**WP REST API and AJAX FORMS**

**Step-By-Step Course**

**Craig West**

**DESIRED OUTCOME:**

* To understand WP REST API.
* How to use FETCH API to get and handle JSON data.
* How to use to get data and to display data using JavaScript.
* How to create new endpoints and edit existing ones in the built in REST API for GET and POST requests.
* Create forms that use AJAX to WP REST API.
* Secure FORMS with WP NONCE.

**Final project** is a form that creates a new post using a custom WP REST API endpoint that we will create that validates data types and has a NONCE to provide security from Cross Site Request Forgery.

Live site **user: demo pwd: demo**

<https://49plus.co.uk/udemy/>

**DOWNLOAD WORKSHOP PACK**

Or <https://github.com/iwswordpress/rest>

The pack contains all the code necessary to replicate the final project, the slides for the talk and a resource list.

I have a number of videos that cover this workshop at:

<https://www.youtube.com/playlist?list=PLsszRSbzjyvka3TZUBr4Kfs10w9yry41c>

*Much inspiration was drawn from the following playlist on YouTube, particularly lessons 14 onwards:*

<https://www.youtube.com/watch?v=c_piVnQrJuY&list=PLT9miexWCpPU3TtDIVxA765dh2MaJY5X3>

*I highly recommend watching these tutorials.*

### **DISABLING ENDPOINTS**

<http://juha.blog/dev/wordpress/disable-wordpress-rest-api-endpoints-example-user-endpoint/>

### **How to disable user endpoint**

Easy way to solve this is to disable user endpoint (if you don’t need it in your application). This can be done using rest\_endpoints filter in your functions.php. Following filter will disable user endpoints. You can user same logic to any endpoint you want to close.

add\_filter( 'rest\_endpoints', function( $endpoints ){

if ( isset( $endpoints['/wp/v2/users'] ) ) {

unset( $endpoints['/wp/v2/users'] );

}

if ( isset( $endpoints['/wp/v2/users/(?P<id>[\d]+)'] ) ) {

unset( $endpoints['/wp/v2/users/(?P<id>[\d]+)'] );

}

return $endpoints;

});

### **How to disable entire REST API**

If you don’t need Rest API at all and you want to disable it for some reason you can use this snippet in your functions.php.

add\_filter('rest\_enabled', '\_return\_false');

add\_filter('rest\_jsonp\_enabled', '\_return\_false');

**SET UP**

**Step by step**

**PHP** has two folders:

1. mu-plugins
2. php

**mu-plugins** has files that contain functions which will be automatically loaded like plugins but cannot be turned on or off unless removed.

You can use the functions to make a plugin or insert into theme functions.php but if not in a child theme they can be overwritten on a theme update.

Site structure:

A screenshot of a cell phone

Description automatically generated

Copy mu-plugins into wp-content folder.

A screenshot of a cell phone

Description automatically generated

Copy page-xxxx.php files into root of your theme. A screenshot of a cell phone

Description automatically generated

**Create blank pages for the following:**

ajax, ajax-search, post, nonce and post-nonce

and add to menu if you wish…

The functions.php contains just code to generate a child theme called **child** from theme **generatepress.**

A screenshot of a computer screen

Description automatically generated

Test yourWPsite/ajax gives you the same page as me.

**Enable <post name> in Settings > Permalinks**

**server.php** is and include that has the site URL as a variable so that you can change to your server in just one place.

<?php

    //$SITE = "https://49plus.co.uk/udemy/";

    $SITE = site\_url().'/';

?>

If we have a file page-ajax.php and a blank page in WP ajax, then [www.site.com/ajax](http://www.site.com/ajax) will render that page.

**SQL** folder has one MySQL script for 01\_tblTest as sample data

**Chrome Extension - JSON Formatter**

It is useful to have a JSON formatter in the browser. Search for *Chrome Extensions JSON Formatter* and install.

<https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa?hl=en>

Or use <https://jsoneditoronline.org/> which we will need later anyway.

**What are REST/AJAX/JSON?**

PowerPoint slide->

Roy Fielding propose REST as a guiding practice in his thesis in 2000:

<https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>

By its nature, HTTP is very RESTful.

We will look at the wp-json object that is the REST API for WP.

Open <https://jsoneditoronline.org/>

In browser type <https://49plus.co.uk/udemy/wp-json/> and copy and paste this into JSON EDITOR.

We can now see what the REST API contains…

Let us look at the ‘hello world’ of the REST API:

<https://49plus.co.uk/udemy/wp-json/wp/v2/posts>

We can now see what the REST.

I will now go into detail…

* author
* authorName (added field)
* title.rendered

ACF fields needs ACF to REST plugin.

**FETCH API**

**Let’s look at fetch.html, fetch404.html and fetch500.html**

This show the nature of the fetch API and how data returned is initially a stream that needs to be converted to JSON (or TEXT, BLOB etc).

The use of **.then** and **.catch** is shown.

**ENDPOINTS**

**HTTP VERB (GET/POST/DELETE…) + URL = ENDPPOINT**

One URL can have two endpoints if it is used as GET and POST.

The URLs below can be two endpoints 1) GET 2) POST

A selection of useful endpoints, shaded ones most useful:

[**https://49plus.co.uk/udemy/wp-json/wp/v2/posts**](https://49plus.co.uk/udemy/wp-json/wp/v2/posts)

[**https://49plus.co.uk/udemy/wp-json/wp/v2/posts?\_fields=authorName,id,excerpt,title,link,acf**](https://49plus.co.uk/udemy/wp-json/wp/v2/posts?_fields=authorName,id,excerpt,title,link,acf%20) (underscore before fields)

https://49plus.co.uk/udemy/wp-json/wp/v2/posts?search=json

https://49plus.co.uk/udemy/wp-json/wp/v2/posts?order=asc

https://49plus.co.uk/udemy/wp-json/wp/v2/posts?\_fields[]=author&\_fields[]=id&\_fields[]=excerpt&\_fields[]=title&\_fields[]=link

https://49plus.co.uk/udemy/wp-json/wp/v2/categories

https://49plus.co.uk/udemy/wp-json/wp/v2/categories/24

https://49plus.co.uk/udemy/wp-json/wp/v2/posts/?per\_page=2

https://49plus.co.uk/udemy/wp-json/wp/v2/posts?categories=4

https://49plus.co.uk/udemy/wp-json/wp/v2/users

https://49plus.co.uk/udemy/wp-json/wp/v2/users/14

**We can have our own custom endpoints:**

[**https://49plus.co.uk/udemy/wp-json/wordcamp/v2/districts**](%20https:/49plus.co.uk/udemy/wp-json/wordcamp/v2/districts)

[**https://49plus.co.uk/udemy/wp-json/wordcamp/v2/latest-posts/4**](https://49plus.co.uk/udemy/wp-json/wordcamp/v2/latest-posts/4)

(latest posts in category id=4)

[**https://49plus.co.uk/udemy/wp-json/wordcamp/v2/totalusers**](https://49plus.co.uk/udemy/wp-json/wordcamp/v2/totalusers%20)

(total number of users)

<https://jsoneditoronline.org/> to format it.

Look in Dev Tools > Console and you can see the output of an array of posts.

These are in rest-custom.php in mu-plugins.

Let’s look at then now…

**CREATING ENDPOINTS IN WP REST API**

**Sample page of GET requests for POSTS and DATA:**

<https://49plus.co.uk/udemy/ajax/>

We will look at this file to see how to process returned JSON data and render to the page…

**SEARCH FORM:**

<https://49plus.co.uk/udemy/ajax-search/>

**Create a new post using a custom endpoint:**

<https://49plus.co.uk/udemy/post/>

**Examples of nonces**

<https://49plus.co.uk/udemy/nonce/>

**POST data to create a new post using nonces**

<https://49plus.co.uk/udemy/post-nonce/>