□□

□□

- □□□□ ___
  - Payment Details Properties
  - □□□□□ ____
    - Products Meta Items Properties
  - Shipping Lines Properties
  - Tax Lines Properties
  - Fee Lines Properites
  - Coupon Lines Properties
- □□□□ ___
  - HTTP □□ __
    - /wc-api/v2/orders
- □□□□ ___
  - HTTP □□ __
    - /wc-api/v2/orders/<id>
- □□□□□ ____
  - HTTP □□ __
    - /wc-api/v2/orders
  - Available Filters
- □□□□ ___
  - HTTP □□ __
    - /wc-api/v2/orders/<id>
- Delete An Order
  - HTTP Request
    - /wc-api/v2/orders/<id>
  - Parameters
- □□□□□ ____
  - HTTP □□ __
    - /wc-api/v2/orders/count
  - Available Filters

- [□□□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/statuses
- [□□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/notes
  - Order Notes Properties
- [□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/notes/<note_id>
- View List Of Notes From An Order
  - HTTP Request
    - /wc-api/v2/orders/<id>/notes
- [□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/notes/<note_id>
- Delete An Order Note
  - HTTP Request
    - /wc-api/v2/orders/<id>/notes/<note_id>
- [□□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/refunds
  - Order Refunds Properties
- [□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/refunds/<refund_id>
- [□□□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/refunds
- [□□□□□□]
  - HTTP □□
    - /wc-api/v2/orders/<id>/refunds/<refund_id>
- Delete An Order Refund
  - HTTP Request

- /wc-api/v2/orders/<id>/refunds/<refund_id>

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API□

□□□□

Attribute	Type	Description
`id`	integer	Order ID (post ID) *read-only*
`order_number`	integer	Order number *read-only*
`created_at`	string	UTC DateTime when the order was created *read-only*
`updated_at`	string	UTC DateTime when the order was last updated *read-only*
`completed_at`	string	UTC DateTime when the order was last completed *read-only*
`status`	string	Order status. By default are available the status: `pending`, `processing`, `on-hold`, `completed`, `cancelled`, `refunded` and `failed`. See View List of Order Statuses
`currency`	string	Currency in ISO format, e.g `USD`
`total`	string	Order total *read-only*
`subtotal`	string	Order subtotal *read-only*
`total_line_items_quantity`	integer	Total of order items *read-only*
`total_tax`	string	Order tax total *read-only*
`total_shipping`	string	Order shipping total *read-only*
`cart_tax`	string	Order cart tax *read-only*
`shipping_tax`	string	Order shipping tax *read-only*
`total_discount`	string	Order total discount *read-only*
`shipping_methods`	string	Text list of the shipping methods used in the order *read-only*
`payment_details`	array	List of payment details. See Payment Details Properties
`billing_address`	array	List of customer billing address. See Customer Billing Address Properties

Attribute	Type	Description
`shipping_address`	array	List of customer shipping address. See Customer Shipping Address Properties
`note`	string	Customer order notes
`customer_ip`	string	Customer IP address *read-only*
`customer_user_agent`	string	Customer User-Agent *read-only*
`customer_id`	integer	Customer ID (user ID) *required*
`view_order_url`	string	URL to view the order in frontend *read-only*
`line_items`	array	List of order line items. See Line Items Properties
`shipping_lines`	array	List of shipping line items. See Shipping Lines Properties
`tax_lines`	array	List of tax line items. See Tax Lines Properties *read-only*
`fee_lines`	array	List of fee line items. See Fee Lines Properites
`coupon_lines`	array	List of cupon line items. See Coupon Lines Properties
`customer`	array	Customer data. See Customer Properties

Payment Details Properties

Attribute	Type	Description
`method_id`	string	Payment method ID *required*
`method_title`	string	Payment method title *required*
`paid`	boolean	Shows/define if the order is paid using this payment method. Use `true` to complate the payment.
`transaction_id`	string	Transaction ID, an optional field to set the transacion ID when complate one payment (to set this you need set the `paid` as `true` too)

Attribute	Type	Description
`id`	integer	Line item ID *read-only*
`subtotal`	string	Line item subtotal
`subtotal_tax`	string	Line item tax subtotal
`total`	string	Line item total
`total_tax`	string	Line item tax total
`price`	string	Product price *read-only*
`quantity`	integer	Quantity
`tax_class`	string	Product tax class *read-only*
`name`	string	Product name *read-only*
`product_id`	integer	Product ID *required*
`sku`	string	Product SKU *read-only*
`meta`	array	List of product meta items. See Products Meta Items Properties
`variations`	array	List of product variation attributes. e.g: `"variation": {"pa_color": "Black", "pa_size": "XGG"}` (Use `pa_` prefix when is a product attribute) *write-only*

Products Meta Items Properties

Attribute	Type	Description
`key`	string	Meta item key
`label`	string	Meta item label
`value`	string	Meta item value

Shipping Lines Properties

Attribute	Type	Description
`id`	integer	Shipping line ID *read-only*
`method_id`	string	Shipping method ID *required*
`method_title`	string	Shipping method title *required*

Attribute	Type	Description
`total`	string	Total amount

Tax Lines Properties

Attribute	Type	Description
`id`	integer	Tax rate line ID *read-only*
`rate_id`	integer	Tax rate ID *read-only*
`code`	string	Tax rate code *read-only*
`title`	string	Tax rate title/name *read-only*
`total`	string	Tax rate total *read-only*
`compound`	boolean	Shows if is or not a compound rate. Compound tax rates are applied on top of other tax rates. *read-only*

Fee Lines Properites

Attribute	Type	Description
`id`	integer	Fee line ID *read-only*
`title`	string	Shipping method title *required*
`taxable`	boolean	Shows/define if the fee is taxable *write-only*
`tax_class`	string	Tax class, requered in write-mode if the fee is taxable
`total`	string	Total amount
`total_tax`	string	Tax total

Coupon Lines Properties

Attribute	Type	Description
`id`	integer	Coupon line ID *read-only*
`code`	string	Coupon code *required*
`amount`	string	Total amount *required*

□□□□

□ API □□□□□□□□□□

HTTP □□

POST

/wc-api/v2/orders

> “ □□□□□□□□□□□□

```
curl -X POST https://example.com/wc-api/v2/orders \
□-u consumer_key:consumer_secret \
□-H "Content-Type: application/json" \
□-d '{
  "order": {
    "payment_details": {
      "method_id": "bacs",
      "method_title": "Direct Bank Transfer",
      "paid": true
    },
    "billing_address": {
      "first_name": "John",
      "last_name": "Doe",
      "address_1": "969 Market",
      "address_2": "",
      "city": "San Francisco",
      "state": "CA",
      "postcode": "94103",
      "country": "US",
      "email": "john.doe@example.com",
      "phone": "(555) 555-5555"
    },
    "shipping_address": {
      "first_name": "John",
      "last_name": "Doe",
      "address_1": "969 Market",
```

```
      "address_2": "",
      "city": "San Francisco",
      "state": "CA",
      "postcode": "94103",
      "country": "US"
    },
    "customer_id": 2,
    "line_items": [
      {
        "product_id": 546,
        "quantity": 2
      },
      {
        "product_id": 613,
        "quantity": 1,
        "variations": {
          "pa_color": "Black"
        }
      }
    ],
    "shipping_lines": [
      {
        "method_id": "flat_rate",
        "method_title": "Flat Rate",
        "total": "10.00"
      }
    ]
  }
}'
```

```
var data = {
  order: {
    payment_details: {
      method_id: 'bacs',
      method_title: 'Direct Bank Transfer',
      paid: true
    },
    billing_address: {
      first_name: 'John',
      last_name: 'Doe',
```

```
    address_1: '969 Market',
    address_2: '',
    city: 'San Francisco',
    state: 'CA',
    postcode: '94103',
    country: 'US',
    email: 'john.doe@example.com',
    phone: '(555) 555-5555'
  },
  shipping_address: {
    first_name: 'John',
    last_name: 'Doe',
    address_1: '969 Market',
    address_2: '',
    city: 'San Francisco',
    state: 'CA',
    postcode: '94103',
    country: 'US'
  },
  customer_id: 2,
  line_items: [
    {
      product_id: 546,
      quantity: 2
    },
    {
      product_id: 613,
      quantity: 1,
      variations: {
        pa_color: 'Black'
      }
    }
  ],
  shipping_lines: [
    {
      method_id: 'flat_rate',
      method_title: 'Flat Rate',
      total: '10.00'
    }
  ]
```

```
  }
};

WooCommerce.post('orders', data, function(err, data, res) {
  console.log(res);
});
```

```
data = {
    "order": {
        "payment_details": {
            "method_id": "bacs",
            "method_title": "Direct Bank Transfer",
            "paid": True
        },
        "billing_address": {
            "first_name": "John",
            "last_name": "Doe",
            "address_1": "969 Market",
            "address_2": "",
            "city": "San Francisco",
            "state": "CA",
            "postcode": "94103",
            "country": "US",
            "email": "john.doe@example.com",
            "phone": "(555) 555-5555"
        },
        "shipping_address": {
            "first_name": "John",
            "last_name": "Doe",
            "address_1": "969 Market",
            "address_2": "",
            "city": "San Francisco",
            "state": "CA",
            "postcode": "94103",
            "country": "US"
        },
        "customer_id": 2,
        "line_items": [
            {
                "product_id": 546,
```

```
                "quantity": 2
            },
            {
                "product_id": 613,
                "quantity": 1,
                "variations": {
                    "pa_color": "Black"
                }
            }
        ],
        "shipping_lines": [
            {
                "method_id": "flat_rate",
                "method_title": "Flat Rate",
                "total": "10.00"
            }
        ]
    }
}

print(wcapi.post("orders", data).json())
```

```
<?php
$data = array(
    'order' => array(
        'payment_details' => array(
            'method_id' => 'bacs',
            'method_title' => 'Direct Bank Transfer',
            'paid' => true
        ),
        'billing_address' => array(
            'first_name' => 'John',
            'last_name' => 'Doe',
            'address_1' => '969 Market',
            'address_2' => '',
            'city' => 'San Francisco',
            'state' => 'CA',
            'postcode' => '94103',
            'country' => 'US',
            'email' => 'john.doe@example.com',
```

```
            'phone' => '(555) 555-5555'
        ),
        'shipping_address' => array(
            'first_name' => 'John',
            'last_name' => 'Doe',
            'address_1' => '969 Market',
            'address_2' => '',
            'city' => 'San Francisco',
            'state' => 'CA',
            'postcode' => '94103',
            'country' => 'US'
        ),
        'customer_id' => 2,
        'line_items' => array(
            array(
                'product_id' => 546,
                'quantity' => 2
            ),
            array(
                'product_id' => 613,
                'quantity' => 1,
                'variations' => array(
                    'pa_color' => 'Black'
                )
            )
        ),
        'shipping_lines' => array(
            array(
                'method_id' => 'flat_rate',
                'method_title' => 'Flat Rate',
                'total' => '10.00'
            )
        )
    )
);
```

print_r($woocommerce->orders->create($data)); ?>

```ruby
```

```
data = {
  order: {
    payment_details: {
      method_id: "bacs",
      method_title: "Direct Bank Transfer",
      paid: true
    },
    billing_address: {
      first_name: "John",
      last_name: "Doe",
      address_1: "969 Market",
      address_2: "",
      city: "San Francisco",
      state: "CA",
      postcode: "94103",
      country: "US",
      email: "john.doe@example.com",
      phone: "(555) 555-5555"
    },
    shipping_address: {
      first_name: "John",
      last_name: "Doe",
      address_1: "969 Market",
      address_2: "",
      city: "San Francisco",
      state: "CA",
      postcode: "94103",
      country: "US"
    },
    customer_id: 2,
    line_items: [
        {
          product_id: 546,
          quantity: 2
        },
        {
          product_id: 613,
          quantity: 1,
          variations: {
            pa_color: "Black"
```

```
          }
        }
    ],
    shipping_lines: [
        {
          method_id: "flat_rate",
          method_title: "Flat Rate",
          total: "10.00"
        }
    ]
  }
}

woocommerce.post("orders", data).parsed_response
```

> JSON □□□□□

```
{
  "order": {
    "id": 645,
    "order_number": 645,
    "created_at": "2015-01-26T20:00:21Z",
    "updated_at": "2015-01-26T20:00:21Z",
    "completed_at": "2015-01-26T20:00:21Z",
    "status": "processing",
    "currency": "USD",
    "total": "79.87",
    "subtotal": "63.97",
    "total_line_items_quantity": 3,
    "total_tax": "5.90",
    "total_shipping": "10.00",
    "cart_tax": "5.40",
    "shipping_tax": "0.50",
    "total_discount": "0.00",
    "shipping_methods": "Flat Rate",
    "payment_details": {
      "method_id": "bacs",
      "method_title": "Direct Bank Transfer",
```

```
    "paid": true
  },
  "billing_address": {
    "first_name": "John",
    "last_name": "Doe",
    "company": "",
    "address_1": "969 Market",
    "address_2": "",
    "city": "San Francisco",
    "state": "CA",
    "postcode": "94103",
    "country": "US",
    "email": "john.doe@example.com",
    "phone": "(555) 555-5555"
  },
  "shipping_address": {
    "first_name": "John",
    "last_name": "Doe",
    "company": "",
    "address_1": "969 Market",
    "address_2": "",
    "city": "San Francisco",
    "state": "CA",
    "postcode": "94103",
    "country": "US"
  },
  "note": "",
  "customer_ip": "127.0.0.1",
  "customer_user_agent": "WordPress/4.1; http://example.com",
  "customer_id": 2,
  "view_order_url": "https://example.com/my-account/view-order/645",
  "line_items": [
    {
      "id": 504,
      "subtotal": "43.98",
      "subtotal_tax": "4.40",
      "total": "43.98",
      "total_tax": "4.40",
      "price": "21.99",
```

```
      "quantity": 2,
      "tax_class": "reduced-rate",
      "name": "Premium Quality",
      "product_id": 546,
      "sku": "",
      "meta": []
    },
    {
      "id": 505,
      "subtotal": "19.99",
      "subtotal_tax": "1.00",
      "total": "19.99",
      "total_tax": "1.00",
      "price": "19.99",
      "quantity": 1,
      "tax_class": null,
      "name": "Ship Your Idea",
      "product_id": 613,
      "sku": "",
      "meta": [
        {
          "key": "pa_color",
          "label": "Color",
          "value": "Black"
        }
      ]
    }
  ],
  "shipping_lines": [
    {
      "id": 506,
      "method_id": "flat_rate",
      "method_title": "Flat Rate",
      "total": "10.00"
    }
  ],
  "tax_lines": [
    {
      "id": 507,
```

```
        "rate_id": "5",
        "code": "US-CA-TAX-1",
        "title": "Tax",
        "total": "4.40",
        "compound": false
      },
      {
        "id": 508,
        "rate_id": "4",
        "code": "US-STANDARD-1",
        "title": "Standard",
        "total": "1.50",
        "compound": false
      }
    ],
    "fee_lines": [],
    "coupon_lines": [],
    "customer": {
      "id": 2,
      "created_at": "2014-11-19T18:34:19Z",
      "email": "john.doe@example.com",
      "first_name": "",
      "last_name": "",
      "username": "john.doe",
      "last_order_id": "645",
      "last_order_date": "2015-01-26T20:00:21Z",
      "orders_count": 2,
      "total_spent": "19.00",
      "avatar_url":
"https://secure.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=96",
      "billing_address": {
        "first_name": "John",
        "last_name": "Doe",
        "company": "",
        "address_1": "969 Market",
        "address_2": "",
        "city": "San Francisco",
        "state": "CA",
        "postcode": "94103",
```

```
        "country": "US",
        "email": "john.doe@example.com",
        "phone": "(555) 555-5555"
      },
      "shipping_address": {
        "first_name": "John",
        "last_name": "Doe",
        "company": "",
        "address_1": "969 Market",
        "address_2": "",
        "city": "San Francisco",
        "state": "CA",
        "postcode": "94103",
        "country": "US"
      }
    }
  }
}
```

□□□□

□ API □□□□□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders/<id>

```
curl https://example.com/wc-api/v2/orders/645 \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/645', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/645").json())
```

```
<?php print_r($woocommerce->orders->get(645)); ?>
```

```
woocommerce.get("orders/645").parsed_response
```

> JSON □□□□


```
{
  "order": {
    "id": 645,
    "order_number": 645,
    "created_at": "2015-01-26T20:00:21Z",
    "updated_at": "2015-01-26T20:00:21Z",
    "completed_at": "2015-01-26T20:00:21Z",
    "status": "processing",
    "currency": "USD",
    "total": "79.87",
    "subtotal": "63.97",
    "total_line_items_quantity": 3,
    "total_tax": "5.90",
    "total_shipping": "10.00",
    "cart_tax": "5.40",
    "shipping_tax": "0.50",
    "total_discount": "0.00",
    "shipping_methods": "Flat Rate",
    "payment_details": {
      "method_id": "bacs",
      "method_title": "Direct Bank Transfer",
      "paid": true
    },
    "billing_address": {
      "first_name": "John",
      "last_name": "Doe",
      "company": "",
      "address_1": "969 Market",
      "address_2": "",
      "city": "San Francisco",
      "state": "CA",
      "postcode": "94103",
      "country": "US",
```

```
    "email": "john.doe@example.com",
    "phone": "(555) 555-5555"
  },
  "shipping_address": {
    "first_name": "John",
    "last_name": "Doe",
    "company": "",
    "address_1": "969 Market",
    "address_2": "",
    "city": "San Francisco",
    "state": "CA",
    "postcode": "94103",
    "country": "US"
  },
  "note": "",
  "customer_ip": "127.0.0.1",
  "customer_user_agent": "WordPress/4.1; http://example.com",
  "customer_id": 2,
  "view_order_url": "https://example.com/my-account/view-order/645",
  "line_items": [
    {
      "id": 504,
      "subtotal": "43.98",
      "subtotal_tax": "4.40",
      "total": "43.98",
      "total_tax": "4.40",
      "price": "21.99",
      "quantity": 2,
      "tax_class": "reduced-rate",
      "name": "Premium Quality",
      "product_id": 546,
      "sku": "",
      "meta": []
    },
    {
      "id": 505,
      "subtotal": "19.99",
      "subtotal_tax": "1.00",
      "total": "19.99",
```

```
      "total_tax": "1.00",
      "price": "19.99",
      "quantity": 1,
      "tax_class": null,
      "name": "Ship Your Idea",
      "product_id": 613,
      "sku": "",
      "meta": [
        {
          "key": "pa_color",
          "label": "Color",
          "value": "Black"
        }
      ]
    }
  ],
  "shipping_lines": [
    {
      "id": 506,
      "method_id": "flat_rate",
      "method_title": "Flat Rate",
      "total": "10.00"
    }
  ],
  "tax_lines": [
    {
      "id": 507,
      "rate_id": "5",
      "code": "US-CA-TAX-1",
      "title": "Tax",
      "total": "4.40",
      "compound": false
    },
    {
      "id": 508,
      "rate_id": "4",
      "code": "US-STANDARD-1",
      "title": "Standard",
      "total": "1.50",
```

```
        "compound": false
      }
    ],
    "fee_lines": [],
    "coupon_lines": [],
    "customer": {
      "id": 2,
      "created_at": "2014-11-19T18:34:19Z",
      "email": "john.doe@example.com",
      "first_name": "",
      "last_name": "",
      "username": "john.doe",
      "last_order_id": "645",
      "last_order_date": "2015-01-26T20:00:21Z",
      "orders_count": 2,
      "total_spent": "19.00",
      "avatar_url":
"https://secure.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=96",
      "billing_address": {
        "first_name": "John",
        "last_name": "Doe",
        "company": "",
        "address_1": "969 Market",
        "address_2": "",
        "city": "San Francisco",
        "state": "CA",
        "postcode": "94103",
        "country": "US",
        "email": "john.doe@example.com",
        "phone": "(555) 555-5555"
      },
      "shipping_address": {
        "first_name": "John",
        "last_name": "Doe",
        "company": "",
        "address_1": "969 Market",
        "address_2": "",
        "city": "San Francisco",
        "state": "CA",
```

```
        "postcode": "94103",
        "country": "US"
      }
    }
  }
}
```

□□□□□

□ API □□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders

```
curl https://example.com/wc-api/v2/orders \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders").json())
```

```
<?php print_r($woocommerce->orders->get()); ?>
```

```
woocommerce.get("orders").parsed_response
```

> JSON □□□□

```
{
  "orders": [
    {
      "id": 645,
      "order_number": 645,
```

```
"created_at": "2015-01-26T20:00:21Z",
"updated_at": "2015-01-26T20:00:21Z",
"completed_at": "2015-01-26T20:00:21Z",
"status": "processing",
"currency": "USD",
"total": "79.87",
"subtotal": "63.97",
"total_line_items_quantity": 3,
"total_tax": "5.90",
"total_shipping": "10.00",
"cart_tax": "5.40",
"shipping_tax": "0.50",
"total_discount": "0.00",
"shipping_methods": "Flat Rate",
"payment_details": {
  "method_id": "bacs",
  "method_title": "Direct Bank Transfer",
  "paid": true
},
"billing_address": {
  "first_name": "John",
  "last_name": "Doe",
  "company": "",
  "address_1": "969 Market",
  "address_2": "",
  "city": "San Francisco",
  "state": "CA",
  "postcode": "94103",
  "country": "US",
  "email": "john.doe@example.com",
  "phone": "(555) 555-5555"
},
"shipping_address": {
  "first_name": "John",
  "last_name": "Doe",
  "company": "",
  "address_1": "969 Market",
  "address_2": "",
  "city": "San Francisco",
  "state": "CA",
```

```
    "postcode": "94103",
    "country": "US"
  },
  "note": "",
  "customer_ip": "127.0.0.1",
  "customer_user_agent": "WordPress/4.1; http://example.com",
  "customer_id": 2,
  "view_order_url": "https://example.com/my-account/view-order/645",
  "line_items": [
    {
      "id": 504,
      "subtotal": "43.98",
      "subtotal_tax": "4.40",
      "total": "43.98",
      "total_tax": "4.40",
      "price": "21.99",
      "quantity": 2,
      "tax_class": "reduced-rate",
      "name": "Premium Quality",
      "product_id": 546,
      "sku": "",
      "meta": []
    },
    {
      "id": 505,
      "subtotal": "19.99",
      "subtotal_tax": "1.00",
      "total": "19.99",
      "total_tax": "1.00",
      "price": "19.99",
      "quantity": 1,
      "tax_class": null,
      "name": "Ship Your Idea",
      "product_id": 613,
      "sku": "",
      "meta": [
        {
          "key": "pa_color",
          "label": "Color",
          "value": "Black"
```

```
      }
    ]
  }
],
"shipping_lines": [
  {
    "id": 506,
    "method_id": "flat_rate",
    "method_title": "Flat Rate",
    "total": "10.00"
  }
],
"tax_lines": [
  {
    "id": 507,
    "rate_id": "5",
    "code": "US-CA-TAX-1",
    "title": "Tax",
    "total": "4.40",
    "compound": false
  },
  {
    "id": 508,
    "rate_id": "4",
    "code": "US-STANDARD-1",
    "title": "Standard",
    "total": "1.50",
    "compound": false
  }
],
"fee_lines": [],
"coupon_lines": [],
"customer": {
  "id": 2,
  "created_at": "2014-11-19T18:34:19Z",
  "email": "john.doe@example.com",
  "first_name": "",
  "last_name": "",
  "username": "john.doe",
  "last_order_id": "645",
```

```
        "last_order_date": "2015-01-26T20:00:21Z",
        "orders_count": 2,
        "total_spent": "19.00",
        "avatar_url":
"https://secure.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=96",
        "billing_address": {
          "first_name": "John",
          "last_name": "Doe",
          "company": "",
          "address_1": "969 Market",
          "address_2": "",
          "city": "San Francisco",
          "state": "CA",
          "postcode": "94103",
          "country": "US",
          "email": "john.doe@example.com",
          "phone": "(555) 555-5555"
        },
        "shipping_address": {
          "first_name": "John",
          "last_name": "Doe",
          "company": "",
          "address_1": "969 Market",
          "address_2": "",
          "city": "San Francisco",
          "state": "CA",
          "postcode": "94103",
          "country": "US"
        }
      }
    },
    {
      "id": 644,
      "order_number": 644,
      "created_at": "2015-01-26T19:33:42Z",
      "updated_at": "2015-01-26T19:33:42Z",
      "completed_at": "2015-01-26T19:33:42Z",
      "status": "on-hold",
      "currency": "USD",
      "total": "44.14",
```

```
"subtotal": "30.99",
"total_line_items_quantity": 2,
"total_tax": "3.15",
"total_shipping": "10.00",
"cart_tax": "2.65",
"shipping_tax": "0.50",
"total_discount": "0.00",
"shipping_methods": "Flat Rate",
"payment_details": {
  "method_id": "bacs",
  "method_title": "Direct Bank Transfer",
  "paid": false
},
"billing_address": {
  "first_name": "John",
  "last_name": "Doe",
  "company": "",
  "address_1": "969 Market",
  "address_2": "",
  "city": "San Francisco",
  "state": "CA",
  "postcode": "94103",
  "country": "US",
  "email": "john.doe@example.com",
  "phone": "(555) 555-5555"
},
"shipping_address": {
  "first_name": "John",
  "last_name": "Doe",
  "company": "",
  "address_1": "969 Market",
  "address_2": "",
  "city": "San Francisco",
  "state": "CA",
  "postcode": "94103",
  "country": "US"
},
"note": "",
"customer_ip": "127.0.0.1",
"customer_user_agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
```

```
Gecko) Chrome/40.0.2214.91 Safari/537.36",
      "customer_id": 2,
      "view_order_url": "https://example.com/my-account/view-order/644",
      "line_items": [
        {
          "id": 499,
          "subtotal": "21.99",
          "subtotal_tax": "2.20",
          "total": "21.99",
          "total_tax": "2.20",
          "price": "21.99",
          "quantity": 1,
          "tax_class": "reduced-rate",
          "name": "Premium Quality",
          "product_id": 546,
          "sku": "",
          "meta": []
        },
        {
          "id": 500,
          "subtotal": "9.00",
          "subtotal_tax": "0.45",
          "total": "9.00",
          "total_tax": "0.45",
          "price": "9.00",
          "quantity": 1,
          "tax_class": null,
          "name": "Woo Album #4",
          "product_id": 96,
          "sku": "",
          "meta": []
        }
      ],
      "shipping_lines": [
        {
          "id": 501,
          "method_id": "flat_rate",
          "method_title": "Flat Rate",
          "total": "10.00"
        }
```

```
      ],
      "tax_lines": [
        {
          "id": 502,
          "rate_id": "5",
          "code": "US-CA-TAX-1",
          "title": "Tax",
          "total": "4.40",
          "compound": false
        },
        {
          "id": 503,
          "rate_id": "4",
          "code": "US-STANDARD-1",
          "title": "Standard",
          "total": "1.50",
          "compound": false
        }
      ],
      "fee_lines": [],
      "coupon_lines": [],
      "customer": {
        "id": 2,
        "created_at": "2014-11-19T18:34:19Z",
        "email": "john.doe@example.com",
        "first_name": "",
        "last_name": "",
        "username": "john.doe",
        "last_order_id": "645",
        "last_order_date": "2015-01-26T20:00:21Z",
        "orders_count": 2,
        "total_spent": "19.00",
        "avatar_url":
"https://secure.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=96",
        "billing_address": {
          "first_name": "John",
          "last_name": "Doe",
          "company": "",
          "address_1": "969 Market",
          "address_2": "",
```

```
          "city": "San Francisco",
          "state": "CA",
          "postcode": "94103",
          "country": "US",
          "email": "john.doe@example.com",
          "phone": "(555) 555-5555"
        },
        "shipping_address": {
          "first_name": "John",
          "last_name": "Doe",
          "company": "",
          "address_1": "969 Market",
          "address_2": "",
          "city": "San Francisco",
          "state": "CA",
          "postcode": "94103",
          "country": "US"
        }
      }
    }
  ]
}
```


Available Filters

Filter	Type	Description
`status`	string	Orders by status. eg: `processing` or `cancelled`

□□□□

□ API □□□□□□□□

HTTP □□

PUT

/wc-api/v2/orders/<id>

```
curl -X PUT https://example.com/wc-api/v2/orders/645 \
-u consumer_key:consumer_secret \
-H "Content-Type: application/json" \
-d '{
  "order": {
    "status": "completed"
  }
}'
```

```
var data = {
  order: {
    status: 'completed'
  }
};

WooCommerce.put('orders/645', data, function(err, data, res) {
  console.log(res);
});
```

```
data = {
    "order": {
        "status": "completed"
    }
}

print(wcapi.put("orders/645", data).json())
```

```
<?php
$data = array(
    'order' => array(
        'status' => 'completed'
    )
);

print_r($woocommerce->orders->update(645, $data));
?>
```

```
data = {
  order: {
```

```
    status: "completed"
  }
}

woocommerce.put("orders/645", data).parsed_response
```

> JSON □□□□

```
{
  "order": {
    "id": 645,
    "order_number": 645,
    "created_at": "2015-01-26T20:00:21Z",
    "updated_at": "2015-01-26T20:00:21Z",
    "completed_at": "2015-01-26T20:00:21Z",
    "status": "completed",
    "currency": "USD",
    "total": "79.87",
    "subtotal": "63.97",
    "total_line_items_quantity": 3,
    "total_tax": "5.90",
    "total_shipping": "10.00",
    "cart_tax": "5.40",
    "shipping_tax": "0.50",
    "total_discount": "0.00",
    "shipping_methods": "Flat Rate",
    "payment_details": {
      "method_id": "bacs",
      "method_title": "Direct Bank Transfer",
      "paid": true
    },
    "billing_address": {
      "first_name": "John",
      "last_name": "Doe",
      "company": "",
      "address_1": "969 Market",
      "address_2": "",
      "city": "San Francisco",
```

```
    "state": "CA",
    "postcode": "94103",
    "country": "US",
    "email": "john.doe@example.com",
    "phone": "(555) 555-5555"
  },
  "shipping_address": {
    "first_name": "John",
    "last_name": "Doe",
    "company": "",
    "address_1": "969 Market",
    "address_2": "",
    "city": "San Francisco",
    "state": "CA",
    "postcode": "94103",
    "country": "US"
  },
  "note": "",
  "customer_ip": "127.0.0.1",
  "customer_user_agent": "WordPress/4.1; http://example.com",
  "customer_id": 2,
  "view_order_url": "https://example.com/my-account/view-order/645",
  "line_items": [
    {
      "id": 504,
      "subtotal": "43.98",
      "subtotal_tax": "4.40",
      "total": "43.98",
      "total_tax": "4.40",
      "price": "21.99",
      "quantity": 2,
      "tax_class": "reduced-rate",
      "name": "Premium Quality",
      "product_id": 546,
      "sku": "",
      "meta": []
    },
    {
      "id": 505,
```

```
      "subtotal": "19.99",
      "subtotal_tax": "1.00",
      "total": "19.99",
      "total_tax": "1.00",
      "price": "19.99",
      "quantity": 1,
      "tax_class": null,
      "name": "Ship Your Idea",
      "product_id": 613,
      "sku": "",
      "meta": [
        {
          "key": "pa_color",
          "label": "Color",
          "value": "Black"
        }
      ]
    }
  ],
  "shipping_lines": [
    {
      "id": 506,
      "method_id": "flat_rate",
      "method_title": "Flat Rate",
      "total": "10.00"
    }
  ],
  "tax_lines": [
    {
      "id": 507,
      "rate_id": "5",
      "code": "US-CA-TAX-1",
      "title": "Tax",
      "total": "4.40",
      "compound": false
    },
    {
      "id": 508,
      "rate_id": "4",
```

```
        "code": "US-STANDARD-1",
        "title": "Standard",
        "total": "1.50",
        "compound": false
      }
    ],
    "fee_lines": [],
    "coupon_lines": [],
    "customer": {
      "id": 2,
      "created_at": "2014-11-19T18:34:19Z",
      "email": "john.doe@example.com",
      "first_name": "",
      "last_name": "",
      "username": "john.doe",
      "last_order_id": "645",
      "last_order_date": "2015-01-26T20:00:21Z",
      "orders_count": 2,
      "total_spent": "19.00",
      "avatar_url":
"https://secure.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=96",
      "billing_address": {
        "first_name": "John",
        "last_name": "Doe",
        "company": "",
        "address_1": "969 Market",
        "address_2": "",
        "city": "San Francisco",
        "state": "CA",
        "postcode": "94103",
        "country": "US",
        "email": "john.doe@example.com",
        "phone": "(555) 555-5555"
      },
      "shipping_address": {
        "first_name": "John",
        "last_name": "Doe",
        "company": "",
        "address_1": "969 Market",
```

```
        "address_2": "",
        "city": "San Francisco",
        "state": "CA",
        "postcode": "94103",
        "country": "US"
      }
    }
  }
}
```

Delete An Order

This API helps you delete an order.

HTTP Request

DELETE

/wc-api/v2/orders/<id>

```
curl -X DELETE https://example.com/wc-api/v2/orders/645/?force=true \
-u consumer_key:consumer_secret
```

```
WooCommerce.delete('orders/645/?force=true', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.delete("orders/645/", params={"force": True}).json())
```

```
<?php print_r($woocommerce->orders->delete(645, true)); ?>
```

```
woocommerce.delete("orders/645/, force: true).parsed_response
```

> JSON response example:

```
{
  "message": "Permanently deleted order"
```

```
}
```

Parameters

Parameter	Type	Description
`force`	string	Use `true` whether to permanently delete the order, defaults to `false`. Note that permanently deleting the order will return HTTP 200 rather than HTTP 202.

□□□□□

□ API □□□□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders/count

```
curl https://example.com/wc-api/v2/orders/count \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/count', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/count").json())
```

```
<?php print_r($woocommerce->orders->get_count()); ?>
```

```
woocommerce.get("orders/count").parsed_response
```

> JSON □□□□

```
{
  "count": 2
}
```

Available Filters

Filter	Type	Description
`status`	string	Orders by status. eg: `processing` or `cancelled`

□□□□□□□

□ API □□□□□□□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders/statuses

```
curl https://example.com/wc-api/v2/orders/statuses \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/statuses', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/statuses").json())
```

```
<?php print_r($woocommerce->orders->get_statuses()); ?>
```

```
woocommerce.get("orders/statuses").parsed_response
```

> JSON □□□□□

```
{
  "order_statuses": {
```

```
    "pending": "Pending Payment",
    "processing": "Processing",
    "on-hold": "On Hold",
    "completed": "Completed",
    "cancelled": "Cancelled",
    "refunded": "Refunded",
    "failed": "Failed"
  }
}
```

□□□□□□

□ API □□□□□□□□□□□□□□

HTTP □□

POST

/wc-api/v2/orders/<id>/notes

```
curl -X POST https://example.com/wc-api/v2/orders/645/notes \
□-u consumer_key:consumer_secret \
□-H "Content-Type: application/json" \
□-d '{
  "order_note": {
    "note": "Order ok!!!"
  }
}'
```

```
var data = {
  order_note: {
    note: 'Order ok!!!'
  }
};

WooCommerce.post('orders/645/notes', data, function(err, data, res) {
  console.log(res);
});
```

```
data = {
    "order_note": {
        "note": "Order ok!!!"
    }
}

print(wcapi.post("orders/645/notes", data).json())
```

```
<?php
$data = array(
    'order_note' => array(
        'note' => 'Order ok!!!'
    )
);

print_r($woocommerce->order_notes->create(645, $data));
?>
```

```
data = {
  order_note: {
    note: "Order ok!!!"
  }
}

woocommerce.post("orders/645/notes", data).parsed_response
```

> JSON □□□□□

```
{
  "order_note": {
    "id": "416",
    "created_at": "2015-01-26T20:56:44Z",
    "note": "Order ok!!!",
    "customer_note": false
  }
}
```

Order Notes Properties

Attribute	Type	Description
`id`	integer	Order note ID *read-only*
`created_at`	string	UTC DateTime when the order note was created *read-only*
`note`	string	Order note *required*
`customer_note`	boolean	Shows/define if the note is only for reference or for the customer (the user will be notified). Default is `false`

□□□□□

□ API □□□□□□□□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders/<id>/notes/<note_id>

```
curl https://example.com/wc-api/v2/orders/645/notes/416 \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/645/notes/416', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/645/notes/416").json())
```

```
<?php print_r($woocommerce->order_notes->get(645, 416)); ?>
```

```
woocommerce.get("orders/645/notes/416").parsed_response
```

> “ JSON □□□□

```
{
  "order_note": {
    "id": "416",
    "created_at": "2015-01-26T20:56:44Z",
    "note": "Order ok!!!",
    "customer_note": false
  }
}
```

□□ □□□□□□ □□□□□□□□□□□□□□□□□

View List Of Notes From An Order

This API helps you to view all the notes from an order.

HTTP Request

GET

/wc-api/v2/orders/<id>/notes

```
curl https://example.com/wc-api/v2/orders/645/notes \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/645/notes', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/645/notes").json())
```

```
<?php print_r($woocommerce->order_notes->get(645)); ?>
```

```
woocommerce.get("orders/645/notes").parsed_response
```

> JSON response example:

```
{
  "order_notes": [
    {
      "id": "416",
      "created_at": "2015-01-26T20:56:44Z",
      "note": "Order ok!!!",
      "customer_note": false
    },
    {
      "id": "415",
      "created_at": "2015-01-26T20:16:14Z",
      "note": "Order status changed from Processing to Completed.",
      "customer_note": false
    },
    {
      "id": "412",
      "created_at": "2015-01-26T20:00:21Z",
      "note": "Order item stock reduced successfully.",
      "customer_note": false
    },
    {
      "id": "411",
      "created_at": "2015-01-26T20:00:09Z",
      "note": "Order status changed from Pending Payment to Processing.",
      "customer_note": false
    }
  ]
}
```


View the Order Notes Properties for more details on this response.

□□□□□

□ API □□□□□□□□□

HTTP □□

PUT

/wc-api/v2/orders/<id>/notes/<note_id>

```
curl -X PUT https://example.com/wc-api/v2/orders/645/notes/416 \
-u consumer_key:consumer_secret \
-H "Content-Type: application/json" \
-d '{
  "order_note": {
    "note": "Ok!"
  }
}'
```

```
var data = {
  order_note: {
    note: 'Ok!'
  }
};

WooCommerce.put('orders/645/notes/416', data, function(err, data, res) {
  console.log(res);
});
```

```
data = {
    "order_note": {
        "note": "Ok!"
    }
}

print(wcapi.put("orders/645/notes/416", data).json())
```

```
<?php
$data = array(
    'order_note' => array(
        'note' => 'Ok!'
    )
);

print_r($woocommerce->order_notes->update(645, 416, $data));
?>
```

```
data = {
  order_note: {
    note: "Ok!"
  }
}

woocommerce.put("orders/645/notes/416", data).parsed_response
```

> JSON □□□□□

```
{
  "order_note": {
    "id": "416",
    "created_at": "2015-01-26T20:56:44Z",
    "note": "Ok!",
    "customer_note": false
  }
}
```

□□ □□□□□□ □□□□□□□□□□□□□□□□□

Delete An Order Note

This API helps you delete an order note.

HTTP Request

DELETE

/wc-api/v2/orders/<id>/notes/<note_id>

```
curl -X DELETE https://example.com/wc-api/v2/orders/645/notes/416 \
-u consumer_key:consumer_secret
```

```
WooCommerce.delete('orders/645/notes/416', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.delete("orders/645/notes/416").json())
```

```
<?php print_r($woocommerce->order_notes->delete(645, 416)); ?>
```

```
woocommerce.delete("orders/645/notes/416").parsed_response
```

> JSON response example:

```
{
  "message": "Permanently deleted order note"
}
```

□□□□□□

□ API □□□□□□□□□□□□□□

HTTP □□

POST

/wc-api/v2/orders/<id>/refunds

```
curl -X POST https://example.com/wc-api/v2/orders/645/refunds \
□-u consumer_key:consumer_secret \
□-H "Content-Type: application/json" \
□-d '{
  "order_refund": {
    "amount": 10
  }
}'
```

```
var data = {
  order_refund: {
    amount: 10
  }
};
```

```
WooCommerce.post('orders/645/refunds', data, function(err, data, res) {
  console.log(res);
});
```

```
data = {
    "order_refund": {
        "amount": 10
    }
}

print(wcapi.post("orders/645/refunds", data).json())
```

```
<?php
$data = array(
    'order_refund' => array(
        'amount' => 10
    )
);

print_r($woocommerce->order_refunds->create(645, $data));
?>
```

```
data = {
  order_refund: {
    amount: 10
  }
}

woocommerce.post("orders/645/refunds", data).parsed_response
```

> JSON □□□□□

```
{
  "order_refund": {
    "id": 649,
    "created_at": "2015-01-26T19:29:32Z",
    "amount": "10.00",
    "reason": "",
```

```
    "line_items": []
  }
}
```

Order Refunds Properties

Attribute	Type	Description
`id`	integer	Order note ID *read-only*
`created_at`	string	UTC DateTime when the order refund was created *read-only*
`amount`	string	Refund amount *required*
`reason`	string	Reason for refund
`line_items`	array	List of order items to refund. See Line Items Properties

□□□□□

□ API □□□□□□□□□□□□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders/<id>/refunds/<refund_id>

```
curl https://example.com/wc-api/v2/orders/645/refunds/649 \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/645/refunds/649', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/645/refunds/649").json())
```

```
<?php print_r($woocommerce->order_refunds->get(645, 649)); ?>
```

```
woocommerce.get("orders/645/refunds/649").parsed_response
```

> JSON □□□□□


```
{
  "order_refund": {
    "id": 649,
    "created_at": "2015-01-26T19:29:32Z",
    "amount": "10.00",
    "reason": "",
    "line_items": []
  }
}
```


□□ □□□□□□ □□□□□□□□□□□□□□□□

□□□□□□□

□ API □□□□□□□□□□□□□□□□□

HTTP □□

GET

/wc-api/v2/orders/<id>/refunds

```
curl https://example.com/wc-api/v2/orders/645/refunds \
□-u consumer_key:consumer_secret
```

```
WooCommerce.get('orders/645/refunds', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.get("orders/645/refunds").json())
```

```
<?php print_r($woocommerce->order_refunds->get(645)); ?>
```

```
woocommerce.get("orders/645/refunds").parsed_response
```

>

> JSON □□□□□


```
{
  "order_refunds": [
    {
      "id": 649,
      "created_at": "2015-01-26T19:29:32Z",
      "amount": "10.00",
      "reason": "",
      "line_items": []
    },
    {
      "id": 647,
      "created_at": "2015-01-26T19:19:06Z",
      "amount": "21.99",
      "reason": "",
      "line_items": [
        {
          "id": 514,
          "subtotal": "-21.99",
          "subtotal_tax": "0.00",
          "total": "-21.99",
          "total_tax": "0.00",
          "price": "-21.99",
          "quantity": 1,
          "tax_class": "reduced-rate",
          "name": "Premium Quality",
          "product_id": 546,
          "sku": "",
          "meta": []
        },
        {
          "id": 515,
          "subtotal": "0.00",
          "subtotal_tax": "0.00",
          "total": "0.00",
          "total_tax": "0.00",
          "price": "0.00",
          "quantity": 0,
```

```
          "tax_class": null,
          "name": "Ship Your Idea",
          "product_id": 613,
          "sku": "",
          "meta": []
        }
      ]
    }
  ]
}
```

有关此响应的更多详细信息，请查看 订单退款属性_____。

更新订单退款

此 API 允许您修改订单退款。

HTTP 请求

PUT

/wc-api/v2/orders/<id>/refunds/<refund_id>

```
curl -X PUT https://example.com/wc-api/v2/orders/645/refunds/649 \
-u consumer_key:consumer_secret \
-H "Content-Type: application/json" \
-d '{
  "order_refund": {
    "reason": "Because was it necessary!"
  }
}'
```

```
var data = {
  order_refund: {
    reason: 'Because was it necessary!'
  }
};

WooCommerce.put('orders/645/refunds/649', data, function(err, data, res) {
```

```
  console.log(res);
});
```

```
data = {
    "order_refund": {
        "reason": "Because was it necessary!"
    }
}

print(wcapi.put("orders/645/refunds/649", data).json())
```

```
<?php
$data = array(
    'order_refund' => array(
        'reason' => 'Because was it necessary!'
    )
);

print_r($woocommerce->order_refunds->update(645, 649, $data));
?>
```

```
data = {
  order_refund: {
    reason: "Because was it necessary!"
  }
}

woocommerce.put("orders/645/refunds/649", data).parsed_response
```

> JSON □□□□□

```
{
  "order_refund": {
    "id": 649,
    "created_at": "2015-01-26T19:29:32Z",
    "amount": "10.00",
    "reason": "Because was it necessary!",
    "line_items": []
```

```
  }
}
```

□□ □□□□□□ □□□□□□□□□□□□□□□□

Delete An Order Refund

This API helps you delete an order refund.

HTTP Request

DELETE

/wc-api/v2/orders/<id>/refunds/<refund_id>

```
curl -X DELETE https://example.com/wc-api/v2/orders/645/refunds/649 \
□-u consumer_key:consumer_secret
```

```
WooCommerce.delete('orders/645/refunds/649', function(err, data, res) {
  console.log(res);
});
```

```
print(wcapi.delete("orders/645/refunds/649").json())
```

```
<?php print_r($woocommerce->order_refunds->delete(645, 649)); ?>
```

```
woocommerce.delete("orders/645/refunds/649").parsed_response
```

> JSON response example:

```
{
  "message": "Permanently deleted refund"
}
```

□□□ #2
□ □□□ □□□ 2026-03-26 14:00:56 UTC
□ □□□ □□□ 2026-03-26 14:00:56 UTC