

PPOM API Documentation

Version 1.0

(beta)

Enable Rest API	3
Endpoint	3
Products - Operations	3
1- Get all ppom fields attached with product.	3
2- Set/Update PPOM Fields	4
3- Delete Fields	6
// Delete all fields against product	6
Order- Operations	8
1- Get all meta data information attached with Order	8
2- Set/Update Meta/Fields against order	9
Return: All meta fields attached to order	9
3- Delete Meta/Fields in Order	10
Return: All meta fields attached to order	10
3- Field Types Table	11
3.1- Options	11

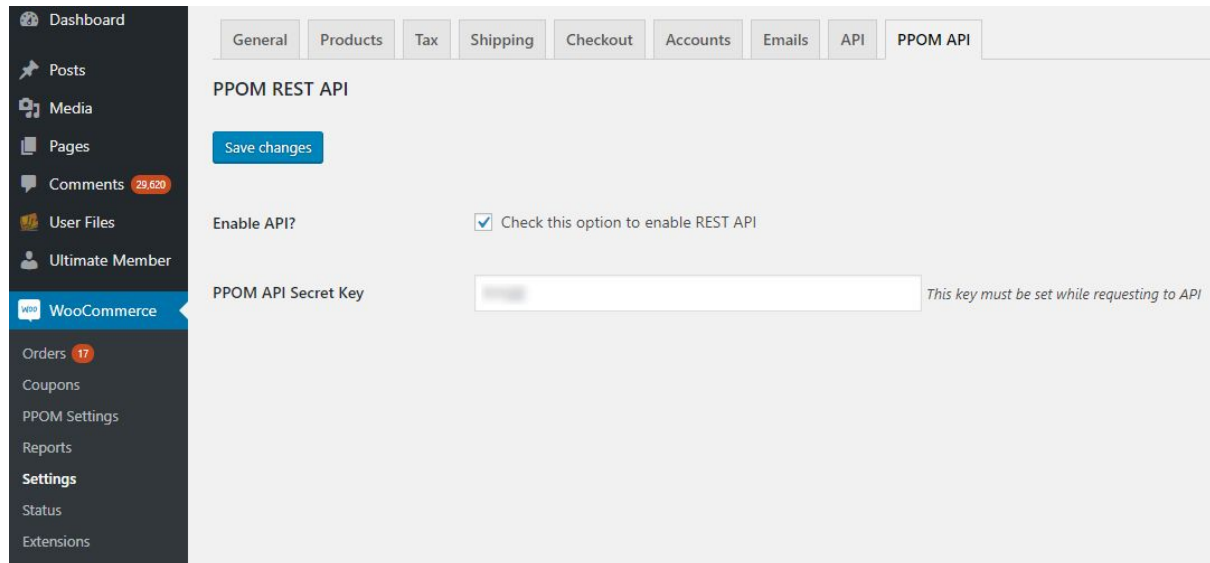


N-Media
We Extend Wordpress

[PPOM](#) version 11.8+ now have API support. It's a beta release. Following is information about API operations.

Enable Rest API

Make sure REST API is enabled from WooCommerce → Settings → PPOM API Tab.



Endpoint

Base Endpoint : `siteurl/wp-json/ppom/v1/{$operations}`

Products - Operations

1- Get all ppom fields attached with product.

1.1 Operation	get/product/?product_id=238&secret_key=11122		
1.2 Method	GET		
1.3 Parameters			
	Key	Type	Detail
	product_id	integer	mandatory
	secret_key	string	mandatory
1.4 Example Response	{		

	<pre> "status": "success", "message": "Meta found 121", "meta_id": 121, "product_id": 238, "ppom_fields": [{ "title": "My text", "type": "text", "data_name": "my_test", "description": "Just a text input", "required": 1 }, { "title": "My text 23", "type": "text", "data_name": "my_test3", "description": "Another text", "required": 1 }, { "title": "Type your email", "type": "email", "data_name": "your_email", "description": "Email please", "required": 1 }] </pre>
Other Status	success: if operation successful no_product: if product is not found no_meta: if no fields/meta attached with product

2- Set/Update PPOM Fields

If requested fields are exists, it will update otherwise add them to product. If no PPOM meta is attached to product then it will create a new meta group with same name as of Product title in WooCommerce and add these fields.

2.1 Operation	set/product/		
2.2 Method	POST		
2.3 Parameters			
	Key	Type	Detail
	product_id	integer	mandatory
	secret_key	string	mandatory
	fields	Array (see section 3.0) (json_encode)	mandatory
2.4 Fields Meta			
	Fields	See section 3-Fields Types	

	<table><tr><td>Options</td><td colspan="2">Extra options for Checkbox, Radio, Select</td></tr><tr><td></td><td colspan="2"><table><tr><td>Property</td><td>Type</td></tr><tr><td>options</td><td>Array (see section 2.5)</td></tr></table></td></tr></table>	Options	Extra options for Checkbox, Radio, Select			<table><tr><td>Property</td><td>Type</td></tr><tr><td>options</td><td>Array (see section 2.5)</td></tr></table>		Property	Type	options	Array (see section 2.5)
Options	Extra options for Checkbox, Radio, Select										
	<table><tr><td>Property</td><td>Type</td></tr><tr><td>options</td><td>Array (see section 2.5)</td></tr></table>		Property	Type	options	Array (see section 2.5)					
Property	Type										
options	Array (see section 2.5)										
2.5 Example Request	<pre>\$ppom_fields = array(array('title' => 'My text', 'type' => 'text', 'data_name' => 'my_test', 'description' => 'Just a text input', 'required' => 1), array('title' => 'Package', 'type' => 'checkbox', 'data_name' => 'package', 'description' => 'Select Package', 'options' => array(array('option'=>"Basic", 'price'=>''), array('option'=>"Student", 'price'=>'5.20'), array('option'=>"Super", 'price'=>'5%'),), 'required' => 1), array('title' => 'Type your email', 'type' => 'email', 'data_name' => 'your_email', 'description' => 'Email please', 'required' => 1),);</pre> <table><tr><td>Actual Post Request</td></tr><tr><td><pre>\$post_data = array('product_id' => 238, 'secret_key' => 123123, 'fields' => json_encode(\$ppom_fields),);</pre></td></tr></table>		Actual Post Request	<pre>\$post_data = array('product_id' => 238, 'secret_key' => 123123, 'fields' => json_encode(\$ppom_fields),);</pre>							
Actual Post Request											
<pre>\$post_data = array('product_id' => 238, 'secret_key' => 123123, 'fields' => json_encode(\$ppom_fields),);</pre>											
2.6 Example Response	<pre>{ "status": "success", "meta_id": 122, "product_id": 238, "fields": [{ "title": "My text", "type": "text", "data_name": "my_test", "description": "Just a text input", "required": 1 }]</pre>										

	<pre> }, { "title": "Package", "type": "checkbox", "data_name": "package", "description": "Select Package", "options": [{ "option": "Basic", "price": "" }, { "option": "Student", "price": "5.20" }, { "option": "Super", "price": "5%" }], "required": 1 }, { "title": "Type your email", "type": "email", "data_name": "your_email", "description": "Email please", "required": 1 }] }</pre>
--	--

3- Delete Fields

3.1 Operation	delete/product/														
3.2 Method	POST														
3.3 Parameters	<table><tr><th>Key</th><th>Type</th><th>Detail</th></tr><tr><td>product_id</td><td>integer</td><td>mandatory</td></tr><tr><td>secret_key</td><td>string</td><td>mandatory</td></tr><tr><td>fields</td><td>Array (only data_name)</td><td>mandatory</td></tr></table>			Key	Type	Detail	product_id	integer	mandatory	secret_key	string	mandatory	fields	Array (only data_name)	mandatory
Key	Type	Detail													
product_id	integer	mandatory													
secret_key	string	mandatory													
fields	Array (only data_name)	mandatory													
3.4 Example Request	<pre>\$delete_fields = array('field_1','field_2'); \$post_data = array('product_id' => 238, 'secret_key' => 123123, 'fields' => json_encode(\$delete_fields),); // Delete all fields against product \$delete_fields = array('__all_keys');</pre>														

3.5 Example Response

```
{
  "status": "success",
  "meta_id": "123",
  "product_id": 238,
  "fields": [{
    "title": "My text",
    "type": "text",
    "data_name": "my_test",
    "description": "Just a text input",
    "required": 1
  }, {
    "title": "Package",
    "type": "checkbox",
    "data_name": "package",
    "description": "Select Package",
    "options": [{
      "option": "Basic",
      "price": ""
    }, {
      "option": "Student",
      "price": "5.20"
    }, {
      "option": "Super",
      "price": "7%"
    }],
    "required": 0
  }]
}
```

Order- Operations

1- Get all meta data information attached with Order

1.1 Operation	get/order/?order_id=2141&secret_key=11122											
1.2 Method	GET											
1.3 Parameters	<table><tr><th>Key</th><th>Type</th><th>Detail</th></tr><tr><td>order_id</td><td>integer</td><td>mandatory</td></tr><tr><td>secret_key</td><td>string</td><td>mandatory</td></tr></table>			Key	Type	Detail	order_id	integer	mandatory	secret_key	string	mandatory
Key	Type	Detail										
order_id	integer	mandatory										
secret_key	string	mandatory										
1.4 Example Response	<pre>{ "status": "success", "message": "Order found 2141", "order_items_meta": [{ "product_id": 2028, "product_meta_data": [{ "id": 3932, "key": "thehidden", "value": "22" }, { "id": 3933, "key": "you_compaany", "value": "nmedia" }, { "id": 3934, "key": "about_compny", "value": "I love Again" }, { "id": 3935, "key": "email_input", "value": "najeeb.ahmd82@gmail.com" }, { "id": 3936, "key": "dob", "value": "28-2-2200" }, { "id": 3937, "key": "gender", "value": "Male" }, { "id": 3938, "key": "your_date_range", "value": "22" }, { "id": 3939, "key": "select_timzone", "value": "22" }, { "id": 3940, "key": "your_interest", "value": "22" }] }</pre>											

	<pre> }, { "id": 3941, "key": "chose_itww", "value": "extra,upsize" }, { "id": 3942, "key": "colors_palettes", "value": "22" }, { "id": 3943, "key": "select_image", "value": "22" }, { "id": 3951, "key": "your_age", "value": "22" }] } </pre>
Other Status	success: if operation successful no_order: if order is not found
Checkbox value	Checkbox values will be shown as separated by comma link see above under key: <code>chose_itww</code>

2- Set/Update Meta/Fields against order

If requested fields are exists, it will update otherwise add them to order.

Return: All meta fields attached to order

2.1 Operation	set/order/														
2.2 Method	POST														
2.3 Parameters	<table><tr><th>Key</th><th>Type</th><th>Detail</th></tr><tr><td>order_id</td><td>integer</td><td>mandatory</td></tr><tr><td>secret_key</td><td>string</td><td>mandatory</td></tr><tr><td>fields</td><td>Array (see section 2.4) (json_encode)</td><td>mandatory</td></tr></table>			Key	Type	Detail	order_id	integer	mandatory	secret_key	string	mandatory	fields	Array (see section 2.4) (json_encode)	mandatory
Key	Type	Detail													
order_id	integer	mandatory													
secret_key	string	mandatory													
fields	Array (see section 2.4) (json_encode)	mandatory													
2.4 Example Request	<pre>\$ppom_fields = array(2028 => array('about_compny'=>'I love Again', 'email_input'=>'najeeb.ahmd82@gmail.com', 'your_age'=>36),);</pre>														

	<pre>\$post_data = array('product_id' => 238, 'secret_key' => 123123, 'fields' => json_encode(\$ppom_fields),);</pre> <p>In above example \$ppom_fields has key as item id in order with array is given with data_name=>value pair.</p>
2.5 Example Response	Same as in 1.4 in Orders

3- Delete Meta/Fields in Order

Return: All meta fields attached to order

2.1 Operation	delete/order/														
2.2 Method	POST														
2.3 Parameters	<table><tr><th>Key</th><th>Type</th><th>Detail</th></tr><tr><td>order_id</td><td>integer</td><td>mandatory</td></tr><tr><td>secret_key</td><td>string</td><td>mandatory</td></tr><tr><td>fields</td><td>Array (see section 2.4) (json_encode)</td><td>mandatory</td></tr></table>			Key	Type	Detail	order_id	integer	mandatory	secret_key	string	mandatory	fields	Array (see section 2.4) (json_encode)	mandatory
Key	Type	Detail													
order_id	integer	mandatory													
secret_key	string	mandatory													
fields	Array (see section 2.4) (json_encode)	mandatory													
2.4 Example Request	<pre>\$delete_fields = array(2028 => array('your_email', 'select_image'),); \$post_data = array('product_id' => 238, 'secret_key' => 123123, 'fields' => json_encode(\$delete_fields),);</pre> <p>In above example \$delete_fields has key as item id in order with array is given with data_name=>value pair.</p>														
	Same as in 1.4 in Orders														

3- Field Types Table

Field	type	data_name	title	description	required	options
Text	text	unique	ANY	ANY	true:1,false: 0	N/A
Email	email	unique	ANY	ANY	true:1,false: 0	N/A
Date	date	unique	ANY	ANY	true:1,false: 0	N/A
Number	number	unique	ANY	ANY	true:1,false: 0	N/A
Textarea	textarea	unique	ANY	ANY	true:1,false: 0	N/A
Date Range	daterange	unique	ANY	ANY	true:1,false: 0	N/A
Timezone	timezone	unique	ANY	ANY	true:1,false: 0	N/A
Color Picker	color	unique	ANY	ANY	true:1,false: 0	N/A
Select	select	unique	ANY	ANY	true:1,false: 0	See section 3.1
Radio	radio	unique	ANY	ANY	true:1,false: 0	See section 3.1
Checkbox	checkbox	unique	ANY	ANY	true:1,false: 0	See section 3.1

3.1- Options

Options is an array with following keys:

```
'options' => array(  
    array('option'=>"Basic", 'price'=> ''),  
    array('option'=>"Student", 'price'=>'5.20'),  
    array('option'=>"Super", 'price'=>'5%'),  
);
```

Note: It's just a beta release. If you see any issue or want to improve this guide please feel free to contact us at sales@najeebmedia.com