Track OpenAI Ads within UTMSimple and trigger Events

Track OpenAI Ads within UTMSimple and trigger Events

You can track ChatGPT Ads conversions within UTMSimple without adding OpenAI's pixel snippet yourself. UTMSimple loads the OpenAI Ads Measurement Pixel, initializes it with your Pixel ID, and fires a `page_viewed` event on load.

Usage is straightforward. You only need your Pixel ID from the **Conversions** tab in OpenAI Ads Manager. Create a global variable named `openai_pixel_id` and assign your Pixel ID to it before UTMSimple loads:

```
<!-- Global site UTM Simple Tracking Start -->
<script>
    openai_pixel_id = "<YOUR OPENAI PIXEL ID>";
    handl_custom_params = [];
    var handl_js = document.createElement("script");
    handl_js.setAttribute("src", "https://track.utmsimple.com/utm.js?license=<LICENSE KEY>");
    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 loads here. */
    };
</script>
<!-- Global site UTM Simple Tracking End -->
```

If `openai_pixel_id` is not defined, OpenAI tracking stays off — same pattern as `fb_pixel_id` for Facebook.

After that, UTMSimple will:

- Load the OpenAI measurement SDK

- Initialize the pixel with your Pixel ID
- Automatically send a `page_viewed` event
- Store `openai_pixel_id` as a HandL cookie (available for form fields and attribution data)

Optional: Debug mode

While testing your integration, enable console logging from the OpenAI SDK:

```
<script>
    openai_pixel_id = "<YOUR OPENAI PIXEL ID>";
    handl_openai_debug = true;
</script>
```

Open the browser console to inspect pixel activity. Turn this off in production.

Trigger conversion events

To send conversion events (leads, purchases, checkouts, etc.), use `HandL.sendOpenAIEvent()`:

```
HandL.sendOpenAIEvent(eventName, eventProps, eventOptions);
```

Argument	Required	Description
`eventName`	Yes	A supported standard event name, or `custom`
`eventProps`	No	Event data (type, amount, currency, contents, etc.)
`eventOptions`	No	Delivery options such as `event_id` for deduplication

Example events

Lead generation

```
HandL.sendOpenAIEvent('lead_created', {
    type: 'customer_action',
});
```

```
HandL.sendOpenAIEvent('registration_completed', {
    type: 'customer_action',
});
```

Commerce

```
HandL.sendOpenAIEvent('items_added', {
    type: 'contents',
    amount: 2599,
    currency: 'USD',
    contents: [
        {
            id: 'sku_123',
            name: 'Starter bundle',
            content_type: 'product',
            quantity: 1,
            amount: 2599,
            currency: 'USD',
        },
    ],
});

HandL.sendOpenAIEvent('checkout_started', {
    type: 'contents',
    amount: 2599,
    currency: 'USD',
    contents: [
        {
            id: 'sku_123',
            name: 'Starter bundle',
            content_type: 'product',
            quantity: 1,
        },
    ],
});

HandL.sendOpenAIEvent('order_created', {
    type: 'contents',
```

```
    amount: 2599,
    currency: 'USD',
    contents: [
        {
            id: 'sku_123',
            name: 'Starter bundle',
            content_type: 'product',
            quantity: 1,
        },
    ],
});
```

Subscriptions and trials

```
HandL.sendOpenAIEvent('subscription_created', {
    type: 'plan_enrollment',
    plan_id: 'pro_monthly',
    amount: 2000,
    currency: 'USD',
});

HandL.sendOpenAIEvent('trial_started', {
    type: 'plan_enrollment',
    plan_id: 'pro_trial',
});
```

Custom events

For custom events, use `custom` as the event name and set `custom_event_name` in the third argument:

```
HandL.sendOpenAIEvent(
    'custom',
    {
        type: 'custom',
        amount: 12999,
        currency: 'USD',
        plan_id: 'enterprise_annual',
    },
```

```
    {
        custom_event_name: 'quote_requested',
        event_id: 'quote_req_123',
    }
);
```

> **Note:** Put `custom_event_name` and `event_id` in the third argument (`eventOptions`), not inside `eventProps`.

Deduplicate browser and server events

If you also send conversions from a server-side integration, reuse the same `event_id` in both places:

```
HandL.sendOpenAIEvent(
    'order_created',
    {
        type: 'contents',
        amount: 2599,
        currency: 'USD',
    },
    {
        event_id: 'order_12345',
    }
);
```

UTMSimple generates an event ID automatically when you omit one, which is fine for client-only tracking.

What UTMSimple handles automatically

When the OpenAI pixel is enabled, the SDK handles:

- Capturing `oppref` from the landing page URL
- Storing `oppref` in a first-party `__oppref` cookie
- Adding the current page origin as `source_url`
- Timestamping and batching events

You do not need to configure these manually.

UTMSimple events

You can listen for pixel lifecycle events in your own code:

```
document.addEventListener('UTMSimpleOpenAIInitCompleted', function() {
    console.log('OpenAI pixel initialized');
});

document.addEventListener('UTMSimpleOpenAIEventCompleted', function() {
    console.log('OpenAI event sent');
});
```

Tips

- Use **integer values** for `amount` and `quantity` (e.g. cents, not dollars).
- Use only the documented fields inside `contents[]`.
- Keep `handl_openai_debug = true` while testing, then remove it in production.
- For the full list of supported events and payload shapes, see the OpenAI supported events docs.

Related docs

- Track Facebook within UTMSimple and trigger Events
- OpenAI JavaScript Pixel

Revision #2
Created 29 May 2026 20:55:20 by UTMSimple
Updated 29 May 2026 20:59:04 by UTMSimple