

## BUZZ API OVERVIEW

- Getting Started with the Buzz API >
- Upcoming Changes to the Buzz API
- GET-ting Data from the API >
- POST-ing to Creating Objects
- PUT-ting Objects with the API
- DELETE-ing Objects with the API

## BUZZ CONCEPTS

- Accounts, Users, Roles, Permissions >
- Advertisers, Campaigns, Line\_Items,...
- Creatives, Creative Assets, Templa... >
- Conversion Events >
- Reporting >
- Segments
- Line Items, Targeting Templates a... >

## STINGER

### About the Stinger Bidder

- Beeswax Architecture / Life of a Bid
- Bidding Agent >
- Data Augmentation
- Auction Filtering Available for Stinger
- Cookie Syncing

## ANTENNA

- Introduction to Antenna
- Data Feeds
- Antenna SQL Access >

# About the Stinger Bidder

 SUGGEST EDITS

## Overview and Goals

Stinger is the bidding engine of the Beeswax stack. Stinger was developed with the intention of allowing customization of key components including data augmentation of requests, targeting modules, bidding algorithms and the models that feed into algorithms.

 TABLE OF CONTENTS

- Overview and Goals
- Reading

## Reading

Stinger documentation covers the following

- Bidding algorithms customization: Bidding Agent
- First party data augmentation: Data Augmentation
- Log and streaming data: Data Feeds
- Beeswax Architecture / Life of a Bid

 Updated 3 months ago

## What's Next

- > Bidding Agent
- > Data Augmentation
- > Beeswax Architecture / Life of a Bid

## BUZZ API OVERVIEW

- Getting Started with the Buzz API >
- Upcoming Changes to the Buzz API
- GET-ting Data from the API >
- POST-ing to Creating Objects
- PUT-ting Objects with the API
- DELETE-ing Objects with the API

## BUZZ CONCEPTS

- Accounts, Users, Roles, Permissions >
- Advertisers, Campaigns, Line\_Items,...
- Creatives, Creative Assets, Templa... >
- Conversion Events >
- Reporting >
- Segments
- Line Items, Targeting Templates a... >

## STINGER

- About the Stinger Bidder
- Beeswax Architecture / Life of a Bid

## Bidding Agent

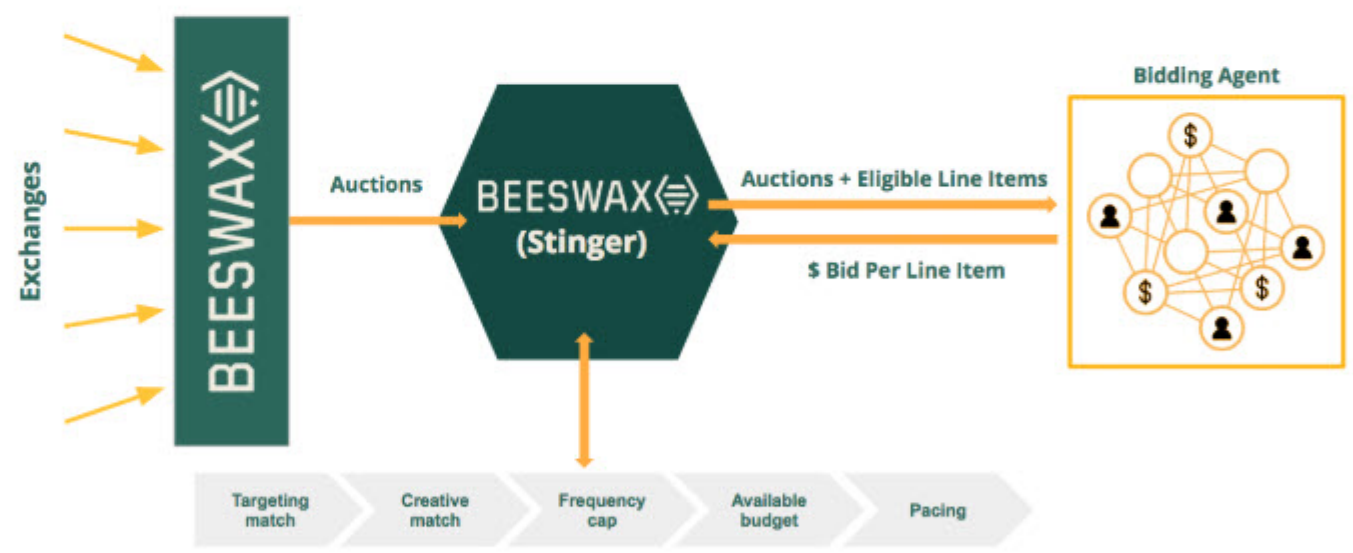
- Bidding Agent and Augmentor Re...
- Dynamic Macros (Real-Time Custo...
- Production Best Practices
- Data Augmentation
- Auction Filtering Available for Stinger
- Cookie Syncing

# Bidding Agent

SUGGEST EDITS

Having your own "bidding agent" in Beeswax allows you to implement custom real-time decisioning in your private bidder. Common use cases for real-time decisioning include custom bid price algorithms, custom budget pacing or custom creative decisioning including dynamic creatives. Bidding agents can be written in any language, they simply must be able to parse our data interfaces and respond within the strict timeout of 10ms.

The diagram below shows how a bidding agent for real time price decisioning works:



## Bidding Agents in Practice

Reading from left to right in the diagram, the Stinger bidder receives a fully normalized open RTB request. Once it enters the bidder, Beeswax applies several layers of filtering to determine which line items configured in Buzz match the given request and are thus eligible to serve. The layers of filtering include matching targeting and creative attributes, respecting any user-specific frequency caps, keeping within the daily and total budgets, and pacing evenly. What remains after filtering is a list of line items that are eligible to bid for this request – we call these `ad candidates` in our protobuf definitions.

## BUZZ API OVERVIEW

- Getting Started with the Buzz API >
- Upcoming Changes to the Buzz API
- GET-ting Data from the API >
- POST-ing to Creating Objects
- PUT-ting Objects with the API
- DELETE-ing Objects with the API

## BUZZ CONCEPTS

- Accounts, Users, Roles, Permissions >
- Advertisers, Campaigns, Line\_Items,...
- Creatives, Creative Assets, Templa... >
- Conversion Events >
- Reporting >
- Segments
- Line Items, Targeting Templates a... >

## STINGER

- About the Stinger Bidder
- Beeswax Architecture / Life of a Bid
- Bidding Agent** ▾
- Bidding Agent and Augmentor Re...
- Dynamic Macros (Real-Time Custo...
- Production Best Practices
- Data Augmentation
- Auction Filtering Available for Stinger
- Cookie Syncing

## ANTENNA

- Introduction to Antenna
- Data Feeds
- Antenna SQL Access >

Next, the Private Bidder makes an HTTP POST containing the `request.proto` to a bidding agent outside of the Beeswax cloud, typically hosted by a customer in the same AWS Region. The request proto contains both the full openRTB request along with a list of the eligible line items. The Bidding Agent must receive this request, parse it, and either respond with an HTTP 204 to not bid, or an HTTP 200 with a list of line items and corresponding bid prices. We enforce a strict timeout of 10ms for your HTTP response, so it is important that your Bidding Agent is in the same AWS region and highly performant.

## Data Interfaces and Tools

The following are all the data interfaces used when implementing a bidding agent:

File	Description
<code>request.proto</code>	The interface between the bidder and the bidding agent. Includes both the openRTB request and the adcandidates
<code>openrtb.proto</code>	This proto is imported by the request.proto. It contains the full openRTB request
<code>adcandidates.proto</code>	This proto is imported by the request.proto. It contains all the candidate line items for the given request

The following tool can be used to stream sample bidding agent requests to an endpoint

Tool	Description
Bid requestor tool	This tools generates Beeswax bid http requests and sends to a designated endpoint with http headers used in the bidding service and http body generated from the specified input file.

## Pacing-as-a-Service

A common technique for bidding algorithms is to "pace" buying during the day or over the lifetime of the campaign in order to deliver ads on a regular schedule. Beeswax supports what we call "Pacing-as-a-Service" for Bidding Agents where customers can utilize their own algorithms with our built-in pacing. This service works by filtering bids from your Bidding Agent based on the ratio of current delivery against goals.

There is no technical implementation required to utilize Beeswax pacing, as long as the strategy is set up to do so when first created by the Beeswax team.

## Response Codes

See also: [Bidding Agent and Augmentor Response Codes](#).

Google Chrome

83.0.4103.61

Google Chrome

83.0.4103.61

11:46:21 PM 7/6/2020

Windows 10 Pro for Workstations 64-bit Build 17134

11:46:21 PM 7/6/2020

Windows 10 Pro for Workstations 64-bit Build 17134



## Buzz — REST APIs for campaign management and reporting

[GET STARTED WITH THE BUZZ API](#)

Many ad tech vendors still don't offer comprehensive APIs. At Beeswax we built our API first, and everything we do is built on top. Our Buzz REST API allows your team to customize workflow, dashboards or anything else in our system.



## Stinger — An extensible RTB Bidder

[LEARN ABOUT EXTENDING STINGER](#)

Stinger, our proprietary bidder, allows you to write your own algorithms, add your own targeting, and manage huge numbers of campaigns and line items. And you never need to worry about the basics: frequency caps, budgeting, targeting, or any of the other basics of running a bidder.

### Buzz API Overview

- Getting Started with the Buzz API
- Upcoming Changes to the Buzz API
- GET-ting Data from the API
- [View All 6](#)

### Buzz Concepts

- Accounts, Users, Roles, Permissions
- Advertisers, Campaigns, Line\_Items, Creatives
- Creatives, Creative Assets, Templates, Rules
- [View All 7](#)

### Stinger

- About the Stinger Bidder
- Beeswax Architecture / Life of a Bid
- Bidding Agent
- [View All 6](#)

## Antenna

- Introduction to Antenna
- Data Feeds
- Antenna SQL Access

## Buzz API Reference

- Advertisers
- Account Alerts
- Accounts
- View All 43

## Changelog

VIEW ALL

- New Log Fields
- Added bid\_shading field to bidding json
- Antenna Docs Added
- Upcoming Changes to the API
- New Macros

## Guides

VIEW ALL

- Buzz API Overview
- Buzz Concepts
- Stinger
- Antenna