AppNexus Console / ... / Pass in User Data Offline

Added by Mike Nolet, last edited by Rachel Sadres on Oct 15, 2013

User ID Mapping with getUID and mapUID

You may have internal offline data about users that you need to insert into the AppNexus Server Side Cookie Store so that it is available for decisioning. If so, we must first map your internal user IDs and AppNexus user IDs in order to be able to match user IDs on future requests. Depending on whether you will store the mapping in your system or in ours, there are two ways to do this through pixel calls: mapUID and getUID.

The **getUID** service retrieves the AppNexus ID so you can coordinate it with your own in-house ID server side or in your own cookie space. Then you can pass in an offline data feed that says, "update AppNexus user ABC with the following segment data." The **mapUID** service passes your internal ID to us for mapping to the AppNexus ID within the AppNexus cookie store.

Note that the AppNexus user ID macro described below is `$UID`. Unlike many of our other macros, it does not have curly braces.

On This Page

A Basic Mapping Example

Prostarr User ID	PS123
AppNexus User ID	2894234234
Shoe buyer Segment ID	0005

If the advertiser Prostarr tells us that user `PS123` is in segment `0005`, this will mean nothing to us. Instead, they have to tell us that user `2894234234` is in segment `0005`.

To do this, Prostarr has done a previous mapping of all AppNexus and Prostarr Sportswear user IDs using the GetUID service and stored that mapping in its internal database. A GetUID call looks like

```
http://ib.adnxs.com/getuid?URL
```

where the *URL* is the the one you want the mapping directed to. This pixel is placed on Prostarr's webpages, and when it fires, the Impression Bus inserts the AppNexus user ID into the URL by replacing the `$UID` macro.

For a detailed walk-through of how this works, see the User Mapping Visual.

The getUID Service

A URL of your choice will receive a client-side call with the AppNexus user ID as a querystring parameter. During the ad call, AppNexus redirects the user to the member's desired URL. The Impression Bus inserts the AppNexus user ID into the URL, either by replacing a `$UID` macro, or by appending it to the end of the URL. When the user does not have an AppNexus cookie "0" is returned and AppNexus attempts to create one.

Code

The format of the getUID Service is fairly simple:

```
http://ib.adnxs.com/getuid?URL
```

The secure version uses this format:

```
https://secure.adnxs.com/getuid?URL
```

Users with non-sticky cookies or users who have opted out will be redirected with a user ID of 0. If you would prefer not to receive user IDs of 0, use this format:

```
http://ib.adnxs.com/getuidnb?URL
```

If you need the URL to remain URI-encoded, use this format:

```
http://ib.adnxs.com/getuidu?URL
```

A getUID example

In most cases, it will be necessary to pass the user ID into a specified parameter within the query string of the URL. To accomplish this, you may use the `$UID` macro:

```
http://ib.adnxs.com/getuid?http://ad.adserver.com/pixel?
adnxs_uid=$UID&geo=35&referrer=http://www.website.com
```

The `$UID` macro will be replaced with the AppNexus user ID by the Impression Bus and the user will be directed to the following URL:

```
http://ad.adserver.com/pixel?adnxs_uid=19241908471992&geo=35&referrer=http://www.website.com
```

Use case: piggyback on a pixel call

An easy way to create this mapping is to piggyback the getUID Service off of one of your existing pixel calls. You can adjust your pixel so that when the pixel call is made, there will be an immediate redirect to the AppNexus getUID service, which will pass back the user ID within the URL indicated by the AppNexus Console.

For example, let's assume the following pixel exists on a publisher site today:

```
<img src="http://ad.adserver.com/pixel?id=234123" width=1 height=1/>
```

Instead, fire the AppNexus getuid pixel, providing the URL of the adserver.com pixel:

```
http://ib.adnxs.com/getuid?http://ad.adserver.com/pixel?id=234123&adnxs_uid=$UID
```

The getUID service redirects to the specified URL with the AppNexus user ID in place of the `$UID` macro:

```
http://ad.adserver.com/pixel?id=234123&adnxs_uid=19241908471992
```

When `adserver.com` receives this call, the AppNexus user ID can be inserted into `adserver.com`'s server-side user data store where this mapping of user IDs can be used by future impression and pixel calls through the AppNexus Impression Bus.

The mapUID Service

> The mapUID service has fees associated with it. If you're interested in using this option, please speak with your AppNexus account manager.

The **mapUID** Service inserts your in-house ID into the AppNexus Server Side Cookie Store.

The format for a client-side pixel that maps your ID to ours in our system is

```
http://ib.adnxs.com/mapuid?member=MEMBER_ID&user=USER_ID
```

where the member ID is the ID assigned to your company by AppNexus and the `USER_ID` is your company's user ID. If you don't know your member ID, you can find out by making a `GET` request to the Member Service.

Expiration of mappings

The average time to live (TTL) for mapUID mappings is around 2.5 weeks. Therefore it's very important that you fire the mapUID pixel as frequently as possible and on as many pages as possible to keep your mappings live.

Piggybacking a segment call off of a mapUID call

You can add users to one or more segments in the AppNexus cookie store at the same time that you map their ID to ours.

Here is the syntax:

```
http://ib.adnxs.com/mapuid?member=[MEMBER_ID]&user=[USER_ID]&seg=[SEG_ID],[SEG_ID],[SEG_ID]

http://ib.adnxs.com/mapuid?member=123&user=12abc34565&seg=1,2,3
```

To pass in values for each segment use colons. (This example used segment codes instead of IDs.)

```
http://ib.adnxs.com/mapuid?member=123&user=[USER_ID]&seg_code=[code1]:[code1value],[code2]:[code2value],[code3]:[code3value]
```

Sending us your segment data offline

For more information about how to send us your segment data offline, see the API's Batch Segment Service.

Related Topics

- User ID Syncing
- Asynchronous Usersync Pixels

Was This Page Helpful?

Yes No

We're glad you liked it. Can you tell us what you found helpful?

What should we know about this page?

- □ Innacurate info
- □ Needs more examples
- □ Confusingly written / difficult to understand
- □ Not the topic I was looking for
- □ Not detailed enough
- □ Broken links or other technical problems

How can we improve this page?

Send Feedback

Thank you.

Your feedback helps us improve AppNexus documentation.

Like