

PubAdHub S2S Postback API

API Integration Documentation

Endpoint: GET or POST /api/postback/{access_key}

Base URL: https://statsapi.pubadhub.com

This document provides developers with essential details for integrating PubAdHub server-to-server (S2S) conversion postbacks. It covers authentication via access key, required and optional parameters, response handling, and implementation notes.

1. Overview

1.1 Introduction

The S2S Postback API accepts conversion events from your affiliate / offer network. Authentication is performed by including your platform access_key in the URL path. No AccessKey request header is required for this endpoint.

1.2 Access Key

Important: Contact your PubAdHub Account Manager to obtain your platform access_key and the complete postback URL. Each platform receives a unique key. Do not reuse a key across platforms.

1.3 Integration Overview

To send a conversion postback, fire a GET or POST request to the postback endpoint with your access_key in the path and the required parameters populated.

- Protocol: HTTP GET or POST → JSON response
- Authentication: access_key in URL path (no header)
- Required fields: txnid, cid, edate, payout
- Optional fields: sales, currency, status, s2, s3, surl

2. Endpoint

2.1 Base URL

Base URL: https://statsapi.pubadhub.com

API Format

```
GET or POST /api/postback/{access_key}
```

2.2 HTTP Method

Method: GET or POST

Important: Replace {access_key} with the key provided by your Account Manager. Send parameters as query-string values (GET) or as query/form fields (POST). Do not send an AccessKey header.

2.3 Example Postback URL

```
https://statsapi.pubadhub.com/api/postback/{access_key}?txnid={txnid}&cid={cid}&edate={edate}&payout={payout}&sales={sales}&s2={s2}&s3={s3}&surl={surl}&currency={currency}&status={status}
```

Map your network's macros to the PubAdHub parameter names below. Ensure every required parameter is populated before the postback is fired. Placeholder values that are not replaced by your network (for example {txnid}) are treated as empty and will be rejected.

3. Request Parameters

3.1 Parameter Reference

Parameter	Type	Requirement	Description
txnid	string	Required	Unique transaction or conversion ID from your platform.
cid	string	Required	PubAdHub click ID from the outbound click. Prefer the 32-character MD5 click hash ({LandingClickID}) so the conversion can be matched to the original click.
edate	date / datetime / unix	Required	Conversion event time. Accepts YYYY-MM-DD, a datetime string, or a unix timestamp (seconds or milliseconds). Values are stored in UTC.
payout	number	Required	Commission / payout amount for the conversion.
sales	number	Optional	Sale, GMV, or order amount.
currency	string	Optional	Currency code (for example USD).
status	string	Optional	Conversion status from your network (for example approved or pending).
s2	string	Optional	Additional sub-ID / tracking value.
s3	string	Optional	Additional sub-ID / tracking value.
surl	URL	Optional	Event source URL or landing URL associated with the conversion.

4. Response

4.1 Response Structure

The endpoint returns HTTP 200 for both successful and failed postbacks so that pixel-style or network callbacks are not dropped. Always inspect the response body use message and, when present, error_code to determine the outcome.

4.2 Response Fields

Field	Type	Description
message	string	Human-readable result. On success: Postback received successfully.
error_code	number (errors only)	1 = invalid access key or validation failure. 2 = processing / insert failure.
errors	object (errors only)	Field-level validation details when error_code is 1.

4.3 Example Responses

Success Response

```
{
  "message": "Postback received successfully"
}
```

Validation Failed (error_code 1)

```
{
  "message": "The given data was invalid.",
  "errors": {
    "txnid": ["The transaction ID is required."],
    "cid": ["The click ID is required."],
    "edate": ["The event date is required and must be a valid date."],
    "payout": ["The payout is required and must be a number."]
  },
  "error_code": 1
}
```

Invalid Access Key (error_code 1)

```
{
  "message": "The given data was invalid.",
  "errors": {
    "access_key": ["The access key is invalid."]
  },
  "error_code": 1
}
```

Processing Failed (error_code 2)

```
{
  "message": "something went wrong",
  "error_code": 2
}
```

5. Errors & Troubleshooting

5.1 Troubleshooting

- Confirm the request uses GET or POST against `/api/postback/{access_key}` on `https://statsapi.pubadhub.com`.
- Verify `access_key` is present in the URL path and was issued by your PubAdHub Account Manager.
- Ensure required parameters `txnid`, `cid`, `edate`, and `payout` are populated (no unresolved macros).
- Prefer a 32-character MD5 PubAdHub click hash for `cid` so conversions can be matched.
- Inspect `message` / `error_code` in the JSON body even when HTTP status is 200.
- Set appropriate timeout values and implement proper error and timeout handling.

6. Integration Checklist

Use the following checklist when integrating with the PubAdHub S2S Postback API.

☐	Access key and postback URL received from PubAdHub Account Manager
☐	Postback URL configured with <code>/api/postback/{access_key}</code>
☐	Macros mapped for <code>txnid</code> , <code>cid</code> , <code>edate</code> , <code>payout</code>
☐	Optional macros mapped where available (<code>sales</code> , <code>currency</code> , <code>status</code> , <code>s2</code> , <code>s3</code> , <code>surl</code>)
☐	Test success response: Postback received successfully
☐	Test validation failure handling via <code>error_code</code> / <code>errors</code>
☐	Timeout and redirect settings configured

Appendix A - Quick Reference

Base URL: `https://statsapi.pubadhub.com`

GET or POST `/api/postback/{access_key}`

Authentication	<code>access_key</code> in URL path
Method	GET or POST
Required parameters	<code>txnid</code> , <code>cid</code> , <code>edate</code> , <code>payout</code>

Success message	Postback received successfully
-----------------	--------------------------------