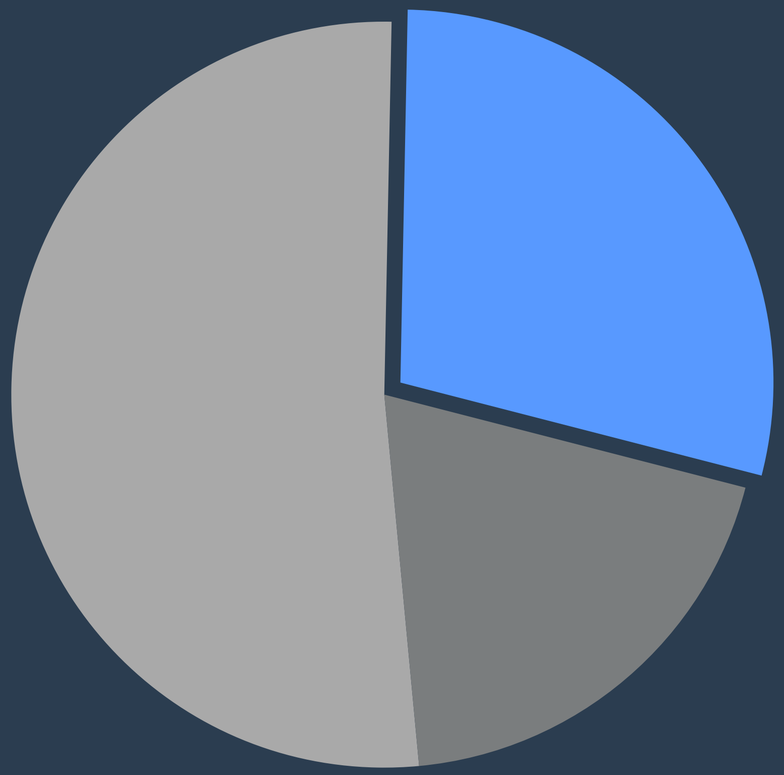
2015 v2.0 developed
Infrastructure added to WP4.4

2016 Endpoints added to WP4.7


None
WordPress
Joomla
Drupal
Magento
Blogger
70
60
50
40
30
20
10
0
1 Oct'14
1 Jan'15
1 Apr
1 Jul
1 Oct
1 Jan'16
1 Apr
1 Jul
1 Oct
1 Jan'17
1 Apr
1 Jul
1 Oct
Usage of content management systems for websites, 2 Oct 2017, W3Techs.com

28.7% of the web

Now available to you via an API



WordPress
Other CMS
None

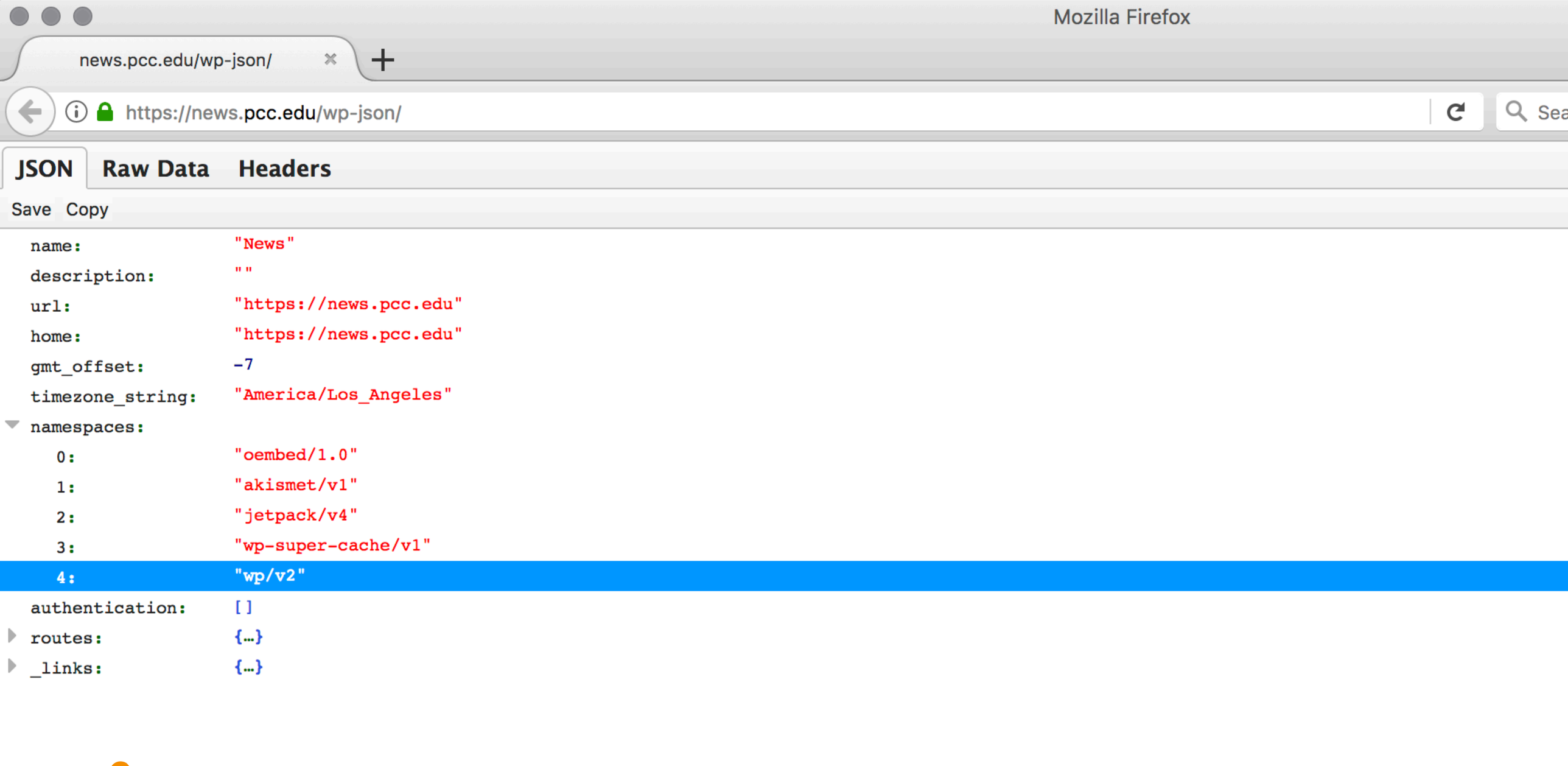
Tools

Examples: github.com/nagmay/WRK7

Web browser

Recommended

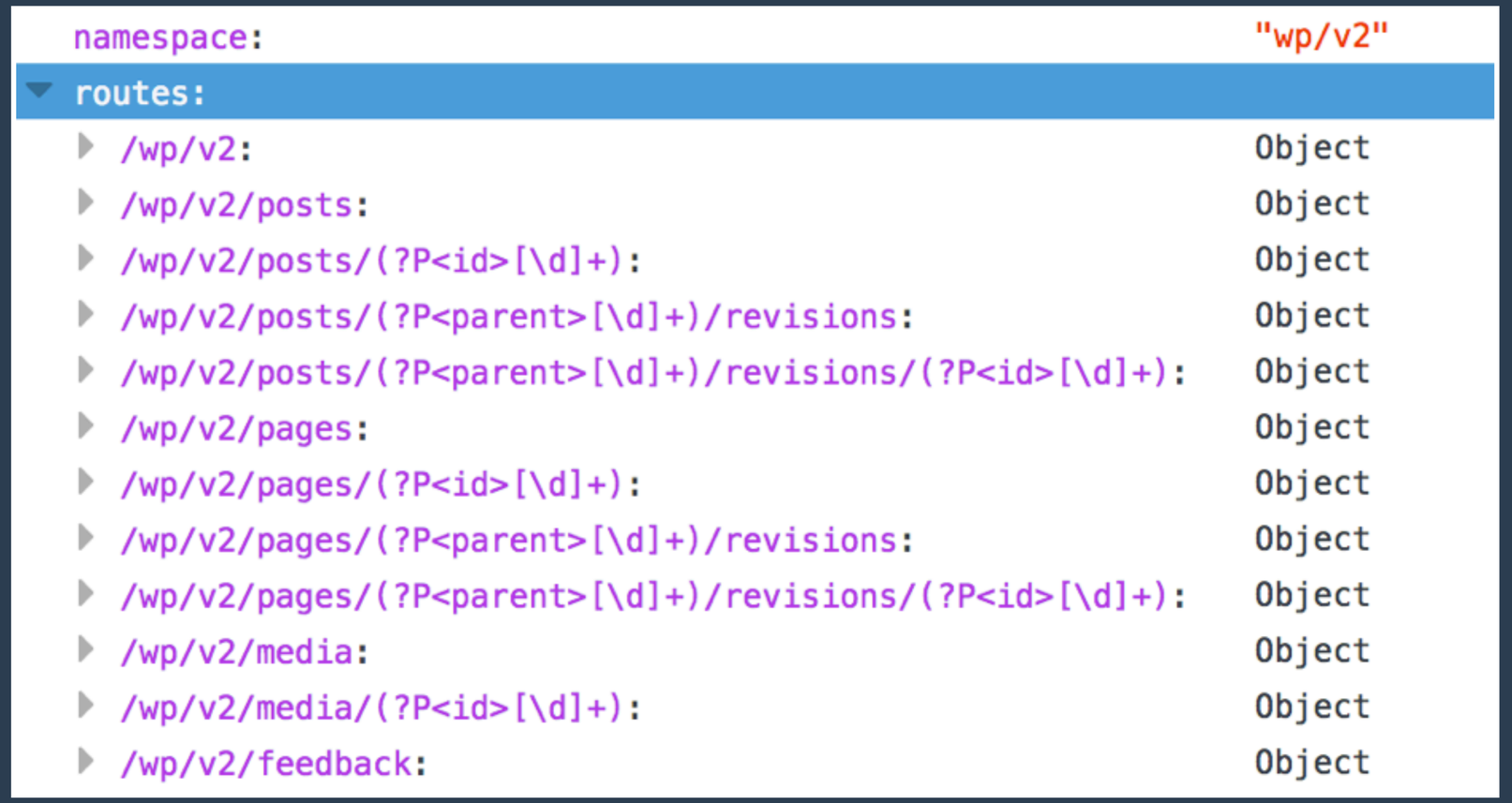
- WP REST API Handbook
- Postman: getpostman.com



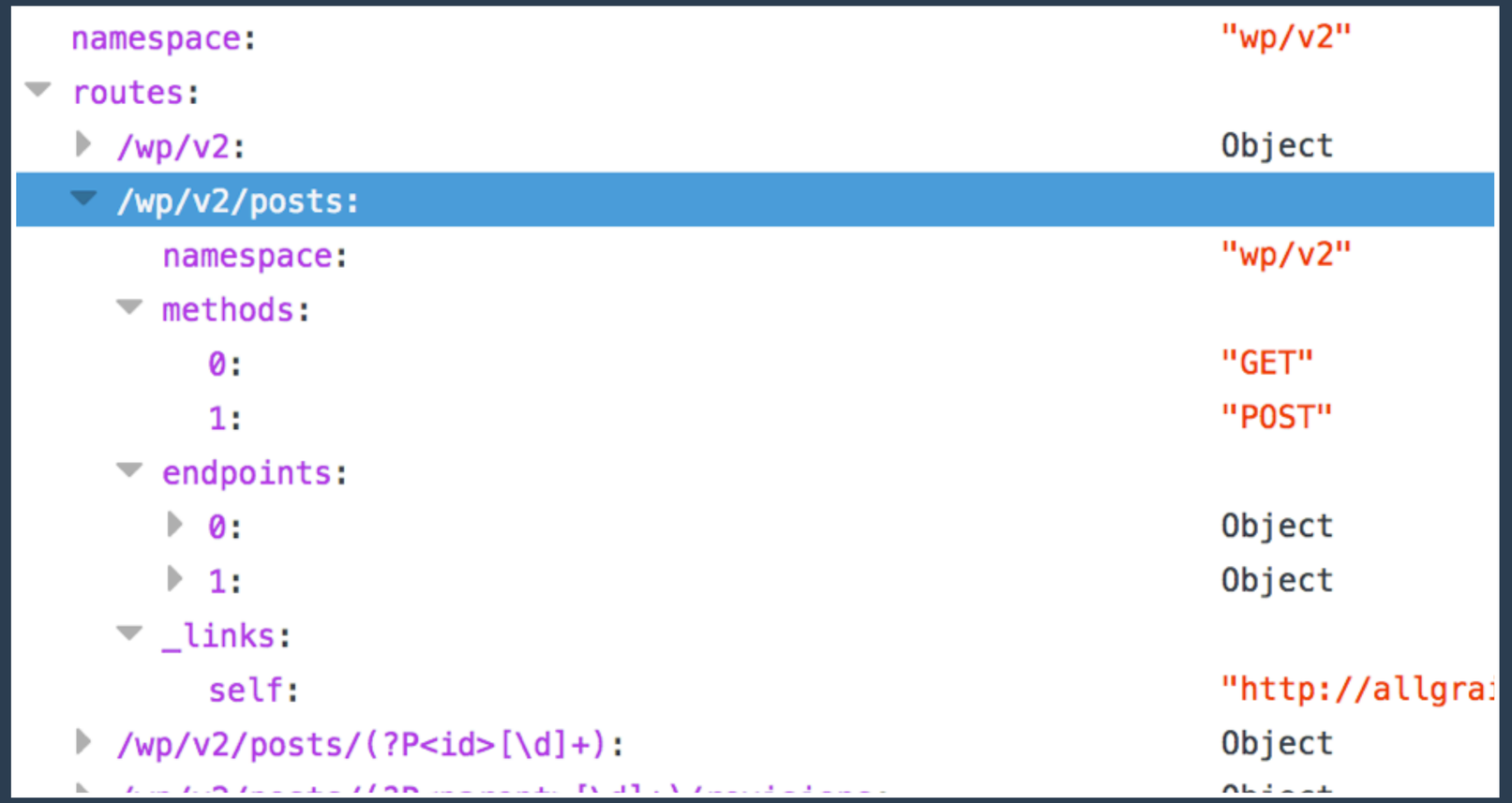
The WordPress API

@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: "http://allgra
/wp/v2/posts/(?P<id>[\d]+): 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

"WTF? Why did you choose JS for your examples"

- someone in this workshop?

WordPress.org

Search WordPress.org

Showcase Themes Plugins Mobile Support Get Involved About Blog Hosting

Download WordPress

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

POSTMAN

Developing APIs is hard
Postman makes it easy

The WordPress API

@nagmay

Pulling Data

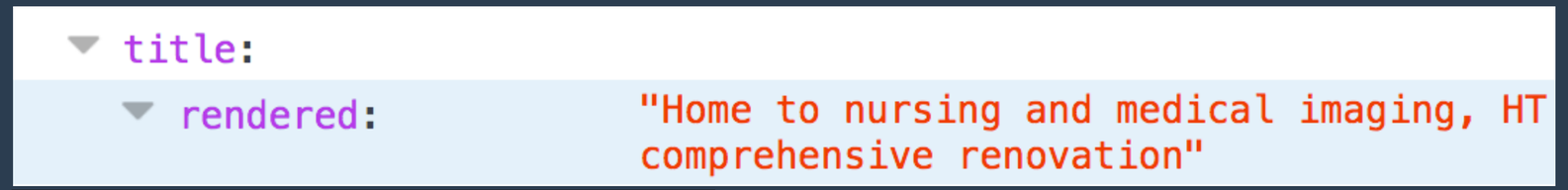
POST .../posts	→	Create
GET .../posts	→	Retrieve
PUT .../posts	→	Update
DELETE .../posts	→	Delete

01: GET JSON

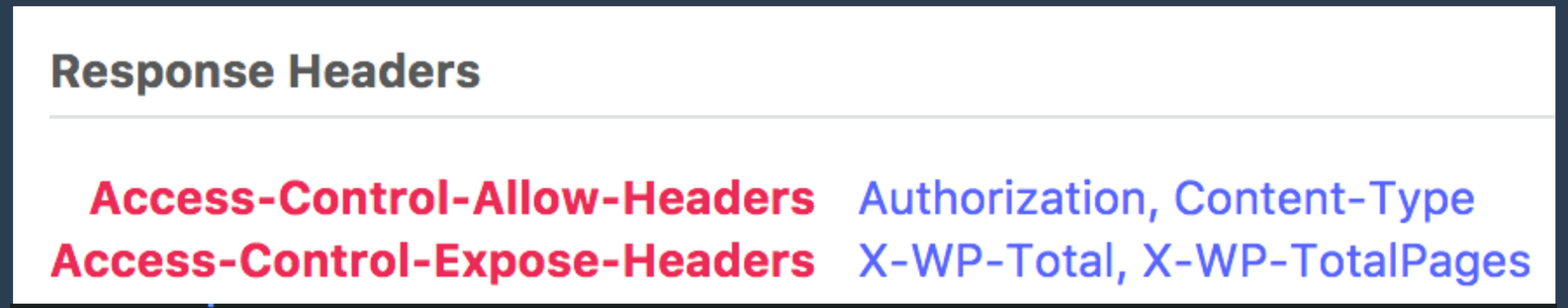
```
$.ajax({
    method: "GET",
    url: url,
    dataType: "json"
}).done(function( data ) {
    // do something
});
```

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

02: GET Posts



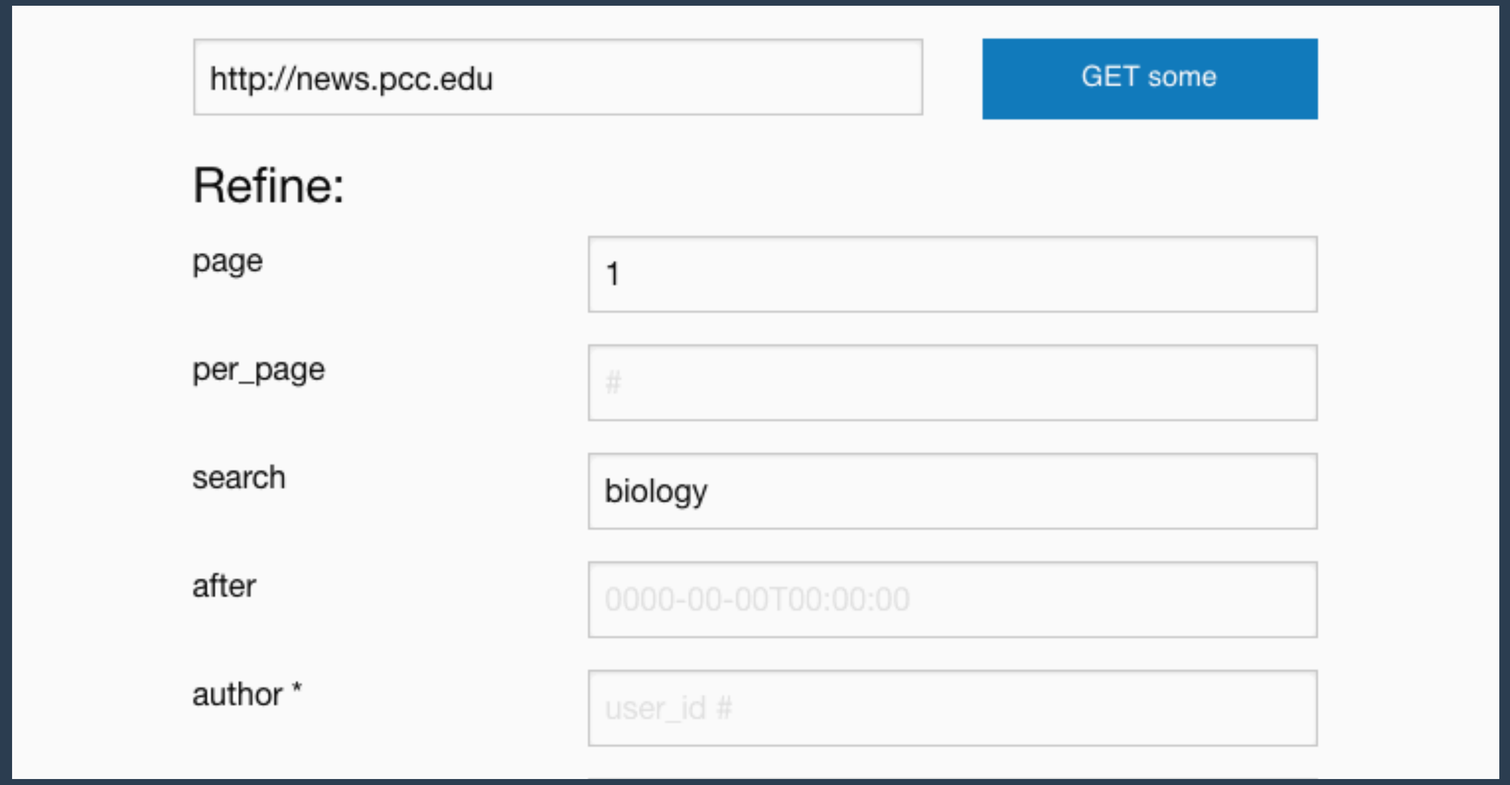

```
data[i].title.rendered
```




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

03: Posts Arguments

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



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 su
        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'; // Ge
$.getJSON(url, function (data, status, request) { // if succes

        // 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

```php
/* === 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
CITYSTEAM
NAUGHTY NURSE IPA

The WordPress API

@nagmay

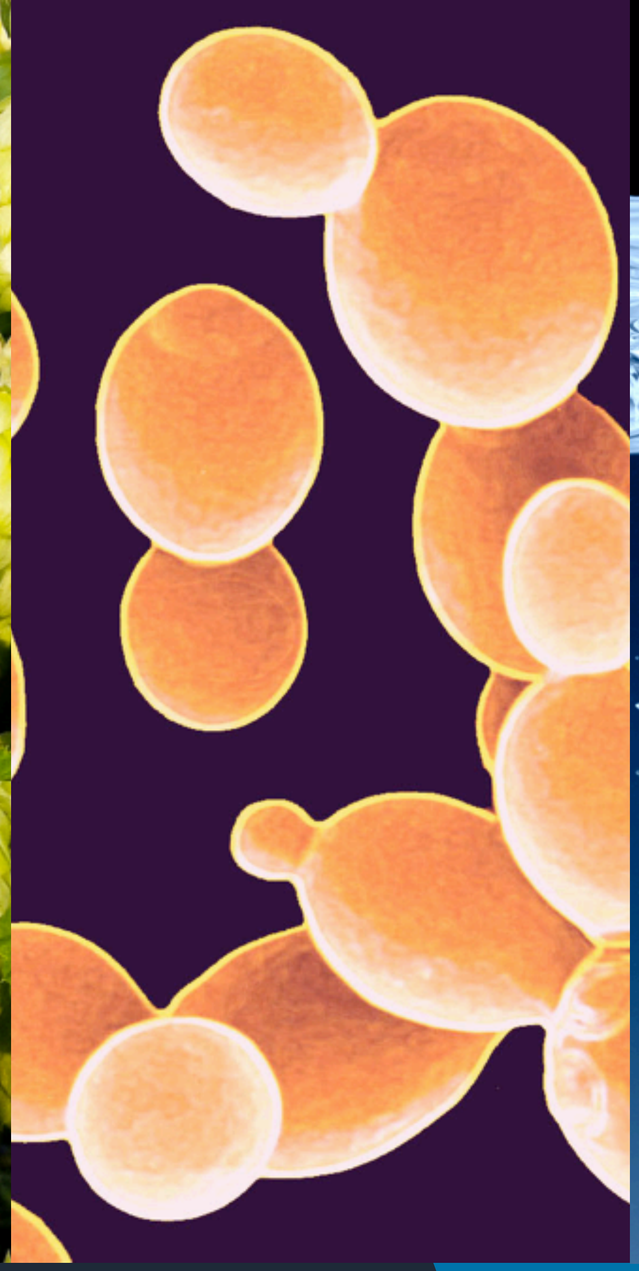
The WordPress API

@nagmay

The WordPress API

@nagmay





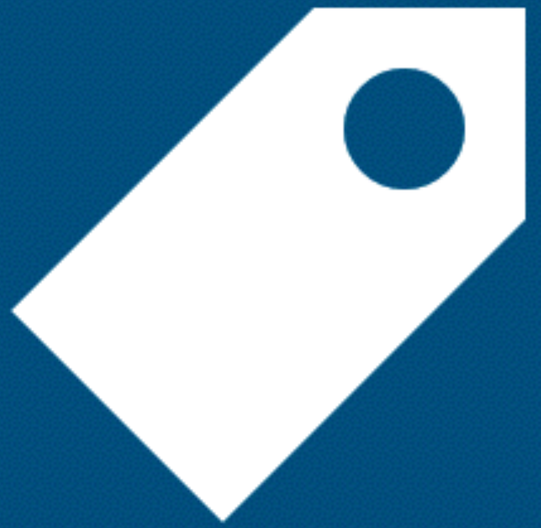

The WordPress API

@nagmay

Posts

Pages

Media



Tags

AllGrainBeer

HTM-Ale

10A: American Pale Ale

5 gal | 60 min | 70 %

ABV	6.24
IBU	39.3
OG	1.060
FG	1.014
SRM	9.5
B:G	0.7

Grain | Hops | Yeast | Water

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

+ 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

```php
/* ========================================================================
 *  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

Authentication

Cookies & Nonce

OAuth plugin

Basic Auth plugin

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 );
} );
```

OAuth: Server

WordPress REST API – OAuth 1.0a Server

Basic Auth plugin

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 );
    }
})
```

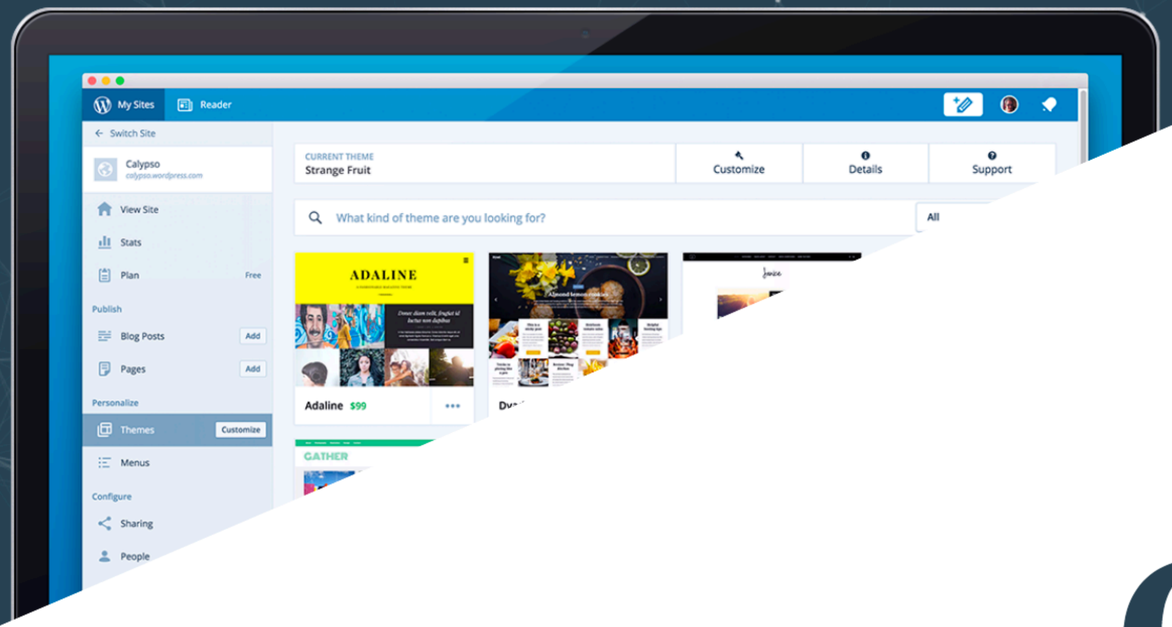
15: Users

```
// 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' 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?

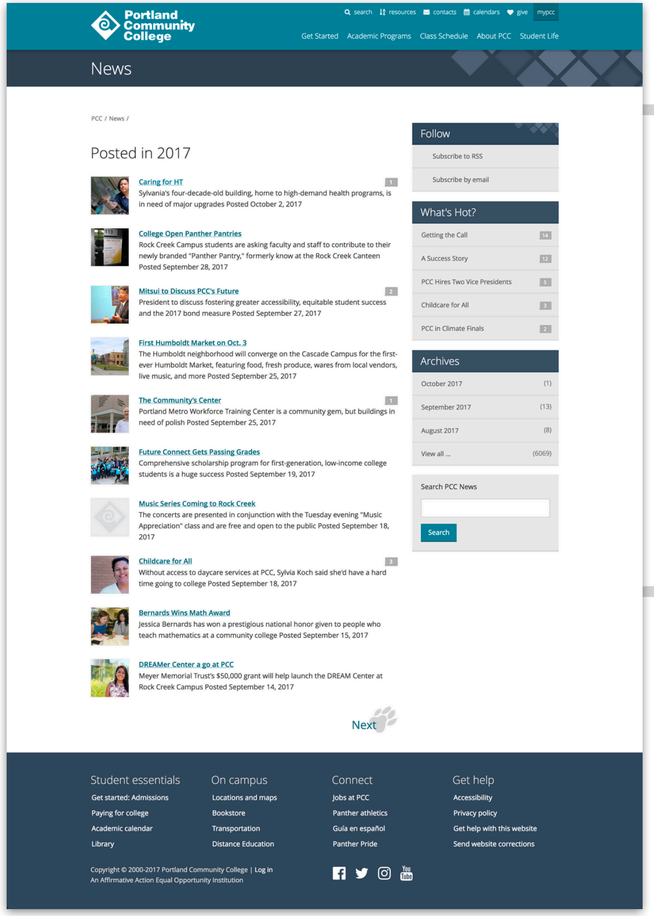


Introducing

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

Thanks Everyone

Questions?

@nagmay

WPCAMPUS

WHERE WORDPRESS MEETS HIGHER EDUCATION

wpcampus.org
@wpcampusorg
#WPCampus

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 for more information.

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 to learn more about the event and how to host.

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

```php
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;
}
```