BidList Creation to Enable Differential Bidding on the Trade Desk

Prasanna venkataraman Hebbar[1], Prof. Praveena T[2]

[1]Student, Dept. of CSE, R V College of Engineering, Bangalore
[2]Assistant Professor, Dept. of CSE, R V College of Engineering, Bangalore
---***---

Abstract - *Differential Bidding helps the traders to carru out the Digital advertising campaigns by considering the multiple scenarios like age, gender, occupation etc. Of the ad-viewer. BidList Creation tool helps the traders to carry out Differential Bidding on The Trade Desk platform utilizing a feature called 'bid factor', provided by The Trade Desk, through which bid value can be changed based on different scenarios. With the help of this tool Traders will be able to save a lot of time and bed at the right price for the required ad-space.*

Key Words: **BidLists, Differential Bidding, The Trade Desk, Bid factor**

1. INTRODUCTION

Marketing Intelligence is the information needed to drive business outcomes[1], elevating insights to fuel brand strategy rather than just using them to optimize tactics. Creating Marketing Intelligence requires applying data science and analytics across multiple data sources to understand trends and the customers that create them.. Digital marketing and online advertising helps traders to target their audience and thus to make their products widely popular. It can be done through interacting with the audience via digital applications. It is easier for the traders to analyse the criterion in order to reach to higher level by analysing their performances.

Like usual advertising, even in Digital advertising bidding happens for an ad-space. This bidding happens based on different scenarios like age, gender, place, and occupation etc of ad-viewer. For all these scenarios bid value has to be set by the trader. If we make combinations of all the scenarios we get millions of them and writing bid cases for all of them is a huge headache. That is where the Differential Bidding comes into picture. If combination of scenarios happen simultaneously then bid value will change by a factor called bid factor. By opting for Differential bidding traders can save a lot of time, since it is automated to place bid values considering combination bid scenarios.

The Trade Desk is a platform where the bidding takes place. Millions of traders will bid for ad-spaces. So ad-space will be allotted to the highest bidder. If bid value is low ad-space will not be allotted. If bid value is high trader will be at loss of winning at higher value. So it is important to bid at the appropriate value. That is where BidLists are created from traders side which facilitates the bidding. BidLists contain all the necessary information required for bdding. They consists of a parameter called 'BidLines' where different scenarios are written along with the bid factor. With the help of this

trader will be able to facilitate the digital advertising process smoothly on the Trade Desk.

2. OBJECTIVE

- Capabilities to perform CRUD operations on bidlists in The Trade Desk. A bidlists will be used by the traders to perform differential bidding on the marketing campaign run on The Trade Desk platform

- Once created, a trader should be able to associate (as well as de-associate) the bidlists from the marketing campaings on The Trade Desk.

- All the failures with respect to data errors, downtime with service, rate limiting issues on the external systems needs to be covered with proper error validations, to communicate back to the trader.

3. METHODOLOGY

- Get the requests from the user via User Interface

- Pass the requests to data Validator to check the authenticity of the user.

- Pass the requests to rate Limiter to check if request rate is under specified range by The Trade desk

- Pass valid requests to Processors, which process it to The Trade desk

4. TOOLS USED

4.1 SPRING BOOT

Spring Boot is a Java Framework that helps us to build scalable Web Applications for RESTFUL Services. With the features like Inversion Control and Dependency Injection, it is easier to write codes for a Web based Application. With feature rich, SpringBoot makes the implementation easier. In this project SpringBoot is used to write Application interfaces.

4.2 THE TRADE DESK

The Trade desk is a platform where a trader can run marketing campaigns. The Trade Desk will handle all the backend aspects of the whole Bidding process. It also provides the different data points for pre-bidding as well as post-bidding. In this project we get API's from The Trade Desk and on the top of these API's we create wrapper API's.

The Trade Desk console is used to test the whole working of the BidList creation Tool.

4.3 POSTMAN

Postman is a tool used to send and get responses for API's. In our project we use Postman to test our API's

4.4 UBUNTU

Ubuntu is an Operating System used for running project. In this project we use Ubuntu Operating System for building and running the project.

5. IMPLEMENTATION

Implementation of BidList Creation can be done in the following two steps

1. BidList Creation

2. Associator/ Disassociator

5.1 BIDLIST CREATION

BidList Creation can be done with the following steps.

- User Interface
- Application Interface
- Data validator
- Rate Limiter
- Processor

5.1.1 USER INTERFACE

This is where the user interacts with the system. User has to feed the system with the business logic. The user will fill the necessary scenarios and its applicable bid factors. This can be done either via graphics module or can take input in the textual forms depending on the scenario. Adding more graphics module will make the life of the trader easier as they will not be much into technacalities. GUI is implemented in React JS for having smoother user interactions.

5.1.2 APPLICATION INTERFACE

User requests will be converted to queries in this process. Data frows from User Interface to Application Interface. Application Interface converts user input to an executable query. This is implemented using Spring Boot.

5.1.3 DATA VALIDATOR

Data validator validates the flow of data. The Trade Desk has certain restrictions and different types of owners. Different owners have different levels of access. Data Validator makes

sure that it is achieved. If request is valid it will be passed to the next level else error will be thrown to the user.

5.1.4 RATE LIMITER

The Trade Desk has certain limitations on the no of requests that can be sent. This may be dependent on the kind of subscription purchased. So Rate Limiter makes sure that request rate is under a given limit. It receives request as the input and outputs whether it is valid or not. If valid, request will flow to processor, else Error will be thrown to the user. This is implemented using the Rate Limiter Library provided by Java.

5.1.5 PROCESSOR

Processor is responsible for processing the user request and coordinating with the Trade desk. It should have a quing system to maintain queue for request flow to the Trade desk. This is also responsible for throwing unexpected errors from The Trade desk.

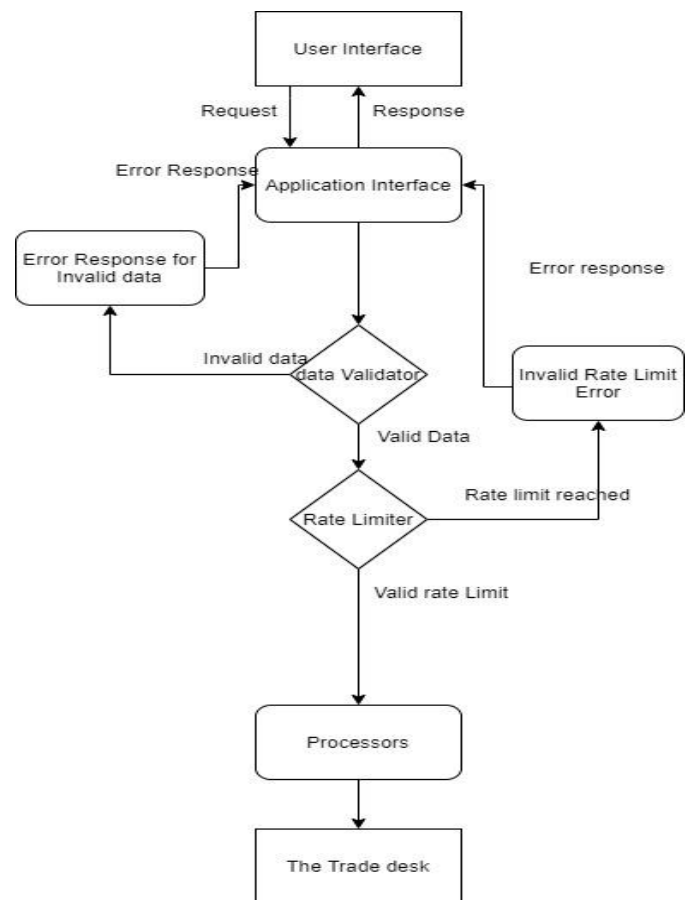The complete flow of BidList Creation is depicted in the fig 1.

Fig -1: BidList Creation

5.2 ASSOCIATOR/DISASSOCIATOR

Once the BidList is created on the Trade desk,it should be associated to marketing Campaings while running them. Associator/Disassociator takes care of this action. So with the help of this same bidlists can be associated to different marketing campaigns. Complete Process has been depicted in the fig 2.
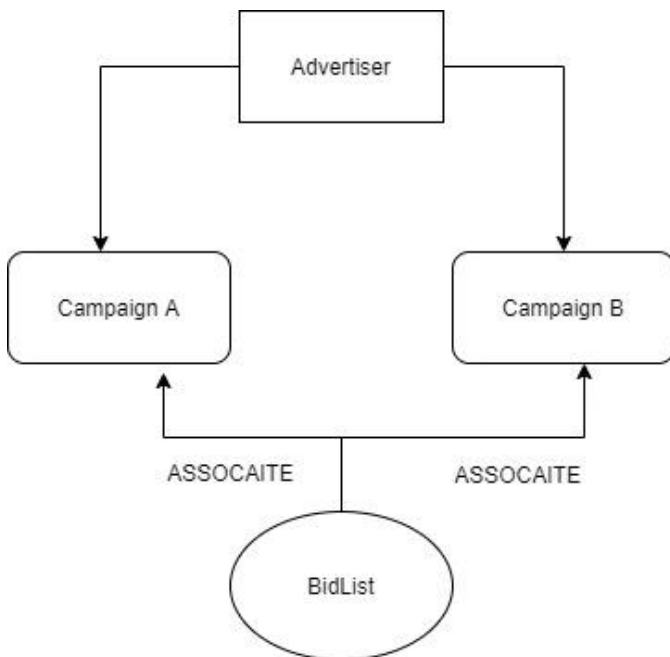

Fig -2: Associator/Disassociator

5. CONCLUSION

This project aims at enabling Differential Bidding on thte Trade Desk by creating BidLists and associate them to different marketing campaigns. Throught the whol processos it has been taken care to handle all the issues and errors like downtime issues, Rate Limiting Issues etc. To make the process user friendly.

ACKNOWLEDGEMENT

REFERENCES

[1] Pandu Prakoso Tardan, Satrio Baskoro Yudhoatmojo, Muhammad Rifki Shihab, Digital Marketing Strategies for mobile commerce Collaberative consumption startups,Information Technology Systems and Innovation(ICITSI),2017

[2] https://www.thetradedesk.com/products/enterprise-apis [The Trade Desk API Portal]

[3] https://guava.dev/releases/19.0/api/docs/index.html?com/google/common/util/concurrent/RateLimiter.htmlhttp://martinfowler.com/eaaDev/EventSourcing.html [Rate Limiter Concepts]

[4] B. Golden. (2016). 3 Reasons Why You Should Always Run Microservices Apps in Containers. Accessed: Nov. 11, 2017.

BIOGRAPHIES

PRASANNA VENKATARAMAN HEBBAR
Final year Engineering student, Computer Science,R V College of Engineering

Prof. Praveena T,
Assistant Professor, Department of Computer Science, R V College of Engineering