TikTok Ads Tracking

- Base Installation
- Event Sending
- Advanced Matching
- Custom Events / Listeners

Base Installation

Prerequisites

- HandL JS library must be loaded on your page
- You must have a TikTok Pixel ID from your TikTok Ads Manager

Step 1: Get Your TikTok Pixel ID

1. Log in to your TikTok Ads Manager
2. Navigate to **Assets → Events**
3. Create a new Pixel or select an existing one
4. Copy your **Pixel ID** (format: `C5A9CTNGE0M9N03H4840`)

Step 2: Initialize TikTok Pixel

Add the following script to your HTML page **before** the HandL script tag:

```
<script>
    // Set your TikTok Pixel ID
    window.handl_tiktok_pixel_id = 'YOUR_TIKTOK_PIXEL_ID_HERE';
</script>

<!-- Load HandL script --> Make sure to change your license key
<script>
var handl_custom_params = [];
var handl_js = document.createElement("script");
handl_js.setAttribute("src", "https://track.utmsimple.com/utm.js?license=<license code goes
here>"), document.head.appendChild(handl_js),
handl_js.onload = function() {
    /*This is the UTM Simple ready event, you can put any code you'd like to trigger after the
script fully loaded here*/
    console.log("I\'m ready")
    console.log("prnting all the cookies")
    console.log(HandL.getAll())
};
</script>
```

Example:

```
<script>
    window.handl_tiktok_pixel_id = 'C5A9CTNGE0M9N03H4840';
</script>
<script src="https://cdn.example.com/utm.js"></script>
```

Step 3: Verify Installation

After the page loads, the TikTok Pixel will automatically:

- Load the TikTok Pixel script
- Fire a `PageView` event
- Initialize advanced matching (if enabled)

You can verify the installation by:

1. Opening browser DevTools (F12)
2. Going to the **Network** tab
3. Filtering for `analytics.tiktok.com`
4. You should see requests to TikTok's analytics endpoint

Event Sending

Basic Event Sending

Use the `HandL.sendTikTokEvent()` function to send events to TikTok:

```
HandL.sendTikTokEvent(eventName, eventData);
```

Parameters:

- `eventName` (string, required): The name of the event
- `eventData` (object, optional): Event properties and parameters

Note: TikTok Pixel only supports one payload parameter (unlike Facebook which supports two).

Standard Events

TikTok supports the following standard events:

PageView

Automatically fired on page load. You can also manually trigger it:

```
HandL.sendTikTokEvent('PageView');
```

Purchase

Track completed purchases:

```
HandL.sendTikTokEvent('Purchase', {
    value: 29.99,
    currency: 'USD',
    content_type: 'product',
    content_id: 'SKU123',
    quantity: 1
});
```

CompleteRegistration

Track when users complete registration:

```
HandL.sendTikTokEvent('CompleteRegistration', {
    content_name: 'Account Registration',
    status: true
});
```

AddToCart

Track when items are added to cart:

```
HandL.sendTikTokEvent('AddToCart', {
    value: 19.99,
    currency: 'USD',
    content_type: 'product',
    content_id: 'SKU456',
    quantity: 1
});
```

InitiateCheckout

Track when checkout process begins:

```
HandL.sendTikTokEvent('InitiateCheckout', {
    value: 49.99,
    currency: 'USD',
    contents: [
        {
            content_id: 'SKU123',
            content_name: 'Product Name',
            quantity: 1,
            price: 29.99
        }
    ]
});
```

ViewContent

Track when users view content:

```
HandL.sendTikTokEvent('ViewContent', {
    content_type: 'product',
    content_id: 'SKU789',
    content_name: 'Product Name',
```

```
    value: 99.99,
    currency: 'USD'
});
```

Search

Track search queries:

```
HandL.sendTikTokEvent('Search', {
    search_string: 'running shoes',
    content_type: 'product'
});
```

Contact

Track contact form submissions:

```
HandL.sendTikTokEvent('Contact', {
    content_name: 'Contact Form',
    content_category: 'Lead Generation'
});
```

Subscribe

Track newsletter subscriptions:

```
HandL.sendTikTokEvent('Subscribe', {
    content_name: 'Newsletter',
    content_category: 'Email Marketing'
});
```

Lead

Track lead generation events:

```
HandL.sendTikTokEvent('Lead', {
    content_name: 'Lead Form',
    value: 0,
    currency: 'USD'
});
```

Custom Events

You can also send custom events with any event name:

```
HandL.sendTikTokEvent('CustomEventName', {
    custom_parameter: 'value',
    another_param: 123
});
```

Event Data Parameters

Common event parameters you can include:

Parameter	Type	Description	Example
`value`	number	Monetary value of the event	`29.99`
`currency`	string	Currency code (ISO 4217)	`'USD'`, `'EUR'`
`content_type`	string	Type of content	`'product'`, `'video'`
`content_id`	string	Unique identifier for content	`'SKU123'`
`content_name`	string	Name of the content	`'Product Name'`
`content_category`	string	Category of content	`'Electronics'`
`quantity`	number	Quantity of items	`2`
`contents`	array	Array of content items	See example above

Advanced Matching

Advanced matching allows TikTok to match events to users more accurately by sending hashed user identifiers.

Requirements

1. **Enable Advanced Matching:**

```
<script>
    window.handl_advanced_matching = 1;
    window.handl_tiktok_pixel_id = 'YOUR_PIXEL_ID';
</script>
```

2. **Collect User Data:** The HandL library automatically collects user data from forms when advanced matching is enabled. It looks for:
   - Email fields (email, email_address, e-mail, mail)
   - Phone fields (phone, telephone, mobile, cell, phone-number, tel)

How It Works

1. **Data Collection:**
   - When users fill out forms, HandL automatically captures email and phone number
   - Data is stored in `localStorage` with keys: `em` (email) and `ph` (phone)
2. **Automatic Hashing:**
   - All identifiers are automatically hashed using SHA-256 before sending to TikTok
   - Email is normalized (lowercase, trimmed)
   - Phone numbers are cleaned (digits only)
3. **External ID:**
   - Uses the `first_handlID` cookie as `external_id`
   - This is automatically generated on first visit
4. **Automatic Identification:**
   - When TikTok Pixel loads and advanced matching is enabled, `ttq.identify()` is called automatically
   - Identification is updated whenever new email/phone data is saved

Manual Data Collection

You can manually save user data for advanced matching:

```
HandL.saveAdvancedMatchingParams({
    em: 'user@example.com',  // Email (will be hashed)
    ph: '+1-555-123-4567'     // Phone (will be cleaned and hashed)
});
```

Note: The data will be automatically hashed when sent to TikTok.

Advanced Matching Parameters

TikTok supports the following identifiers (all SHA-256 hashed):

Parameter	Description	Source
`email`	User's email address	`localStorage.getItem('em')`
`phone_number`	User's phone number	`localStorage.getItem('ph')`
`external_id`	Unique identifier	`Cookies.get('first_handlID')`

Verification

To verify advanced matching is working:

1. Open browser DevTools (F12)
2. Go to **Console** tab
3. Look for: `UTMSimple: TikTok advanced matching parameters sent`
4. Check **Network** tab for requests containing hashed identifiers

Custom Events / Listeners

Listening to TikTok Events

HandL fires custom events you can listen to:

```
// Listen for TikTok initialization
document.body.addEventListener('UTMSimpleTikTokInitCompleted', function(event) {
    console.log('TikTok Pixel initialized');
});

// Listen for any TikTok event completion
document.body.addEventListener('UTMSimpleTikTokEventCompleted', function(event) {
    console.log('TikTok event sent');
});

// Listen for specific event completion
document.body.addEventListener('UTMSimpleTikTokPurchaseCompleted', function(event) {
    console.log('Purchase event sent to TikTok');
});

// Listen for advanced matching completion
document.body.addEventListener('UTMSimpleTikTokAdvancedMatchingCompleted', function(event) {
    console.log('Advanced matching data sent:', event.detail);
});
```

Example: Track Button Clicks

```
document.getElementById('buy-button').addEventListener('click', function() {
    HandL.sendTikTokEvent('InitiateCheckout', {
        value: 99.99,
        currency: 'USD',
        content_type: 'product'
    });
});
```

Example: Track Form Submissions

```
<form id="contact-form" onsubmit="trackFormSubmission(event)">
    <!-- form fields -->
</form>

<script>
function trackFormSubmission(event) {
    // Form submission logic here

    HandL.sendTikTokEvent('Contact', {
        content_name: 'Contact Form',
        content_category: 'Lead Generation'
    });
}
</script>
```