//pixfuture

PixFuture Header Bidding with DFP Documentation

Step One

Install prebid.js (must be version 0.14.0 or later) on your website. For details please follow prebid.js dev docs (http://prebid.org/dev-docs/getting-started.html) and atop docs (http://prebid.org/adops.html).

Note: remember to add "openx" adapter to your prebid.js file.

Step Two

Provide answers to the following questions to your PixFuture representative:

a. What bid buckets do you plan to use?

b. Which ad unit sizes do you plan to use?

c. Are you sending all keys via the pbjs.enableSendAllBids() function?

Step Three

Add the unit and delDomain values provided by your PixFuture representative.

```
//Register PixFuture bidder tag ID
pbjs.que.push(function() {
  var adUnits = [{
    code: 'div-gpt-ad-1438287399331-0',
    sizes: [[300, 250]],     //Include all the ad sizes in this ad div
    bids: [{
      bidder: 'openx',
            params: {
        unit: '538xxxxxx',         //Change to your unit id
        delDomain: 'pixfuture-d.openx.net'
      }
    }]
}];
```

Add the following code inside the pbjs.bidderSettings code to allow PixFuture to pass key-value pairs to DFP. See the Prebid.js documentation (https://github.com/prebid/Prebid.js/blob/master/integrationExamples/gpt/pbjs_example_gpt.html#L340) for details. If standard PixFuture bid buckets are used, please refer to the Standard PixFuture bid bucket code (see next page).

```
openx: {
  alwaysUseBid: true,
  adserverTargeting: [{
    key: "oxb",
    val: function(bidResponse) {
      var bid;
      if (bidResponse.cpm < 20) {
        //Penny Bid Buckets
        bid = ((Math.floor(bidResponse.cpm * 100) / 100) * 100).toFixed(0);
      } else {
        //Twenty dollar bucket
        bid = "2000";
      }
      //Returns creativeSize_bidBucket as the value
      return bidResponse.width + "x" bidResponse.height + "_" + bid;
    }
  }]
}
```

If all keys are not being sent via the pbjs.enableSendAllBids() function, please refer to Additional Metrics.

Standard PixFuture bid bucket code

The following code should be used with standard PixFuture bid buckets.

```
openx: {
  alwaysUseBid: true,
  adserverTargeting: [{
    key: "oxb",
    val: function(bidResponse) {
      var bid;
      if (bidResponse.cpm < 1) {
        bid = ((Math.floor(bidResponse.cpm * 20) / 20) * 100).toFixed(0);
      } else if (bidResponse.cpm < 5) {
        bid = ((Math.floor(bidResponse.cpm * 10) / 10) * 100).toFixed(0);
      } else if (bidResponse.cpm < 20) {
        bid = ((Math.floor(bidResponse.cpm * 2) / 2) * 100).toFixed(0);
      } else {
        bid = "2000";
      }
      return bidResponse.width + "x" + bidResponse.height + "_" + bid;
    }
  }]
}
```

Additional Metrics

If the pbjs.enableSendAllBids() function is not being used to send all keys, use the following steps to add the ox_pb_won key to the prebid.js.

Add the following code to the standard section inside the pbjs.bidderSettings code.

```
{
  key: "ox_pb_won",
  val: function (bidResponse) {
    if (bidResponse.bidderCode.indexOf('openx') !== -1) {
      return "true";
    }
    return "false";
  }
}
```

The following example illustrates the placement within the prebid.js.

```
pbjs.bidderSettings = {
  standard: {
    alwaysUseBid: false,
    adserverTargeting: [{
      key: "hb_bidder",
      val: function(bidResponse) {
        return bidResponse.bidderCode;
      }
    }, {
      key: "hb_adid",
      val: function(bidResponse) {
        return bidResponse.adId;
      }
    }, {
      key: "hb_pb",
      val: function(bidResponse) {
        return bidResponse.pbMg;
      }
    }, {
      key: "hb_size",
      val: function(bidResponse) {
        return bidResponse.size;
      }
    }, {
      key: "ox_pb_won",
      val: function(bidResponse) {
```

```
        if (bidResponse.bidderCode.indexOf('openx') !== -1) {
          return "true";
        }
        return "false";
      }
    }]
  }
}
```

Full Code Example

https://www.pixfuture.com/tools/pbjs_example.html?pbjs_debug=true

Best practices

Ensure the number of ad slots requested in the prebid.js code matches the number of ads on the page.

Instead of reading the slots on the page, Prebid.js solicits bid requests based on the bidders and slots configured. So, if you have 5 DFP ads on the page, you should request 5 ads in your prebid.js config.

Make sure that your DFP line items match the bucketing in your prebid.js code.

If you have line items in DFP at $0.10 increments up to $20, make sure that the code on the page rounds into $0.10 buckets and stops at $20.

Create key values for hb_bidder inside DFP even if you are using only Top Bid.

If you have more than one header bidding partner, using only the hb_pb key to target a single Prebid order does not allow you to pull reports on the performance of each partner. By creating the hb_bidder key and individual key values for each header bidding partner, you will be able to analyze the performance of your header bidding partners.

You can create the hb_bidder key and values using the steps below:

1. Login into DFP
2. Click on Inventory Tab
3. Create Key
4. Enter hb_bidder
5. Create values for each of your header bidding partners

Limit your header bidding partner participation to 3-4 at a time

More header bidder partners does not equal higher revenue. Due to increased page load and browser overhead, too many header bidding partners can potentially cause revenue loss due to the extra processing.