VRTCAL

RTB RAPID INTEGRATION GUIDE

Revision 2.0

The VRTCAL Real-time Bidding (RTB) architecture allows a Demand Side Platform (DSP) to easily integrate its exchange. Our bidding endpoints support DSP connectivity through the OpenRTB 2.6 standard with backwards compatibility with all previous OpenRTB 2.x versions. We support pre-negotiated deals via the OpenRTB PMP standard. Our platform supports:

- In-app display and video
- Mobile web display and video
- Desktop web display and video
- Native display
- CTV video

To maximize the pool of available bids, VRTCAL supports user syncing for mobile-web supply.

This document provides the information necessary for OpenRTB DSPs to integrate with VRTCAL – SSP.

Table of Contents

A. OpenRTB Auction Flow

The following describes the typical flow of a bid request and response facilitated through the VRTCAL endpoints:

1. VRTCAL sends request for bids to the DSP.
2. The DSP returns bid response to VRTCAL.
3. VRTCAL closes auction.
4. VRTCAL selects the DSP as winning bidder.
5. VRTCAL substitutes all OpenRTB standard macro values in the winning bid notification url (nurl) before making a request to notify the winner.
6. VRTCAL serves winning ad creative with all OpenRTB standard macro substitutions made.

B. Endpoint (VRTCAL): Bid Requests

Our endpoints send bid requests to DSPs via the HTTP POST method. All required OpenRTB 2.6 standard JSON objects are sent with every request. In addition, any optional OpenRTB 2.6 data fields are sent with requests when available from upstream supply.

Our OpenRTB platform also supports all industry standard regional privacy and supply chain objects and extensions.

All parameters conform to the data type specifications of IAB OpenRTB 2.6. The specs can be found here:

https://iabtechlab.com/wp-content/uploads/2022/04/OpenRTB-2-6_FINAL.pdf

OpenRTB Protocol Version

VRTCAL supports the OpenRTB 2.6 standard with backwards compatibility with all previous OpenRTB 2.x versions . VRTCAL will indicate the protocol used for the request via the **x-openrtb-version** request header.

C. Endpoint (DSP): Bid Responses

Our endpoints support bid responses that conform to the OpenRTB 2.6 standard with backwards compatibility with all previous OpenRTB 2.x versions. All bid responses should be composed of JSON formatted data that at a minimum, contains:

1. A Response ID matching the ID we provide (Request ID)
2. At least 1 SEAT
3. At least 1 SEAT BID
4. Bids must have IDs and also contain a proper IMPRESSION ID
5. Bids must contain a CPM (price) (in USD)
6. Bids must contain the ad content within the ADM parameter unless it is returned via the win notice (NURL)

D. Ad Content and Win Notices

We support WIN NOTICE (NURL) and ADM based ad content delivery within bid responses. For win notices, we support the following MACROS to be returned with win notices:

${AUCTION_ID} - ID of the bid request; from BidRequest.id attribute. ${AUCTION_BID_ID} ID of the bid; from BidResponse.id attribute.

${AUCTION_IMP_ID} - ID of the impression just won; from imp.id attribute. ${AUCTION_SEAT_ID} ID of the bidder seat for whom the bid was made.

${AUCTION_AD_ID} - ID of the ad markup the bidder wishes to serve; from bid.adid attribute.

${AUCTION_PRICE} - Settlement price using the same currency and units as the bid (always USD).

${AUCTION_CURRENCY} - The currency used in the bid (explicit or implied); for confirmation only.

NOTE: Any macro data encoding needs can be negotiated during integration/integration testing.

E. Integration Setup and Testing

For on-boarding potential bidders, we will set up a specific test environment for integration confirmation and testing. This will allow for fast integration and verification of proper communication for both RTB parties. Our onboarding team will also inquire if you have a metrics API available as we have support for automated DSP metrics API data pulls for fast discrepancy identification internally.