CTV Integration

09/23/2026 8:30 am EDT

Overview

We recommend that you deploy ID5 in CTV environments via a server to server integration. We do support alternative integration pathways depending on your requirements. If required, please reach out to your ID5 customer services representative.

To ensure accurate identity resolution, clients should provide signals, such as hashed emails in the request. We also expect you to store the 'signature' provided in our response on the user's device and provision it to us in future requests for the same user. This is integral to an optimal set up since it helps us re-identify consenting users even if their signals change as well as respect opt out requests.

Below are general instructions on how to retrieve an ID5 ID server-side for CTV inventory and make it available in bid requests to your demand partners. The auction should be delayed in order to retrieve the user's privacy preferences and the ID5 ID prior to sending out bid requests; the amount you delay may vary and so we recommend making it configurable for optimization purposes.

Process Flow

1. Check the stored cache for an available ID5 ID
    - ***IF*** there is no cached ID
      OR if the ID needs to be refreshed (we recommend a cache of max 8 hours)
      OR if the user's privacy preferences have changed:
        - Initialize a new HTTP POST request to the ID5 endpoint. If you previously had a stored response for a given user on their device, then subsequent requests must include the "signature" as part of the request to ID5;
        - Store the response from ID5 locally (either in local storage or in a database on the server);
    - ***ELSE*** if there is a valid, up-to-date value in cache:
        - Pull the latest ID5 ID response from the cache
2. Using the response from step 1 above (from the request or cache), prepare the bid request data.
    - Extract the **eid object(s)** from the ids structure (e.g., from `ids.id5id.eid` for the ID5 Universal ID, and any additional enriched IDs if services like Real-Time Bid Enrichment are enabled).
3. Include this data in each bid request to your demand partners as an eids array in the user object. The EID structure is already formatted for direct use in OpenRTB bid requests. For more information on how this is typically done, including handling provenance fields for enriched IDs, review the OpenRTB EID Enrichment Guide.

Caching the response

The response we return for the request will need to be cached as some of the response data needs to be used in future requests for your users. Use the `ts` parameter sent in the request in order to set a TTL for the cache refresh and a TTL for the cache deletion.

- The TTL for the cache refresh should be set to 8 hours. After the 8 hours, on the back of a user event, a new request to ID5 should be made and the cached data refreshed using the data from the API response.
- The TTL for the cache deletion should be set to 30 days. The TTL for the cache deletion should be refreshed with every response from the API. If there will not be a user event in the app, in the 30 days time window, the data should be deleted.

Building Server-Side Request

Server-Side Fetch endpoints for CTV

> **Global**
>
> https://api.id5-sync.com/gc/v1

Request Type

HTTP POST with JSON body.

Request Headers

`Content-Type: application/json`

Partner Number

The `PARTNER` in the request body will be an ID5-provided Partner Number. This value will be static for you once we set you up in our system. You may use the endpoint during testing with the Partner Number 173. If you haven't already been assigned a Partner Number, please contact us to request one.

Server-Side Request Parameters

Request Body

Name	Required	Type	Description	Format	Example
ts	x	string	Timestamp in form of string which extends the ISO-8601 extended offset date-time format to add the time-zone	`yyyymmddThhmmss<ffffff>+|-hhmm`	`2013-02-01T12:52:34+09:00`
partner	x	integer	Partner Number provided by ID5	int32	173
appid	x	string	The store ID of the app in an app store (e.g., Apple App Store, Google Play).		
			Array of Identifiers For		

Name	Required	Type	Description	[{"id":"<ID>", "type":"<TYPE>"}, {<other identifiers if available>}]	[{"id":"bfd378ed-483e-4b11-be1e-12a599d042e0", "type":"DPID"}]
identifiers	when available	Identifier array	Identifier consists of `id` and `type`. Check Identifier object for supported types		
ver	x		Application version		
provider		string	Provider of the service (value to be provided by ID5)		
ip	x	string	IPv4 address closest to device	ipv4 dot-decimal	
ua	x	string	The User Agent of the device's default browser		Mozilla/5.0 (Linux; Android 11; moto e20 Build/RONS31.267-88-3; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/109.0.5414.118 Mobile Safari/537.36
hardware_code		string	The code pertaining to the device (i.e. TV model)		
signature	when available	string	The ID5 signature from a previous call, cached on the device or your server-side		
gdpr		integer	1 if gdpr applies to this request, 0 otherwise		
gdpr_consent		string	(where applicable) the TCF compliant consent string or see 'allowed_vendors'		
allowed_vendors		string array	ID5 Partner identifiers (starting with 'ID5-') or IAB Vendor IDs of vendors allowed to use the ID5ID		`['ID5-78','134']`
us_privacy		string	US Privacy Consent		`1YNN`
gpp_sid		string	The GPP section ID(s) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up		`6,7`

Name	Required	Type	Description	Format	Example
			a comma. More information in GPP documentation		
gpp_string		string	A valid IAB Global Privacy Platform consent string.		`DBABzw~1---~BAAAAAAAAQA.QA`
hem		string	sha256 hash of the cleansed e-mail address. Learn how to cleanse the data here	sha256	`f97ea86ed181d60b0ba62a30579f1e10ad71eaf21b548e173de75718065c533f`
phone		string	sha256 hash of the cleansed phone number	sha256	
idfv		string	Apple ID for Vendors	uuid	`3fa85f64-5717-4562-b3fc-2c963f66afa6`
ipv6		string	The IPV6 of the device	ipv6	`2001:0db8:5b96:0000:0000:426f:8e17:642a`
country		string	Country the user is located in	ISO-3166-1-alpha-2	`AD`
region		string	Region code using ISO-3166-2; 2-letter state code if USA	ISO-3166-2; 2-letter state code if USA	
city		string	City using United Nations Code for Trade & Transport Locations format	United Nations Code for Trade & Transport Locations format	
accept_language		string	A string representing languages accepted by end-user device, should be compatible with browser Accept-Language header	Accept-Language header format	`de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7`
segments		objects array	The segment ids a user may belong to and the destination platform that the segments should be pushed to. Only certain destination platforms are supported and there are backend configurations that need to be made in both ID5's and the destination platform's systems before this feature can be used. Please reach out to your ID5 representative or	Array of segment objects (see below Segment object)	`[{ "destination": "999", "ids": [ "12345", "67890" ] }]`

Name	Required	Type	Description	Format	Example
			information and to get started.		
ottid	deprecated	string	The user ID provided by the OTT/OEM. This will be the equivalent of an hard signal 3rd party cookie.		
ctvid	deprecated	string	The specific user id for the CTV ecosystem the user is observed in, as received by the CTV/OTT/Publisher provider		`00000000-34sd-gfss-sd54-12345678abcd`
ctvid_type	deprecated	string	CTV ID type		`tifa`
puid	deprecated	string	Partner specific user ID		

Identifier Object

Name	Required	Type	Description	Example
id	x	string	The IFA value, usually in the UUID format. Should be lowercased	`bfd378ed-483e-4b11-be1e-12a599d042e0`
type	x	string	The type of identifier provided, case insensitive. Supported values: AAID - Android TV RIDA - Roku AFAI - Amazon Fire IDFA - Apple tvOS MSAI - Xbox/Microsoft TIFA - Samsung (Tizen ID For Advertizing) LG - LG ID VIDA - Vizio Advertising ID PSID - PlayStation ID DPID - Generic device platform ID PPID - Publisher provided ID SSPID - SSP provided ID SESSIONID - Short-lived session ID UNKNOWN - any other value will be treated as UNKNOWN type	`DPID`

Segment Object

Name	Required	Type	Description	Example
destination	x	string	The destination platform. Should be the IAB Vendor ID	`'999'`
ids	x	string array	List of segment ids to add the user to	`[ '12345', '67890' ]`

Response Body

Name	Required	Type	Description	Format	Example
created_at	x	string	Timestamp in the form of a string that extends the ISO-8601 extended offset date-time format to include the time-zone	`yyyymmddThhmmss<ffffff>+|-hhmm`	`2013-02-01T12:52:34+09:00`
original_uid	x	string	A first-party user ID that remains stable for this user on the domain. It is intended for publisher reference only and should not be shared with other partners. The value will be encrypted and may change periodically, even for the same user on the same domain, while the underlying value remains stable. If ID5 lacks consent, then the value will be "0".		
universal_uid	x	string	This is the UID used for sharing with other parties. The value will be encrypted and will change periodically, even for the same user on the same domain. If ID5 does not have consent, then the value will be "0".		
ids	x	object	Container for all identity providers returned by ID5 in OpenRTB-compatible format See the ids object below.		
privacy	x	object	An object containing privacy information See the privacy object below.		
signature	x	string	The ID5 Signature is a string that must be stored by the caller (either cached on the device or your server-side) and sent back to ID5 with all future requests. See below for details about the Signature.		

Name	Required	Type	Description	Format	Example
ext	x	object	See the extension object below.		
gpId		string	Guarded Publisher ID (see more here)		
enrichment		object	Present only when Real Time Bid Enrichment is enabled for given request. See Enrichment object		

IDS Object

The `ids object` provides identifier data for the ID5 ID and any other Extended Identifiers (EIDs) received from ID5.

Each sub-object is named (e.g., `id5id` for the ID5 Universal ID) and contains an OpenRTB-compatible EID structure. Additional third-party identifiers may be included based on enabled services, such as Real-Time Bid Enrichment.

For example, the **ids object** might include:

- `id5id` : The ID5 Universal ID.
- `euid` : European Unified ID.
- `id5-[provider]` Other providers like PubMatic, Magnite, Xandr, TTD, etc.
  - For example: `id5-ttd` : The Trade Desk ID (enriched via ID5)

Each identifier contains an `eid object` that mirrors the structure of OpenRTB's eids field, enabling direct use in bid requests. For enriched IDs, provenance fields such as **mm** (match method), **matcher**, and **inserter** may be added to improve transparency and compliance with IAB best practices.

Name	Required	Type	Description	Format	Example
id5id	x	object	The container for the ID5 Universal ID. For details, see the EID object below.		

EID Object

Name	Required	Type	Description	Format	Example
source	x	string	Source or technology provider responsible for the set of included IDs. Expressed as a top-level domain.		id5-sync.com
uids	x	object array	An array of extended ID UID objects from the given source. Refer to Object: UIDS.		
mm	For enriched IDs	integer	The match method used by the matcher to establish the ID connection (e.g., exact match, probabilistic, or IP-based). This promotes transparency in identity resolution processes. Refer to AdCOM 1.0 Matching Methods	enum	2
matcher	For enriched IDs	string	The domain of the technology provider responsible for matching or bridging the ID, as defined by the mm value.		id5-sync.com

Name	Required	Type	Description	Format	Example
inserter	For enriched IDs	string	The canonical domain name of the entity (e.g., publisher, SSP, or wrapper) that added the EID to the bid request.		publisher-domain.com

UIDS Object

Name	Required	Type	Description	Format	Example
id	x	string	The identifier for the user. For ID5, the value equals the response field: `universal_uid`		
atype	x	integer	Type of user agent the match is from. Refer to AdCOM Agent Types.	enum	1
ext		object	An extension object for passing additional metadata about this specific ID. Further information on ID5 metadata is available here.		

Privacy Object

Name	Required	Type	Description	Format	Example
jurisdiction	x	string	The legal jurisdiction applicable to the request (e.g. "gdpr", "ccpa", etc), based on the location the request was made from	enum	gdpr
id5_consent	x	boolean	Indicates if ID5 had proper consent on the request to process the user's personal data. See below for more information about how ID5 handles requests with and without consent.		

ext Object

Name	Type	Description	Example
linktype	integer	See details here	1

Enrichment Object

Name	Type	Description	Example
enrichment_selected	boolean	False if request was selected to be in enrichment control group, true if it was selected to be enriched (normal group)	false
enriched	boolean	true if request was selected to be enriched AND at least one enriched ID was found and returned for given request. False otherwise	false

Invalid Request Response

Name	Type	Description	Example
code	string	A code that univocally identifies the error	- partner_id_invalid - sha256_length_invalid - request_format_invalid - user_object_invalid - internal_id5_error

Name	Type	Description	Example
message	string	Human-readable message about the error	- unknown partner - failed to parse json POST - missing ua - missing ts - ts older than allowed - invalid ts format - Fetch of ID5 ID disabled. Please contact ID5 to enable it. - Request from a country that is disallowed. Please contact ID5 to enable it.
type	string	A code that identifies the class of error	- validation_error - invalid_request_error - authentication_error

Example Request

```
curl "https://api.id5-sync.com/gc/v1" -X POST -H 'Content-Type: application/json' -d '{
   "ts": "2023-09-19T12:32:34+02:00",
   "partner": 173,
   "ver": "1.0",
   "ip": "4.5.6.7",
   "ua": "Mozilla/5.0 (Linux; Android 11; moto e20 Build/RONS31.267-88-3; wv) AppleWebKit/537.36 (KHTML, like Ged
   "appid": "ABCDE12345.com.example.foo",
   "gpp_sid": "2",
   "gpp_string": "DBAA",
   "accept_language": "de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7",
   "identifiers": [{"id": "bfd378ed-483e-4b11-be1e-12a599c042e0", "type":"DPID"},{"id": "ID-12345", "type":"PPID"}]
}'
```

Example Valid Response

```
{
   "created_at": "2025-11-14T16:12:20.968Z",
   "original_uid": "ID5*KDdqfh7p46U_Eb_xD8-9.WYkX2xsHEQqpJP2YNzKqC7__",
   "privacy": {
      "id5_consent": true,
      "jurisdiction": "gdpr"
   },
   "signature": "ID5_AoPiwhPFUccYOQLCwfbQTEbMTk85rQD10hAZYixkNIZeozXos46w4uqpJqq",
   "universal_uid": "ID5*1Ls3HtzGAlqquMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
   "ext": {
      "linkType": 2
   },
   "ids": {
      "id5id": {
         "eid": {
            "source": "id5-sync.com",
            "uids": [
               {
                  "atype": 1,
                  "ext": {
                     "linkType": 2,
                     "pba": "Fnqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWJP200AngDPxw=="
```

```
            },
            "id": "ID5*1Ls3HtzGAIdqduMjJmL2CcyuGOw9JArVgXc_LS34MC7__"
          }
        ]
      }
    },
    "euid": {
      "eid": {
        "source": "euid.eu",
        "uids": [
          {
            "atype": 3,
            "ext": {
              "provider": "id5-sync.com"
            },
            "id": "E4.AAAAAezwMkyH6T4wjwiMRLR7C7l77M2Vyl8q_pQSzPLdlsiffl86arrL3FKC4T"
          }
        ]
      }
    },
    "id5-ttd": {
      "eid": {
        "inserter": "publisher-domain.com",
        "matcher": "id5-sync.com",
        "mm": 2,
        "source": "adserver.org",
        "uids": [
          {
            "atype": 1,
            "ext": {
              "provider": "id5-sync.com",
              "rtipartner": "TDID"
            },
            "id": "09d89c84-2e59-4c80-9c80-09db5f901234"
          }
        ]
      }
    }
  }
}
```


Example Invalid Response


```
{
 "code": "partner_id_invalid",
 "message": "unknown partner",
 "type": "invalid_request_error"
}
```


ATTENTION

We recommend configuring your API response parsers to ignore any unrecognized fields. This is a common best practice that ensures your parsing logic remains forward-compatible with future changes and can easily handle dynamic fields supported by OpenRTB.

ID5 Signature

The ID5 Signature will be returned on every response from ID5 and contains all "user state" information necessary to support cross-domain reconciliation. As an example, this could include the following pieces of data:

- Original UID value (an encrypted first party ID for this user on this domain/publisher)
- Cookie Birthdate
- Last Seen Timestamp (from this domain/publisher)
- Current ID5 ID value (encrypted)
- Link Type

> *The Signature is used only by ID5 (it will be encrypted with a private key) and must be passed in every subsequent request to ID5.*