

---

# ntokloapi-php Documentation

*Release beta*

**Fu Hoang**

September 25, 2015



<b>1</b>	<b>Requirements</b>	<b>3</b>
<b>2</b>	<b>Features</b>	<b>5</b>
<b>3</b>	<b>How do I use it?</b>	<b>7</b>
<b>4</b>	<b>Step 1</b>	<b>9</b>
<b>5</b>	<b>Step 2</b>	<b>11</b>
<b>6</b>	<b>Step 3</b>	<b>13</b>
6.1	Installation . . . . .	13
6.2	Usage . . . . .	13
6.3	Reference . . . . .	17
<b>7</b>	<b>Indices and tables</b>	<b>21</b>



ntokloapi-php is an API connector to use the [nToklo recommendation engine](#) for your e-commerce site.



---

## Requirements

---

- PHP 5.3.3 is required but using the latest version of PHP is highly recommended





---

### Features

---

- Events
- Products
- Blacklists
- Recommendations
- Charts



---

### How do I use it?

---

To be able to use the nToklo recommendation engine, you need to create an application in the nToklo console so you can get an API key and API secret.

To get your API key and API secret you must follow the steps below:



---

**Step 1**

---

Register with nToklo



---

### Step 2

---

Create an application on the [nToklo console](#). An application represent your store on nToklo platform.





---

## Step 3

---

Get your nToklo API key and secret and put into the function.

```
$api = new NtokloApi($key = '1112233asdZmUtZmQ4Yy00MjQ4LThjODAtNzBjMjJlODRjYjVh', $secret = 'OTNmMjll
```

Contents:

## 6.1 Installation

To install the nToklo API connector you can do it through composer:

```
$ composer require ntoklo/ntokloapi-php
```

Or if you want you can install the latest version from the git repository:

```
$ git clone https://github.com/nToklo/ntokloapi-php.git
```

Once you have installed it, you will need to include the require path to autoload.php and instantiate a new NtokloApi class and insert the API key and secret into the function.

```
require "vendor/autoload.php";  
$api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
```

Then you can access the functionality like this:

```
$api->postEvent( $array );
```

## 6.2 Usage

To use the ntokloapi connector you just need to include the require path to autoload.php and instantiate a new NtokloApi class and insert the API key and secret into the function.

```
require "vendor/autoload.php";  
$api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
```

### 6.2.1 Universal Variable

The nToklo recommendation engine uses UV(Universal Variable) objects to create the recommendations. UV is a type of JSON object that has a specific set of keys to manage ecommerce entries. You can check the specification [here](#)

**Note:** The ntokloapi will automatically encode the Universal Variable into JSON if it's in PHP format. If the UV is in JSON you will need to use the PHP `json_decoded()` before posting into the API.

---

For example Universal Variable in JSON:

```
1 uv = {
2     "version": 1.2, # If this doesn't exist, the connector will assume latest
3     "user": {
4         "user_id": "nicole12@example.com",
5         "name": "Nicole Watts",
6         "username": "nicoleuser123"
7     },
8     "events": [
9         {
10             "category": "conversion_funnel",
11             "action": "preview"
12         }
13     ],
14     "product": {
15         "id": "1234",
16         "url": "http://www.fashionbay.com/women/shoes/french_sole_flats_1021.html",
17         "name": "French Sole: Classic Ballet",
18         "unit_price": 99,
19         "unit_sale_price": 69,
20         "description": "This is beautifully soft, unstructured ballet flat is ...",
21         "category": "shoes"
22     }
23 }
```

For example Universal Variable in PHP:

```
1 $uv = array('
2     "version"      => "1.2",
3     "user"         => array("user_id" => "112"),
4     "product"      => array(
5         "id"                => "10201",
6         "name"              => "Gabardine A-line skirt",
7         "category"         => "Womens > Skirts",
8         "currency"         => "GBP",
9         "unit_sale_price"   => 98
10    ),
11     "events"       => array(
12         (object) [
13             "category" => "conversion_funnel", "action" => "browse"
14         ]
15    )
16 );
```

## 6.2.2 Products

To keep track of the products, you have to send them first. **It's not a requirement** but if you have a big catalog it will allow you to preprocess the data before starting to send events.

Example:

```

1 require "vendor/autoload.php";
2 $api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
3
4 $uv = array(
5     "version"          => "1.2", // If this doesn't exist, the connector will assume latest
6     "product" => array(
7         "id"           => "10201",
8         "name"          => "Gabardine A-line skirt",
9         "category"      => "Womens > Skirts",
10        "currency"       => "GBP",
11        "unit_sale_price" => 98
12    )
13 );
14
15 $response = $api->postProduct($uv);
16
17 echo $response; //return true if post is successful
18

```

### 6.2.3 Events

An event in the nToklo recommendation system means some kind of action that has performed by the user.

Example:

```

1 require "vendor/autoload.php";
2 $api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
3
4 $uv = array('
5     "version"          => "1.2", // If this doesn't exist, the connector will assume latest
6     "user_id"          => "112",
7     "product" => array(
8         "id"           => "10201",
9         "name"          => "Gabardine A-line skirt",
10        "category"      => "Womens > Skirts",
11        "currency"       => "GBP",
12        "unit_sale_price" => 98
13    ),
14    "events" => array(
15        (object)[
16            "category" => "conversion_funnel", "action" => "browse"
17        ]
18    )
19 );
20
21 $response = $api->postEvent($uv);
22
23 echo $response; // Return true if post is successful
24

```

### 6.2.4 Recommendations

This is the core of the system, the recommendations. This function will return to you a JSON objects with the recommended products for your user and a temporary token.

Example:

```
1 require "vendor/autoload.php";
2 $api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
3
4 $recommendation = $api->recommendations($userID = null, $productId = '10201', $scope = null, $value = null);
5
6 print_r( $recommendation );
```

It should return something like this:

```
1 {
2     "tracker_id": "2b169680-aa86-12e5-9c37-600308a4f234",
3     "items": [
4         {
5             "id": "10242",
6             "name": "Campbell capri in wave print",
7             "currency": "GBP",
8             "unit_sale_price": 98
9         },
10        {
11            "id": "22832",
12            "name": "Andie chino",
13            "currency": "GBP",
14            "unit_sale_price": 79.5
15        },
16        {
17            "id": "12955",
18            "name": "Collection cropped trouser in antique floral",
19            "currency": "GBP",
20            "unit_sale_price": 298
21        }
22    ]
23 }
```

## 6.2.5 Blacklist

The blacklist functionality allows you to add products to a blacklist so they don't show up on the recommendations.

Example:

```
1 require "vendor/autoload.php";
2 $api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
3
4 // Add one product to the blacklist.
5 $api->addBlacklist( '12' );
6
7 // Remove a product from the blacklist.
8 $api->removeBlacklist( '12' );
9
10 //List all the currently blacklisted products
11 $api->fetchBlacklist();
```

## 6.2.6 Charts

Charts allows you to pull information regarding your analytics. It's not a full report, for that you will have to use the [nToklo Console](#). Charts contains a number of options that will be useful to you for filtering the information. Please refer to the [Reference](#).

Example:

```

1 require "vendor/autoload.php";
2 $api = new NtokloApi( 'nToklo API key', 'nToklo API secret' );
3
4 $chart = $api->chart( $timestamp = '1364169600000', $scope = null, $value = null, $action = null , $
5
6
7 print_r( $chart );

```

It should return something like this:

```

1 {
2   "tracker_id": "d91a6a80-a181-12e5-9c37-600308a4f234",
3   "items": [
4     {
5       "currentPosition": 1,
6       "previousPosition": 2,
7       "peakPosition": 1,
8       "score": 58,
9       "timesOnChart": 6,
10      "product": { ... }
11    },
12    {
13      "currentPosition": 2,
14      "previousPosition": 1,
15      "peakPosition": 1,
16      "score": 45,
17      "timesOnChart": 12,
18      "product": { ... }
19    },
20    {
21      "currentPosition": 3,
22      "previousPosition": 0,
23      "peakPosition": 3,
24      "score": 39,
25      "timesOnChart": 1,
26      "product": { ... }
27    }
28  ]
29 }

```

## 6.3 Reference

### 6.3.1 NtokloApi class

**class NtokloApi**

**property \$api\_base**

**\_\_construct** (*\$key*, *\$secret*)

**postEvent** (*\$data*)

This function will allow to post a decoded json object into the nToklo api @uses array | decode json object

Example: *\$data* = array("version" => "1.2",

"user" => array("user\_id" => "112"), "product" => array("id" => "10201",

```

    "name" => "Gabardine A-line skirt", "category" => "Womens > Skirts", "currency" =>
    "GBP", "unit_sale_price" => 98),

```

```

    "events" => array((object)["category" => "conversion_funnel", "action" => "browse" ] ) );

```

@return bool

#### **postProduct** (*\$data*)

This function will allow to post a decode json object into the nToklo api @uses array | decode json object

Example: *\$data* = array("version" => "1.2",

```

    "product" => array("id" => "10201",

```

```

        "name" => "Gabardine A-line skirt", "category" => "Womens > Skirts", "currency" =>
        "GBP", "unit_sale_price" => 98 )

```

```

    );

```

@return bool

#### **getProduct** (*\$id*)

This function will allow to post a decode json object into the nToklo api @param int product id. @return bool

#### **recommendations** (*\$userId = null, \$productId = null, \$scope = null, \$value = null*)

This function will fetch the recommendations from nToklo api @param string \$userId optional. Uniquely identifies the user for which the recommendations are intended.

The userId can be any value of string type. Examples: [dan@gmail.com](mailto:dan@gmail.com), 11245901, user\_123

@param string \$productId optional. The product for which to base recommendations from. The productId can be any string value. Example: 10201,prod8513 @param string \$scope optional. A product attribute for which to scope recommendations.

For example scope=category will consider the product category when returning recommendations. Supports: category, manufacturer, vendor, action.

**@param string \$value optional. The value for the recommendation scope.** For example scope=category&value=shoes will consider the shoe category when returning recommendations. The value parameter can be any string value. Example: shoes, category12, nike, shoes.com

@return Json

#### **chart** (*\$timestamp = null, \$scope = null, \$value = null, \$action = null, \$tw = null, \$maxItems = null*)

This function will fetch the charts from nToklo api @param string \$timestamp Optional. The date for which to retrieve a chart. The date should be an epoch timestamp in milliseconds, truncated to midnight Example: 1364169600000 @param string \$scope Optional. A product attribute for which to scope recommendations.

For example scope=category will consider the product category when returning recommendations. Supports: category, manufacturer, vendor, action.

**@param string \$value Optional. The value for the recommendation scope.** For example scope=category&value=shoes will consider the shoe category when returning recommendations. The value parameter can be any string value. Example: shoes, category12, nike, shoes.com

@param string \$action Optional Filters the requested chart by conversion\_funnel actions. If it's not specified then the chart returned is all actions, equivalent to action=all. @param string \$tw Optional. The time window for which the charts are requested. If not specified then the chart returns daily chart, equivalent to tw=DAILY. Supports: DAILY, WEEKLY. @param string \$maxItems Optional. The max number of items in the charts. Default is 10. Valid range is 1-100.

@return Json object

**addBlacklist** (*\$productId = null*)

This function will post an product id to nToklo api to blacklist @param string \$productId The unique identifier for the product which will added/removed from the blacklist.

The productId can be any string value. In order to add/remove products in batches the productId can be specified multiple times.

@return bool

**removeBlacklist** (*\$productId = null*)

This function will post an product id to nToklo api remove prodct from blacklist @param string \$productId The unique identifier for the product which will added/removed from the blacklist.

The productId can be any string value. In order to add/remove products in batches the productId can be specified multiple times.

@return bool

**fetchBlacklist** ()

This function will fetch all the blacklisted products @return json object

### 6.3.2 NtokloApiBase

class NtokloApiBase

property \$key

property \$secret

property \$endpoint

**\_\_construct** (*\$key, \$secret*)

**api\_request** (*\$method, \$uri, \$data = null*)

This class will take care of authentication and exceptions of the API, retuning the required codes to the functions that use it.

@param string \$method http method POST GET DELETE @param string \$uri @param array \$data post the json object to the nToklo api

@return int status code

**getResponse** ()

### 6.3.3 Curl

class Curl

property \$curl

property \$error

property \$error\_code

property \$error\_message

property \$curl\_error

property \$curl\_error\_code

```
property $curl_error_message
property $http_error
property $http_status_code
property $http_error_message
property $request_headers
property $response_headers
property $response
__construct ()
__destruct ()
```



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## Symbols

`__construct()` (Curl method), [20](#)  
`__construct()` (NtokloApi method), [17](#)  
`__construct()` (NtokloApiBase method), [19](#)  
`__destruct()` (Curl method), [20](#)

## A

`addBlacklist()` (NtokloApi method), [19](#)  
`api_base` (NtokloApi property), [17](#)  
`api_request()` (NtokloApiBase method), [19](#)

## C

`chart()` (NtokloApi method), [18](#)  
`Curl` (class), [19](#)  
`curl` (Curl property), [19](#)  
`curl_error` (Curl property), [19](#)  
`curl_error_code` (Curl property), [19](#)  
`curl_error_message` (Curl property), [20](#)

## E

`endpoint` (NtokloApiBase property), [19](#)  
`error` (Curl property), [19](#)  
`error_code` (Curl property), [19](#)  
`error_message` (Curl property), [19](#)

## F

`fetchBlacklist()` (NtokloApi method), [19](#)

## G

`getProduct()` (NtokloApi method), [18](#)  
`getResponse()` (NtokloApiBase method), [19](#)

## H

`http_error` (Curl property), [20](#)  
`http_error_message` (Curl property), [20](#)  
`http_status_code` (Curl property), [20](#)

## K

`key` (NtokloApiBase property), [19](#)

## N

`NtokloApi` (class), [17](#)  
`NtokloApiBase` (class), [19](#)

## P

`postEvent()` (NtokloApi method), [17](#)  
`postProduct()` (NtokloApi method), [18](#)

## R

`recommendations()` (NtokloApi method), [18](#)  
`removeBlacklist()` (NtokloApi method), [19](#)  
`request_headers` (Curl property), [20](#)  
`response` (Curl property), [20](#)  
`response_headers` (Curl property), [20](#)

## S

`secret` (NtokloApiBase property), [19](#)