

PPOM API Documentation

Version 2.1

(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 |

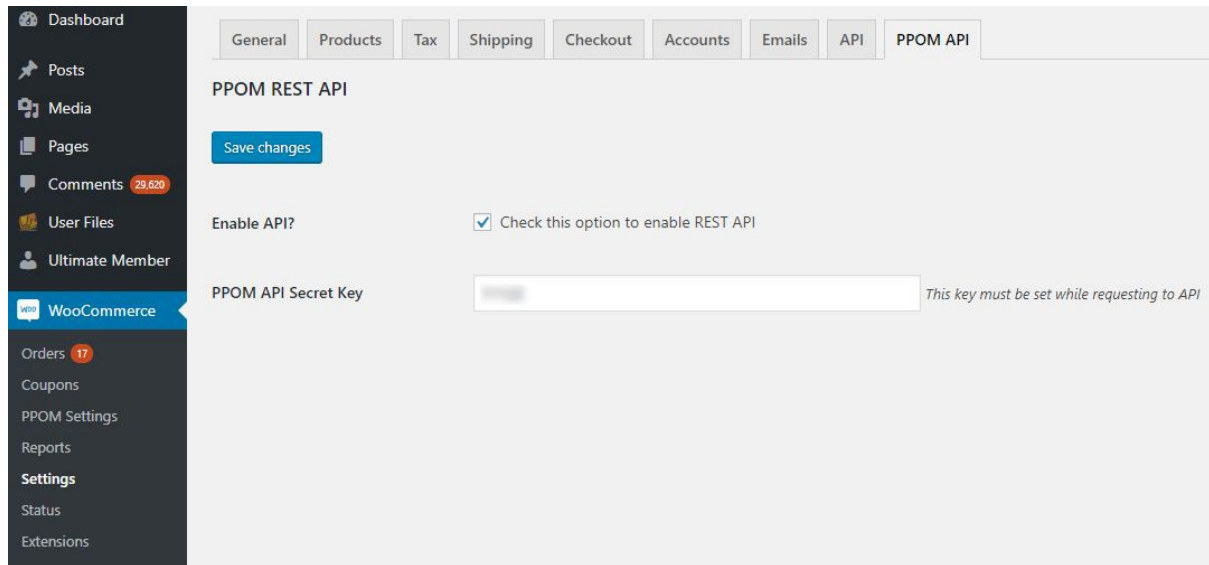


N-Media
We Extend Wordpress

[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 | <table border="1"> <thead> <tr> <th>Key</th> <th>Type</th> <th>Detail</th> </tr> </thead> <tbody> <tr> <td>product_id</td> <td>integer</td> <td>mandatory</td> </tr> <tr> <td>secret_key</td> <td>string</td> <td>mandatory</td> </tr> </tbody> </table> | | | Key | Type | Detail | product_id | integer | mandatory | secret_key | string | mandatory |
| 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 }, { "title": "Package", "type": "checkbox", "data_name": "package", "description": "Select Package", "required": "", "options": [{ "option": "Basic", "price": "", "id": "thebasic" }, { "option": "Student", "price": "5.20", "id": "student" }, { "option": "Super", "price": "7", "id": "super" }] }] </pre> |
| Other Status | <p>success: if operation successful no_product: if product is not found no_meta: if no fields/meta attached with product</p> |

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 | <table border="1"> <thead> <tr> <th>Key</th> <th>Type</th> <th>Detail</th> </tr> </thead> <tbody> <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 (see section 3.0) (json_encode)</td> <td>mandatory</td> </tr> </tbody> </table> | Key | Type | Detail | product_id | integer | mandatory | secret_key | string | mandatory | fields | Array (see section 3.0) (json_encode) | mandatory |
|---|---|---------------------|---|---------|--|---------|---|------------|--------|-----------|-------------------------|--|-----------|
| Key | Type | Detail | | | | | | | | | | | |
| product_id | integer | mandatory | | | | | | | | | | | |
| secret_key | string | mandatory | | | | | | | | | | | |
| fields | Array (see section 3.0) (json_encode) | mandatory | | | | | | | | | | | |
| 2.4 Fields Meta | <table border="1"> <tr> <td>Fields</td> <td>See section 3-Fields Types</td> </tr> <tr> <td>Options</td> <td>Extra options for Checkbox, Radio, Select</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Property</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>options</td> <td>Array (see section 2.5)</td> </tr> </tbody> </table> </td> </tr> </table> | Fields | See section 3-Fields Types | Options | Extra options for Checkbox, Radio, Select | | <table border="1"> <thead> <tr> <th>Property</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>options</td> <td>Array (see section 2.5)</td> </tr> </tbody> </table> | Property | Type | options | Array (see section 2.5) | | |
| Fields | See section 3-Fields Types | | | | | | | | | | | | |
| Options | Extra options for Checkbox, Radio, Select | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Property</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>options</td> <td>Array (see section 2.5)</td> </tr> </tbody> </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', array(array('option'=>"Basic",'price'=>'', 'id'=>'thebasic'), array('option'=>"Student",'price'=>'5.20', 'id'=>'student'), array('option'=>"Super",'price'=>'7', 'id'=>'super'),), 'required' => 1), array('title' => 'Type your email', 'type' => 'email', 'data_name' => 'your_email', 'description' => 'Email please', 'required' => 1),); </pre> <table border="1"> <tr> <td>Actual Post Request</td> </tr> <tr> <td> <pre> \$post_data = array('product_id' => 238, 'secret_key' => 123123, </pre> </td> </tr> </table> | Actual Post Request | <pre> \$post_data = array('product_id' => 238, 'secret_key' => 123123, </pre> | | | | | | | | | | |
| Actual Post Request | | | | | | | | | | | | | |
| <pre> \$post_data = array('product_id' => 238, 'secret_key' => 123123, </pre> | | | | | | | | | | | | | |

| | |
|-----------------------------|--|
| | <pre> '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 }, { "title": "Package", "type": "checkbox", "data_name": "package", "description": "Select Package", "options": [{ "option": "Basic", "price": "", "id": "thebasic" }, { "option": "Student", "price": "5.20", "id": "student" }, { "option": "Super", "price": "5%", "id": "super" }], "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 | |

| | 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 | <pre>{ "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", "required": "", "options": [{ "option": "Basic", "price": "", "id": "thebasic" }, { "option": "Student", "price": "5.20", "id": "student" }, { "option": "Super", "price": "7", "id": "super" }] }] }</pre> | | |

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 border="1"><thead><tr><th>Key</th><th>Type</th><th>Detail</th></tr></thead><tbody><tr><td>order_id</td><td>integer</td><td>mandatory</td></tr><tr><td>secret_key</td><td>string</td><td>mandatory</td></tr></tbody></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": 4442, "key": "package", "value": "[{"option":"Basic2","price":"","id":"basic2"},{"opt ion":"Studend","price":"5.20","id":"studend"}]" }, { "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/Radio/Select value | Checkbox/radio/select type input values will be shown as encode array (string output) in which each option will have following keys: option price id |

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 border="1"> <thead> <tr> <th>Key</th> <th>Type</th> <th>Detail</th> </tr> </thead> <tbody> <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> </tbody> </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>// checkbox,radio or select will be sent as collection of arrays like below: \$cb_options[] = array('id'=>'studen_id','option'=>'Students Data','price'=>'7.2'); \$cb_options[] = array('id'=>'subject','option'=>'Social Studies'); \$ppom_fields = array(2028 => array('about_compny'=>'I love Again', 'email_input'=>'najeeb.ahmd82@gmail.com', 'your_age'=>36, 'package'=> \$cb_options,),); \$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 | <p>Same as in 1.4 in Orders</p> |
| | |

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 border="1"> <thead> <tr> <th data-bbox="544 1659 831 1727">Key</th> <th data-bbox="831 1659 1118 1727">Type</th> <th data-bbox="1118 1659 1402 1727">Detail</th> </tr> </thead> <tbody> <tr> <td data-bbox="544 1727 831 1783">order_id</td> <td data-bbox="831 1727 1118 1783">integer</td> <td data-bbox="1118 1727 1402 1783">mandatory</td> </tr> <tr> <td data-bbox="544 1783 831 1839">secret_key</td> <td data-bbox="831 1783 1118 1839">string</td> <td data-bbox="1118 1783 1402 1839">mandatory</td> </tr> <tr> <td data-bbox="544 1839 831 1951">fields</td> <td data-bbox="831 1839 1118 1951">Array (see section 2.4) (json_encode)</td> <td data-bbox="1118 1839 1402 1951">mandatory</td> </tr> </tbody> </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