

IPONWEB

ENGINEERING MEDIA TRADING EVOLUTION

BID SWITCH

RTB Specification

Version: 2.1
Date: 23-Apr-2014

Provision of this Specification does not guarantee connection to the BidSwitch Platform and/or to any Supply Side Partners and/or any Demand Side Partners that may be connected to the BidSwitch Platform from time to time.

Table of Contents

Introduction.....	4
Change history.....	4
User Matching.....	5
DSP Initiated User Matching.....	5
Bid Switch Initiated User Matching.....	6
Bid Request.....	7
Bid Request Object.....	7
Impression Object.....	8
Impression Ext Object.....	9
Impression Ext Rubicon Object.....	9
Banner Object.....	9
Video Object.....	10
Video Ext Object.....	11
Device Object.....	11
Geo Object.....	12
User Object.....	12
Site Object.....	12
App Object.....	13
Private Marketplace Object.....	13
Deal Object.....	14
Publisher Object.....	14
Ext Object.....	14
SSP Specific Fields.....	15
Google Object.....	15
Detected_vertical Object.....	15
Adtruth Object.....	15
Examples.....	16
Simple Banner Ad.....	16
Private Deal.....	17
Video Ad.....	19
Bid Response.....	20
Top Level Bid Response Object.....	20
Seat Bid Object.....	21
Bid Object.....	21
Ext Object.....	22
SSP Specific Fields.....	23
LiveIntent Object.....	23
Google Object.....	23
Nobid Response.....	24
Examples.....	24
Simple Bid Response.....	24
No Bid Example.....	25

Response to Private Bid Request.....	25
Macros.....	25
Macros may be used in nurl and ext.js_url fields.	25
Win price macro	25
SSP click tracking url macro	26
Creative Preview.....	26
Additional Features	27
QPS Management.....	27
Bid Targeting	27
User Sync Targeting.....	27

Introduction

Bid Switch is an innovative technology developed at IPONWEB to simplify integration of Demand Side Platforms (DSP) with Sell Side Platforms (SSP). The key idea behind the service is to enable fast and cheap SSP integration without many rounds of specification refinements and development between DSP and SSP. Bid Switch acts like an intermediate between DSP and SSP which adapts custom SSP protocols to OpenRTB 2.2 RC standards. The present specification defines the protocol for interaction between DSP and Bid Switch.

Change history

v1.0 - 1.03	Initial version
v1.04	Added support of bcat, batrr and badv fields, ext.openx section containing pub_blocked_content field.
v1.05	Updated wording of bid response bid.url field description. Added bid request ext.aol section containing denied_categories field.
v1.06	Updated wording of fields descriptions. Added bid response ext.agency_name and ext.advertiser_name fields.
v1.07	Added bid request imp.bidfloor field. Added support for HTTP 204 as a valid nobid response.
v 1.08	Added bid request site.cat field.
v 1.09	Added SSP click macro support. Added support for video traffic. Added bid request PMP section. Deprecated ssp-specific bid request fields: ext.deals, ext.rubicon, ext.aol, ext.openx.
v 1.10	Changed video.protocol type to array of integer, fixed video request example. Added bid request ext.is_secure field. Added bid response ext.language and ext.cat fields.
v 1.11	Added ext.google.detected_vertical, imp.video.ext, imp.video.companionad bid request objects. Added bid request fields: imp.video.companiontype, wseat, imp.banner.btype, ext.adtruth. Added bid response fields: seatbid.seat, seatbid.bid.ext.vast_url. Updated bid response seatbid.bid.url field description for video traffic.
v 2.0	Added application traffic support.

	<p>Bid request site object is optional now.</p> <p>Added ext.google.excluded_attribute and ext.google.allowed_vendor_type bid request fields.</p> <p>Added seatbid.bid.ext.google bid response object.</p> <p>Added seatbid.bid.cid bid response field.</p> <p>Removed fields and objects deprecated in v 1.9.</p>
v 2.1	<p>Added seatbid.bid.ext.js_url bid response field. The field is required for mobile application inventory.</p> <p>Bid response seatbid.bid.nurl field is optional now.</p> <p>Added pmp.deals.wseat, imp.banner.expdir and “at” bid request fields.</p>

User Matching

Bid Switch has an integrated user matching functionality to simplify user sync operations for DSP. User matching information is managed within Bid Switch database, so there is no need for DSP to implement user matching logic to store the mappings.

While making bid request to DSP, Bid Switch will send both DSP user ID and Bid Switch user ID in *buyeruid* and *id* fields of User Object respectively (see Bid Request). If DSP user ID is not available, only Bid Switch user ID will be sent. For these purposes Bid Switch needs to update User Object in bid request while proxying it from an exchange to the DSP.

User synchronization can be initiated either by DSP or by Bid Switch itself. For both synchronization types Bid Switch supports secure and insecure communications with DSP using HTTP and HTTPS protocols.

DSP Initiated User Matching

To sync a particular user with Bid Switch, the DSP should initiate the user to call for 1x1.gif image by the following GET URL:

<http://x.bidswitch.net/sync>

providing the following HTTP request parameters:

Param	Value Type	Default	Example	Description
dsp_id	string	-	123	ID assigned to DSP by Bid Switch.
user_id	string	-	1234567890	User ID in DSP system.

expires*	integer	90	30	Expiration time in days for user matching. Should be less or equal to 365.
----------	---------	----	----	--

* This field is optional.

Bid Switch matching URL also supports secure HTTPS connections.

There is an option to exclude a particular user from syncing for a certain time interval. To make it happen DSP can perform 302 redirect to the matching URL leaving *user_id* parameter blank.

Example 1. DSP initiated user matching:

`https://x.bidswitch.net/sync?dsp_id=123&user_id=1234567890&expires=30`

Example 2. User opt out:

Redirecting a user to the following URL will not sync this specific user for the next 5 days and will consider bids from this user as user-unmatched for the given DSP:

`http://x.bidswitch.net/sync?dsp_id=123&user_id=&expires=5`

Bid Switch Initiated User Matching

BidSwitch will also be able to initiate user synchronization. The partner will issue Bid Switch a matching URL that will redirect to the pixel URL described in DSP Initiated User Matching, with properly populated *dsp_id*, *user_id* and *expires* parameters. Note that this URL should be for a pixel that performs a 302 redirect, rather than for an iframe with a script, etc. within it. The only redirect from the DSP's matching URL should be to the Bid Switch pixel.

The matching redirect URL, issued by a DSP should contain `${SSP}` macro. This macro will be substituted with the SSP ID on Bid Switch side before sending a user to the partner. The value of the macro should be sent back to Bid Switch in the *ssp* parameter. The example of the matching URL, provided by partner is shown below:

`http://www.dsp-example.com/bsw_sync?bidswitch_ssp_id=${SSP}`

Here, *bidswitch_ssp_id* parameter name can be changed to any other name chosen by DSP. On receiving call to the URL above the partner should respond with 302 redirect to Bid Switch matching URL with valid *dsp_id*, *user_id* and *expires* parameters as described in DSP Initiated User Matching and providing one additional *ssp* parameter with the value of `${SSP}` macro. So given a call to the partner matching URL of the form

`http://www.dsp-example.com/bsw_sync?bidswitch_ssp_id=rubicon`

DSP server should respond with 302 redirect to Bid Switch matching URL with the following sample *dsp_id*, *user_id* and *expires* parameters:

```
http://x.bidswitch.net/sync?dsp_id=123&user_id=123&expires=5&ssp=rubicon
```

In addition to HTTP protocol the matching redirect URL, provided by the partner should also support secure HTTPS connections. If Bid Switch initiate a user call using HTTPS protocol the partner should redirect the user to Bid Switch matching URL using secure HTTPS connection.

After Bid Switch initiated user matching Bid Switch will associate the value of *user_id* parameter with the user's browser and make it available in any later bid requests from this user's browser. This will be sent in the *buyeruid* field of the JSON bid request. The *user_id* may include more than just the user ID, however it should not exceed 36 characters.

Bid Request

Bid Request Object

This is the top level object that is being sent to DSP. Each bid request sent from Bid Switch to DSP will contain the following fields.

Field	Type	Description
id	string	Unique ID of the bid request, provided by the exchange.
imp	array of objects	Array of Impression Objects.
device	object	Device Object which specifies the device the ad impression will be delivered to.
user	object	User Object which describes the user.
at*	integer	Auction Type. If "1", then first price auction. If "2", then second price auction.
site*	object	Site Object describing the site. Either site or app object should be present.
app*	object	App Object describing the mobile application. Either site or app object should be present.
bcat*	array of strings	Blocked Advertiser Categories, using IAB taxonomy, e.g. {"IAB10-1", "IAB25"}

badv*	array of strings	Array of strings of blocked top-level domains of advertisers. For example, ["company1.com", "company2.com"].
wseat*	array of strings	Array of buyer seats allowed to bid on this auction. Seats are an optional feature of exchange. For example, ["4", "34", "82", "A45"] indicates that only advertisers using these exchange seats are allowed to bid on the impressions in this auction. If this field is present in the bid request then the bid response must include seatbid.seat field containing the id of the buyer that is bidding.
cur	array of strings	Array of allowed currencies for bids on this bid request using ISO-4217 alphabetic codes. Actually always contains exactly one currency.
pmp*	object	Private Marketplace object.
ext	object	Ext Object used for SSP specific properties.

* Fields marked with asterisk are optional and may not be sent in each request.

Impression Object

Field	Type	Description
id	string	Unique ID of the impression being shown.
banner*	object	Banner Object describing the ad properties. Required for banner impressions.
video*	object	Video Object describing the ad properties. Required for video impressions.
bidfloor*	float	Bid floor in CPM as set by the SSP. The currency is the same as specified in cur field.
ext	object	Impression Ext Object

* Fields marked with asterisk are optional and may not be sent in each request.

Impression Ext Object

Field	Type	Description
rubicon*	object	Impression Ext Rubicon Object.

* Fields marked with asterisk are optional and may not be sent in each request.

Impression Ext Rubicon Object

Field	Type	Description
site_size_session_count*	integer	User session counter for this site/size. A "user session" here is 24 hours.

* Fields marked with asterisk are optional and may not be sent in each request.

Banner Object

Field	Type	Description
w	integer	Width of the impression in pixels.
h	integer	Height of the impression in pixels.
battr*	array of integers	Blocked creative attributes as defined in OpenRTB.
btype*	array of integers	Blocked creative types as defined in OpenRTB.
pos*	integer	Ad Position as defined in OpenRTB.
topframe*	integer	If the banner is delivered in the top frame this parameter is "1", if there's an iframe it will be omitted.
expdir*	array of integer	Possible expansion directions for an expandable ad: 1: Left 2: Right 3: Up 4: Down 5: Fullscreen

* Fields marked with asterisk are optional and may not be sent in each request.

Video Object

Field	Type	Description
mimes	array of strings	Content MIME types supported.
linearity	integer	1: Linear/In-stream 2: Non-Linear/Overlay
minduration	integer	Minimum video ad duration in seconds
maxduration	integer	Maximum video ad duration in seconds
protocol	array of integer	Accepted video bid response protocols as defined in OpenRTB.
w*	integer	Width of the player in pixels.
h*	integer	Height of the player in pixels.
startdelay*	integer	Indicates the start delay in seconds or generic values below: 0: Pre-roll -1: Generic mid-roll -2: Generic post-roll
battr*	array of integers	Blocked creative attributes as defined in OpenRTB.
minbitrate*	integer	Minimum bit rate in Kbps.
maxbitrate*	integer	Maximum bit rate in Kbps.
api*	array of integers	List of supported API frameworks for this banner as defined in OpenRTB.
companionad*	array of objects	If companion ads are available, they can be listed as an array of banner objects. See Banner Object.
companiontype*	array of integers	List of allowed companion ad types. Possible values: 1: Static Resource 2: HTML Resource 3: iframe Resource
ext*	object	Video Ext Object

* Fields marked with asterisk are optional and may not be sent in each request.

Video Ext Object

Field	Type	Description
skippable*	integer	Indicates whether the video ad may be skippable (i.e. contain a skip button). The values are: 0: allow skippable 1: require skippable 2: deny skippable

* Fields marked with asterisk are optional and may not be sent in each request.

Device Object

Field	Type	Description
ip	string	IPv4 address closest to the device. Bids containing invalid IP addresses will be discarded.
geo	object	Geo Object as derived from the device's location services.
ua*	string	Browser or application user agent string.
language*	string	Alpha-2/ISO 639-1 code of browser language.
carrier*	string	Carrier or ISP derived from the IP address.
connectiontype*	integer	Connection type as defined in OpenRTB.
didsha1*	string	SHA1 hashed device ID.
didmd5*	string	MD5 hashed device ID.
dpidsha1*	string	SHA1 hashed platform-specific ID.
dpidmd5*	string	MD5 hashed platform-specific ID.
ifa*	string	Native identifier for advertisers; an opaque ID assigned by the device or browser for use as an advertising identifier (e.g. Apple's IFA, Android's Advertising ID, etc).

* Fields marked with asterisk are optional and may not be sent in each request.

Geo Object

The information provided in Geo Object is based on MaxMind database.

Field	Type	Description
lat*	float	Latitude from -90 to 90. South is negative.
lon*	float	Longitude from -180 to 180. West is negative.
country*	string	Country using ISO-3166-1 Alpha-2.
region*	string	Region using ISO-3166-2 or FIPS region codes.
city*	string	City name as provided by MaxMind.
zip*	string	Zip/postal code.

* Fields marked with asterisk are optional and may not be sent in each request.

User Object

Field	Type	Description
id*	string	Unique customer ID of this user on the exchange. It may not be present if DSP opted for unknown users.
buyeruid*	string	DSP user ID for this user as mapped by Bid Switch for the DSP.

* Fields marked with asterisk are optional and may not be sent in each request.

Site Object

Field	Type	Description
id*	string	Site ID on the exchange prefixed with exchange name followed by underscore. E.g. if Rubicon SSP sends bid request for site ID "123", the field becomes "rubicon_123".
domain*	string	Domain of the site, used for advertiser side blocking. For example, "foo.com".
cat*	array of strings	Array of IAB content categories for the publisher site.

page*	string	URL of the page where the impression will be shown.
publisher	object	Publisher Object.

* Fields marked with asterisk are optional and may not be sent in each request.

App Object

Field	Type	Description
id*	string	Application ID on the exchange prefixed with exchange name followed by underscore. E.g. if Rubicon SSP sends bid request for site ID “123”, the field becomes “rubicon_123”.
domain*	string	Domain of the application, for example, “mygame.foo.com”.
cat*	array of strings	Array of IAB content categories for the application.
bundle*	string	Application bundle or package name (e.g., com.foo.mygame).
publisher	object	Publisher Object.
storeurl*	string	App store’s URL for the mobile application.
ver*	string	Application version.

* Fields marked with asterisk are optional and may not be sent in each request.

Private Marketplace Object

Field	Type	Description
private_auction	integer	A value of 1 indicates that only bids submitted inside pmp.deals will take part in the auction. A value of 0 indicates that bids without deal information may also be considered for serving.
deals	array of objects	Array of Deal objects.

Deal Object

Field	Type	Description
id	string	Deal id.
wseat*	array of string	Array of buyer seats allowed to bid on this Direct Deal. Seats are an optional feature of an exchange. For example, ["4", "34", "82", "45"] indicates that only advertisers using these exchange seats are allowed to bid on this direct deal.
bidfloor*	float	Deal price. If deal.at = 1 then this is the exact price of the deal, if deal.at = 2 then this is the bid floor of the deal.
at	integer	Auction type. '1' for fixed price deal, '2' for second price auction.

* Fields marked with asterisk are optional and may not be sent in each request.

Publisher Object

Field	Type	Description
id	string	Publisher ID prefixed with SSP name followed by underscore, e.g. if Rubicon passes publisher ID 25, then the present field will be equal to "rubicon_25". For Google AdX bid requests the Google seller network ID is used.

Ext Object

Field	Type	Description
ssp	string	SSP identification string. E.g., "rubicon" or "openx".
is_secure*	integer	0 for non-secure pages; 1 for secure pages. Creatives for secure pages should be SSL-compliant.
google*	object	Google object. This field is optional and is used only for integrations with Google AdX.

adtruth*	object	Adtruth object. This field is optional and is present if the corresponding data is provided by the SSP.
----------	--------	---

* Fields marked with asterisk are optional and may not be sent in each request.

SSP Specific Fields

Google Object

Field	Type	Description
detected_vertical*	array of objects	List of detected vertical values as defined by Google.
excluded_attribute*	array of integer	List of excluded creative attributes as defined by Google.
allowed_vendor_type*	array of integer	List of allowed vendor types as defined by Google.

* Fields marked with asterisk are optional and may not be sent in each request.

Detected_vertical Object

Field	Type	Description
id	integer	The vertical id as defined by AdX docs.
weight	float	Weight for this vertical, in the (0.0, 1.0] range. More relevant verticals have higher weights.

Adtruth Object

Field	Type	Description
tdl_millis*	int64	Time Difference Linking, which allows for differentiation between similarly-configured devices that might share a DeviceInsight identifier.

RECIPE VERSION_x	string	The specific AdTruth device identification recipe version and the corresponding DeviceInsight identifier. There might be multiple AdTruth recipe versions. E.g. "WEB_APP_BRIDGE_4_0": "4FD87B97751E9C305FD5314A665D2AF1C8FAA2D3"
------------------	--------	--

Examples

Simple Banner Ad

Following is a basic example of a bid request for a banner ad.

```
{
  "id": "1234534625253",
  "imp": [ {
    "id": "1",
    "banner": {
      "w": 300,
      "h": 250,
      "pos": 1,
      "topframe": 0
    },
    "ext": {
      "rubicon": {
        "site_size_session_count": 5
      }
    }
  } ],
  "site": {
    "id": "234563",
    "domain": "siteabcd.com",
    "page": "http://siteabcd.com/page.htm",
    "publisher": {
      "id": "rubicon_25"
    }
  },
  "device": {
    "ip": "64.124.253.1",
    "geo": {
```

```

        "lat": 33.94940185546875,
        "lon": -116.97750091552734,
        "county": "US",
        "region": "NY",
        "city": "White Plains",
        "zip": "10601"
    },
    "ua": "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2.16)
    Gecko/20110319 Firefox/3.6.16",
    "language": "en"
},
"user": {
    "id": "45asdf987656789adfad4678rew656789",
    "buyeruid": "1234567890"
},
"cur": ["USD"],
"at": 2,
"ext": {
    "ssp": "rubicon",
    "adtruth": {
        "tdl_millis": 19534993,
        "WEB_APP_BRIDGE_3_2": "FD5314A665D2AF1C8FAA2D34FD87B977
        51E9C305",
        "WEB_APP_BRIDGE_4_0": "4FD87B97751E9C305FD5314A665D2AF1C
        8FAA2D3"
    }
}
}
}
}

```

Private Deal

Following is a basic example of bid request for private auction.

```

{
    "id": "1234534625254",
    "imp": [ {
        "id": "1",
        "bidfloor": 0.4260085523128510,
        "banner": {
            "w": 300,

```

```

        "h": 250,
        "pos": 1,
        "topframe": 0,
        "expdir": [1, 3]
    }
}],
"site": {
    "id": "234563",
    "domain": "siteabcd.com",
    "page": "https://siteabcd.com/page.htm",
    "publisher": {
        "id": "rubicon_25"
    }
},
"device": {
    "ip": "64.124.253.1",
    "geo": {
        "county": "US",
        "region": "NY",
        "city": "White Plains",
        "zip": "10601"
    },
    "ua": "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2.16)
    Gecko/20110319 Firefox/3.6.16",
    "language": "en"
},
"user": {
    "id": "45asdf987656789adfad4678rew656789"
},
"cur": ["USD"],
"ext": {
    "ssp": "rubicon",
    "deals": ["deal-1", "deal-2"],
    "is_secure": 1,
    "rubicon": {
        "denied_ad_types": [14011, 14006]
    }
}
}
"pmp": {
    "private_auction": 0
    "deals": [
        {
            "id": "deal-1",
            "wseat": ["4", "15"],

```

```

        "bidfloor": 2.5
        "at": 1
    },
    {
        "id": "deal-2",
        "bidfloor": 2.0
        "at": 2
    }
]
}
}

```

Video Ad

Following is a basic example of a bid request for a banner ad.

```

{
  "id": "1234534625253",
  "imp": [ {
    "id": "1",
    "video": {
      "mimes": [
        "video/x-flv",
        "video/mp4",
        "application/x-shockwave-flash",
        "application/javascript"
      ],
      "linearity": 1,
      "minduration": 5,
      "maxduration": 30,
      "protocol": [2, 3, 5, 6],
      "w": 640,
      "h": 480,
      "startdelay": 0,
      "battr": [13,14],
      "minbitrate": 300,
      "maxbitrate": 1500,
      "api": [1],
      "companionad": [ {
        "w": 300,
        "h": 250,
        "id": 1
      } ],
      "companiontype": [1, 2, 3],
    }
  ]
}

```

```

        "ext": {
            "skippable" : 1
        }
    },
    "site": {
        "id": "234563",
        "domain": "siteabcd.com",
        "page" : "http://siteabcd.com/page.htm",
        "publisher": {
            "id": "google_25"
        }
    },
    "device": {
        "ip": "64.124.253.1",
        "geo": {
            "county": "US",
            "region": "NY",
            "city": "White Plains",
            "zip": "10601"
        },
        "ua": "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2.16)
        Gecko/20110319 Firefox/3.6.16",
        "language": "en"
    },
    "user" : {
        "id": "45asdf987656789adfad4678rew656789",
        "buyeruid": "1234567890"
    },
    "cur": ["USD"],
    "ext": {
        "ssp": "google"
    }
}

```

Bid Response

Top Level Bid Response Object

Field	Type	Description
id	string	ID of the bid request.

seatbid	array of objects	Array of Seat Bid objects. The length of the array can be either 1 (for yes-bid) and 0 (for no-bid).
cur*	string	Bid currency using ISO-4217 alphabetic codes. If not provided USD is assumed.

* Fields marked with asterisk are optional and may not be present in each response.

Seat Bid Object

Field	Type	Description
bid	array of objects	Array of Bid Objects. The only reason the length of this object can be greater than 1 is that there are several slots in the bid request.
seat*	string	ID of the bidder seat on whose behalf this bid is made. The field is required if the wseat field is present in the bid request.

* Fields marked with asterisk are optional and may not be present in each response.

Formally there can be multiple bids within Seat Bid for two reasons. The first one is that there can be several slots in request. The second one is when there are more than one bid for a single slot. The BidSwitch allows no more than two bids for a single ad slot.

Bid Object

Field	Type	Description
id	string	ID for the bid object chosen by the bidder for tracking and debugging purposes.
impid	string	ID of the impression object from bid request to which this bid applies.
price	float	Price as a float value, CPM. All prices assumed to be in USD, if cur parameter is omitted.

nurl*	string	Win notice URL. For banner impressions the ad markup should be returned via this URL. Ad markup should be in HTML format. For video impressions the url should contain the url of impression pixel. The url may contain substitution macros, see Macros section below. For application banner inventory this field will be IGNORED, use ext.js_url instead.
adid	string	ID that references the ad to be served if the bid wins.
adomain	array of strings	Advertiser's primary or top-level domain for advertiser checking. This can be a list of domains if there is a rotating creative. Note that some SSPs allow only one domain. To such SSPs we only send the first domain from the list.
cid*	string	Campaign ID or similar that is used by the bidder to track and organize their campaigns. The field is REQUIRED for some SSPs, including Nexage and MoPub.
ext*	object	Ext Object. This field is optional.

* Fields marked with asterisk are optional and may not be present in each response.

Ext Object

Field	Type	Description
deal*	string	This is the ID of the deal between a publisher and a seat. It is used only if an exchange supports private auctions. If this field is present it should be equal to one of the elements in pmp.deals field in the bid request object.
advertiser_name*	string	The name of the advertiser serving the creative. Required in bids responses to AdScale, Centro and YieldOne bids.
agency_name*	string	The name of the agency representing the advertiser. Required in bids responses to AdScale, Centro and YieldOne bids.
language*	string	Alpha-2/ISO 639-1 code of creative language. Required in bid responses to RMX bids.

cat*	string	IAB category of the creative. Required in bid responses to RMX bids.
liveintent*	Object	Required field for bid responses to LiveIntent bid requests.
google*	Object	Contains additional information for Google bids. The field is recommended.
vast_url*	string	The url pointing to the location the VAST document for the bid response. Required for bid responses to video traffic. Note that the VAST document should NOT contain impression tracking urls with win price macros.
js_url*	string	Javascript-based win notice URL. For in-app inventory the ad markup should be returned via this URL. For website or video inventory this field may optionally be used to substitute nurl field. Ad markup should be in JavaScript format. The url may contain substitution macros, see Macros section below.

* Fields marked with asterisk are optional and may not be present in each response.

NB: advertiser_name and agency_name are required by certain SSPs; responses to bid requests from these SSPs with these fields missing will be discarded.

SSP Specific Fields

LiveIntent Object

Field	Type	Description
img_url	string	URL of the creative image. In order to receive user cookie and win price this url should point to the DSP handler and be later redirected to the actual creative location. The url may contain macros, e.g. \${AUCTION_PRICE}. If this field is present, the nurl field of the response will be ignored.
click_url	string	Creative click URL.

Google Object

Field	Type	Description
attribute*	array of integer	List of creative attributes as defined by Google.
vendor_type*	array of integer	List of creative vendor types as defined by Google.

* Fields marked with asterisk are optional and may not be present in each response.

Nobid Response

If the DSP opts to pass on the bid request, the seatbid field in the bid response should contain an empty array of bids. The BidSwitch also accepts HTTP 204 responses as valid nobid responses.

Examples

Simple Bid Response

Following is an example of a bid response with the ad served on win notice. The bid for this impression is a \$9.43 CPM.

```
{
  "id": "1234567890",
  "seatbid": [{
    "bid": [{
      "id": "1",
      "impid": "102",
      "price": 9.43,
      "adid": "314",
      "cid": "42",
      "nurl": "http://adserver.com/winnotice?impid=102&winprice=
${AUCTION_PRICE}&click_url=${CLICK_URL:URLENCODE}",
      "adomain": ["advertiserdomain.com"]
      "ext": {
        "liveintent": {
          "img_url": "http://adserver.com/imp?impid=102",
          "click_url": "http://adserver.com/click?impid=102"
        },
        "advertiser_name": "Coca-Cola",
        "language": "en",
        "cat": "IAB12",
        "agency_name": "CC-advertising"
      }
    }
  ]
}
```

```
    }  
  }  
}
```

No Bid Example

Following is an example of sending no-bid response to the bid.

```
{  
  "id": "1234567891",  
  "seatbid": []  
}
```

Response to Private Bid Request

```
{  
  "id": "1234567892",  
  "cur": "USD",  
  "seatbid": [{  
    "bid": [{  
      "id": "1114125-afaff2f-af2251",  
      "impid": "1",  
      "price": 9.43,  
      "adid": "314",  
      "nurl": "http://adserver.com/winnotice?impid=102&winprice=  
${AUCTION_PRICE}&click_url=${CLICK_URL:URLENCODE}",  
      "adomain": ["advertiserdomain.com"],  
      "ext": {  
        "deal": "deal-1"  
      }  
    }  
  ]  
}]  
}
```

Macros

Macros may be used in nurl and ext.js_url fields.

Win price macro

In order for the exchange to convey certain information to the winning bidder (e.g., the

settlement price), some substitution macros can be inserted into the win notice URL definition. Prior to calling a win notice URL, the exchange will search the specified URL for any of the defined macros and replace them with the appropriate data. Note that the substitution is simple in the sense that wherever a legal macro is found, it will be replaced without regard for syntax correctness. Furthermore, if the source value is an optional parameter that was not specified, the macro will simply be removed (i.e., replaced with a zero-length string).

Macros	Description
<code>\${AUCTION_PRICE}</code>	Settlement price for the auction. The substituted value will be defined as CPM.

SSP click tracking url macro

Some SSPs require the DSP to add an SSP click url macro in all creatives for click reporting. DSPs wishing to trade with such SSPs should add the click tracking macro to the win notice URL definition. The BidSwitch will replace the macro with the SSP click tracking url. If the SSP doesn't support click tracking macros then the tracking macro will be replaced by an empty string.

Macros	Description
<code>\${CLICK_URL}</code>	A placeholder for the SSP click tracking url. Required for bids to SSPs that support click tracking.
<code>\${CLICK_URL:URLENCODE}</code>	SSP click tracking url in urlencoded form.

The SSP click url (if present) should be inserted before the landing page in the creative. The landing page url should be single-escaped.

Creative Preview

To accomplish the integration with Bid Switch, DSP should be able to provide an offline URL or other system to translate from DSP's creative_id to a web page for preview purposes. The example of this URL may look like:

http://example_dsp.com/preview_page?creative_id=1234567&other_info=foo

where one creative corresponds to one creative_id. This URL will only be accessed by SSPs and should not have any login authentication.

Additional Features

Bid Switch comes with a bunch of additional features that can be enabled as per client request. The first group of features is related to bid targeting and can be used to filter out undesired traffic and to keep bidding only on those requests the client is mostly interested in.

QPS Management

Bid Switch allows clients to limit the number of bid requests to a certain level. QPS limits can be applied on per datacenter basis. Please contact the Bid Switch support team for more information on this topic.

Bid Targeting

A comprehensive user targeting system has been one of the core parts of IPONWEB platform for many years. Now it is included into Bid Switch service and allows precise user targeting based on a variety of criteria including geolocation parameters, language preferences, device information, publisher and domain data, date and time targeting and so on. The sections below provide a brief description of targeting system capabilities that can be applied globally for a client.

Targeting Type	Targeting Options
Geo Targeting	country, region, city
Publisher Side Targeting	publisher, domain, page, size
Operation System	type, version
Browser	type, version
Device	type, maker, model

User Sync Targeting

Just like targeting to a specific audiences while bidding there is an option to selectively sync users between exchanges. User sync targeting allows a client to focus on a particular country filtering out all other users for matching. This feature can also be combined with QPS management option thus making it possible to match users from a certain country limited to a desired QPS rate.