Prebid.js UserId Module

What is the ID5 Universal ID?

The ID5 Universal ID is a shared, neutral identifier that publishers and ad tech platforms can use to recognise users even in environments where 3rd party cookies are not available or blocked. ID5 enables publishers to create and distribute a shared 1st party identifier to the entire ecosystem. Ad tech platforms that connect with ID5 can decrypt the Universal ID and improve their user recognition capabilities. The ID5 Universal ID is designed to respect users' privacy choices throughout the advertising value chain.

Registration

The ID5 ID is free to use, but requires a simple registration with us. If you don’t already have an account with ID5, please visit https://id5.io/universal-id to sign up and request your ID5 Partner Number to get started.

Stay up-to-date

We recommend that you sign up for our release notes in order to stay up-to-date with any changes to the implementation of the ID5 Universal ID in Prebid.

How does the ID5 Universal ID work?

By using the ID5 Universal ID, ad tech platforms can eliminate the need to sync their platform-specific IDs with their partners - the equivalent of needing a translator to help two people speaking different languages understand each other. When all platforms are using the Universal ID to transact against, it's like they're all speaking the same, common language. After configuring your Prebid.js setup to pull the ID5 Universal ID, your demand partners configured in Prebid can retrieve the ID and pass it on to their server side RTB partners (typically DSPs). This allows the DSP to target the user, manage frequency and recency capping, and apply additional data, even when cookies are not supported.

What will happen if I turn on the ID5 Universal ID on my page?

By activating the ID5 Universal ID, the following will take place:

1. When a user is on your page, Prebid will make an asynchronous call to ID5 to retrieve the ID5 Universal ID.
2. ID5 returns the ID5 Universal ID to Prebid, which will then store the result in 1st party local storage with an expiration set by your config.
3. On subsequent page loads, if the ID is already present in the browser, there will be no need to call ID5 to retrieve the ID again, so there is minimal latency impact on your page
4. Bid adapters that you have configured can choose to listen to the ID5 Universal ID within their adapter. If they do, they can pass the ID to their server.
5. The bid adapter’s server can then include the ID5 Universal ID in any RTB bid requests as a separate field in the bid request
6. The buyer can then use the ID5 Universal ID to lookup the user in their database and perform any targeting or capping that their campaigns require.

Setup Instructions

Below are step by step instructions for installing and configuring the Prebid.js User ID Module with the ID5 Universal ID. The instructions below assume a basic understanding of building Prebid.js and editing its page-level configuration; for more detailed instructions, getting started guides, and more, please visit the Prebid.org website.

Step 1: Build Prebid.js with the User ID Module

When building Prebid.js, be sure to include both the `userId` and `id5IdSystem` modules, in addition to the other modules you normally include.

```
gulp build –modules=userId,id5IdSystem
```

You may also use the Prebid Download page to build your version of Prebid.js by selecting the `User ID: ID5 ID` module.

Step 2: Configure the User ID Module in the page configuration

Within the `pbjs.setConfig()` function, add the following configuration before you make a request for bids:

```
pbjs.setConfig({
  userSync: {
    userIds: [{
      name: 'id5Id',
      params: {
        partner: 173,            // change to the Partner Number you received from ID5
        pd: 'MT1iNTBjY...'       // optional, see below for a link to how to generate the pd string
        abTesting: {             // optional
          enabled: true,         // false by default
          controlGroupPct: 0.1   // valid values are 0.0 - 1.0 (inclusive)
        }
      },
      storage: {
        type: 'html5',           // 'html5' is the required storage type
        name: 'id5id',           // 'id5id' is the required storage name
        expires: 90,             // storage lasts for 90 days
        refreshInSeconds: 8*3600 // refresh ID every 8 hours to ensure it's fresh
      }
    }],
    auctionDelay: 50             // 50ms maximum auction delay, applies to all userId modules
  }
});
```

ATTENTION

As of Prebid.js v4.13.0, ID5 requires `storage.type` to be `"html5"` and `storage.name` to be `"id5id"`. Using other values will display a warning today, but in an upcoming release, it will prevent the ID5 module from loading. This change is to ensure the ID5 module in Prebid.js interoperates properly with the ID5 API and to reduce the size of publishers' first-party cookies that are sent to their web servers. If you have any questions, please reach out to us at prebid@id5.io.

Configuration Parameters

Name	Required	Type	Description	Example
partner	Required	Number	This is the ID5 Partner Number obtained from registering with ID5.	`173`
pd	Optional	String	Publisher-supplied data used for linking ID5 IDs across domains. See below for details on generating the string. Omit the parameter or leave as an empty string if no data to supply	`'MT1iNTBjY...'`
abTesting	Optional	Object	Allows publishers to easily run an A/B Test. If enabled and the user is in the Control Group, the ID5 ID will NOT be exposed to bid adapters for that request. See below for more details.	Disabled by default
abTesting.enabled	Optional	Boolean	Set this to `true` to turn on this feature	`true` or `false`
abTesting.controlGroupPct	Optional	Number	Must be a number between `0.0` and `1.0` (inclusive) and is used to determine the percentage of requests that fall into the control group (and thus not exposing the ID5 ID). For example, a value of `0.20` will result in 20% of requests without an ID5 ID and 80% with an ID.	`0.10`

Generating the Publisher Data (pd) String

The `pd` string is available in version 3.25.0 of Prebid.js and later

The `pd` field (short for Publisher Data) is an *optional*, base64 encoded string that contains any deterministic user data the publisher has access to. The data will be used strictly to provide better linking of ID5 IDs across domains for improved user identification. If the user has not provided ID5 with a legal basis to process data, the information sent to ID5 will be ignored and neither used nor saved for future requests.

If the publisher does not have any Publisher Data to pass to ID5, the `pd` parameter can be omitted or left with an empty string value (`pd: ""`).

The possible keys in the string are:

Key	Value
0	other
1	sha256 hashed email
2	sha256 hashed phone number

3	cross-domain publisher user_id value
4	cross-domain publisher user_id source (value provided by ID5)
5	publisher user_id value

PD Example

To illustrate how to generate the pd string, let's use an example. Suppose you have an email address for the user, in this example it is `myuser@domain.com`, and want to share it with ID5 to strengthen the value of the UID we respond with. You also have your own user id for this user that you can share: `ABC123`.

First, perform a sha256 hash of the email, resulting in a string `b50ca08271795a8e7e4012813f23d505193d75c0f2e2bb99baa63aa822f66ed3`

Next, create the raw pd string containing the keys `1` (for the hashed email) and `5` (for the publisher user id), separated by &'s (the order doesn't matter):

```
1=b50ca08271795a8e7e4012813f23d505193d75c0f2e2bb99baa63aa822f66ed3&5=ABC123
```

Finally, base64 the entire raw pd string, resulting in the final pd value:

```
MT1iNTBjYTA4MjcxNzk1YThlN2U0MDEyODEzZjIzZDUwNTE5M2Q3NWMwZjJlMmJiOTliYWE2M2FhODIyZjY2ZWQzJjU9QUJDMTIz
```

This is the value you will add to the config when you initialize the ID5 userId module:

```
...
params: {
        partner: 173,   // change to the Partner Number you received from ID5
        pd:
'MT1iNTBjYTA4MjcxNzk1YThlN2U0MDEyODEzZjIzZDUwNTE5M2Q3NWMwZjJlMmJiOTliYWE2M2FhODIyZjY2ZWQzJjU9QUJDMTIz'
},
...
```

A/B Testing

The `abTesting` feature is available in version 4.20.0 of Prebid.js and later

Publishers may want to test the value of the ID5 ID with their downstream partners. While there are various ways to do this, A/B testing is a standard approach. Instead of publishers manually enabling or disabling the ID5 User ID Module based on their control group settings (which leads to fewer calls to ID5, reducing our ability to recognize the user), we have baked this in to our module directly.

To turn on A/B Testing, simply edit the configuration (see details above) to enable it and set what percentage of requests you would like to set for the control group. The control group is the set of requests where an ID5 ID will not be exposed in to bid adapters or in the various user id functions available on the `pbjs` global. An additional value of `ext.abTestingControlGroup` will be set to `true` or `false` that can be used to inform reporting systems that the request was in the control group or not. It's important to note that the control group is *request* based, and not *user* based. In other words, from one page view to another, a user may be in or out of the control group.

GDPR Support

The ID5 ID is a privacy-by-design implementation of a shared ID and fully supports the GDPR. When the ID5 ID is requested by Prebid in a GDPR-relevant country, ID5 will ensure the user has consented to processing by ID5 for the “Information storage and access” purpose (Purpose 1). If not, ID5 will not attempt to read or write our 3P cookie.

To enable GDPR support within Prebid, you will need to include the GDPR Consent Management module when you build Prebid:

```
gulp build –modules=userId,id5IdSystem,consentManagement
```

You will also need to ensure you add the appropriate configuration to your `setConfig()` function to include `consentManagement`:

```
pbjs.setConfig({
  userSync: {
    userIds: [{
      name: 'id5Id',
      params: {
        partner: 173,            // change to the Partner Number you received from ID5
        pd: 'MT1iNTBjY...'       // optional, see below for a link to how to generate the pd string
        abTesting: {             // optional
          enabled: true,         // false by default
          controlGroupPct: 0.1   // valid values are 0.0 - 1.0 (inclusive)
        }
      },
      storage: {
        type: 'html5',           // 'html5' is the required storage type
        name: 'id5id',           // 'id5id' is the required storage name
        expires: 90,             // storage lasts for 90 days
        refreshInSeconds: 8*3600 // refresh ID every 8 hours to ensure it's fresh
      }
    }],
    auctionDelay: 50             // 50ms maximum auction delay, applies to all userId modules
  },

  consentManagement: {
          cmpApi: 'iab',
    timeout: 8000
  }
});
```

A Note About ID5 ID Encryption

The ID5 Universal ID that is delivered to Prebid will be encrypted by ID5 with a rotating key to avoid unauthorized usage and to enforce privacy requirements. Only platforms that have the necessary legal basis to process user data will be able to decrypt the ID and use it for targeting, frequency capping, measurement, etc. Therefore, we strongly recommend setting `storage.refreshInSeconds` to 8 hours (`8*3600` seconds) to ensure all demand partners receive an ID that has been encrypted with the latest key, has up-to-date privacy signals, and allows them to transact against it.