eBay Search Service

Preliminary Draft

Isabel Giang and Maxwell Wenger

CSS490 Group Y4

January 26, 2021

Contents

Overview

Problem Statement

In the last few months, eBay users have reported increasingly slow response times from eBay's website when searching for auctions. Application log analysis for the eBay Master Service has confirmed that our FindAuctions API for the eBay Master service takes significantly longer to return a response when we search for auctions using keywords. It also takes longer to search for a large number of auctions. This is negatively impacting

eBay's user experience for existing users and is hindering the website's chances of being adopted by new users.

Performing further analysis on the eBay Master service schema shows that the FindAuctions API must scan the entire table to first find active auctions. Then it must scan those active auctions to find auctions that have titles with the keywords we are looking for.

Temporary Solution

As a bandaid solution to temporarily address this problem, we have added an AuctionStatus index to the Auction table of the eBay Master Service database. This allows us to immediately access all active auctions instead of being forced to scan the entire Auction table to find active auctions before querying with keywords.

This speeds up the FindAuctions API enough to fix the poor user experience temporarily, but this will not be enough as the company grows. The number of records in the Auction table, and subsequently, the number of active auctions will increase at an exponential rate.

Next Steps

To solve this more permanently, we want to create a separate search service that will handle searching for auctions by keyword and/or category. We will call this new service the eBay Search Service.

This search service will have its own database that can only be changed by external users via its API. Whenever a new FindAuction API request is made, this service will perform the required business logic and databases accesses instead of the eBay Master service.
The eBay Search Service will support searching in the following ways:

- Searching for all auctions
- Searching for auctions based on auction status (closed, open, pending, cancelled)
- Searching for auctions based on keywords
- Searching for auctions based on category

- Searching for auctions based on keywords and category

Our search operation will only allow “and” operations between search terms. “Or” operations must be done using multiple calls.

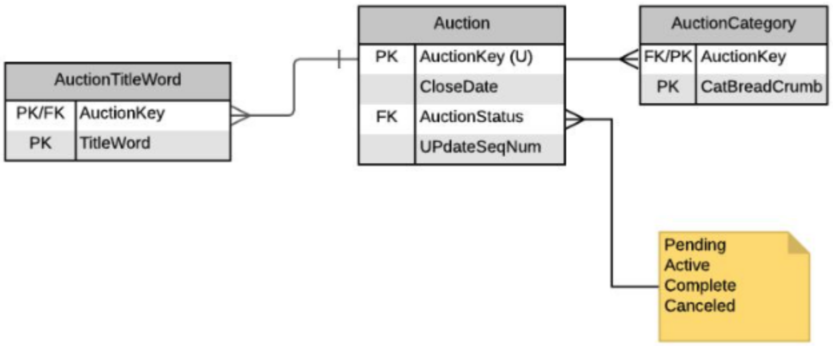
eBay Search Service API Specification

API List

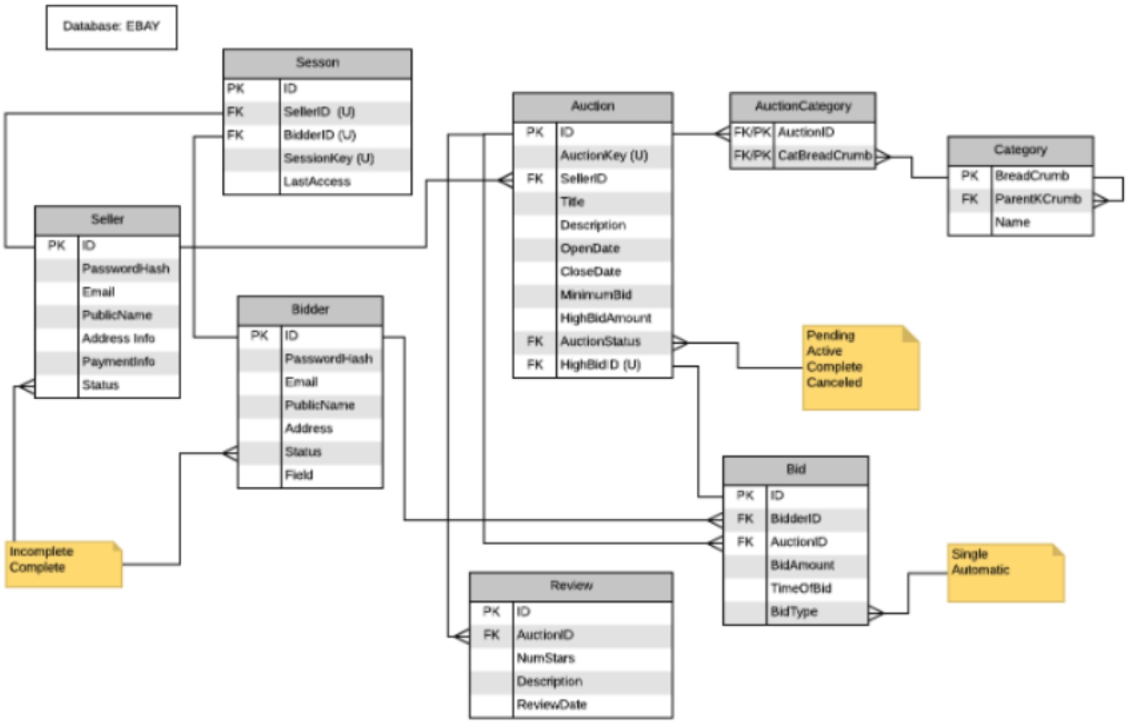
API Name	Description
saveSearchableAuction	Updates or creates a searchable auction
removeSearchableAuction	Removes an auction.
findAuctions	Finds auctions based on the given keywords and/or category.

Schemas

SearchDB



MasterDB


Database: EBAY
Sesson
PK ID
FK SellerID (U)
FK BidderID (U)
SessionKey (U)
LastAccess
Seller
PK ID
PasswordHash
Email
PublicName
Address Info
PaymentInfo
Status
Bidder
PK ID
PasswordHash
Email
PublicName
Address
Status
Field
Incomplete
Complete
Auction
PK ID
AuctionKey (U)
FK SellerID
Title
Description
OpenDate
CloseDate
MinimumBid
HighBidAmount
FK AuctionStatus
FK HighBidID (U)
AuctionCategory
FK/PK AuctionID
FK/PK CatBreadCrumb
Category
PK BreadCrumb
FK ParentKCrumb
Name
Pending
Active
Complete
Canceled
Bid
PK ID
FK BidderID
FK AuctionID
BidAmount
TimeOfBid
BidType
Single
Automatic
Review
PK ID
FK AuctionID
NumStars
Description
ReviewDate

API Descriptions

saveSearchableAuction

Updates an existing auction or creates one if no auction with the auction key exists.

Input

- **auctionKey** - Public key that uniquely identifies an auction from the eBay Master Service.
- **title** - Title of the auction
- **closeDate** - Closing date of the auction (ISO-8601 timestamp)
- **category** - Category breadcrumb
- **status** - Status of the auction (closed, open, pending, cancelled)
- **seqNum** - Sequence number of the auctions table. Each time there is an update, it is incremented by one. Requests made with smaller sequence numbers than the one in the database are ignored.

```
{
    "auctionKey": <string>,
    "title": <string>,
    "closeDate": <string>,
    "categories": [ <string>, ...],
    "status": <string>,
    "seqNum": <int>
}
```

Output

Scenario	Response

Successfully saved an auction	`{`
	`    "success": true,`
	`    "seqNum": <int> // current seq num in`
	`                    // the database.`
	`}`

removeSearchableAuction

Removes an auction from the search database that corresponds with the given auction key.

Input

- **auctionKey** - Key that uniquely identifies the auction

```
{
    "auctionKey": <string>
}
```

Output

Scenario	Response

Successfully removed an auction	`{`
	`    "success": true`
	`}`

findAuctions

findAuctions will return a number of results based on the user's query. The parameters search by "anding" the results together (e.g. if you ask for a keyword "red", "LG" with the category "ELE:PHO" signifying phones in electronics, red LG phones will be returned, but not red Nokia phones, or red LG refrigerators). To perform searches that "or" search terms, multiple searches must be made. lastAuctionKey and numResults are to be used for pagination. findAuctions will return numResults number of entries ordered from oldest to newest, starting from the lastAuctionKey. If no lastAuctionKey is provided, findAuctions will return numResults number of results starting from the first result.

Input

- **keywords** - Keywords that auction results will be matched to
- **category** - Category breadcrumb
- **status** - Status of the auction (closed, open, pending, cancelled)
- **lastAuctionKey** - The last auction key will be last auction not included in the results that are numResults long. If given, results will be returned starting from this page.
- **numResults** - Number of auctions returned per page

```
{
    "keywords": [ <string>, ...], // optional
    "status": <string>, // optional
    "categories": [<string>, ...], // optional
    "lastAuctionKey": <number>, // optional
    "numResults": <number>
}
```

Output

scenario	**response**

Successfully search with found results.	`{`
	`    "success": true,`
	`    "results": [`
	`        auctionKey: <string>,`
	`        ...`
	`    ],`
	`    "pageIndex": <int>`
	`}`

Successfully search with no results.	`{`
	`    "success": true,`
	`    "results": [ ]`
	`}`

eBay Search Service Internals

saveSearchableAuction

Update searchable auction will update fields, keywords, and categories of an existing auction in the search table.
New keywords will be generated if a new title is provided. The new keywords will be added to the database, and keywords associated with they auction key but were not again generated from the updated title will be removed from the database. The fields will be based on the auction key, therefore the auction key may not be updated. Any optional fields not included in the message to the service will not be changed. To remove a title, an empty string must be provided (“”). If an empty string is provided, all keywords will be deleted as before, but no new ones will be generated.

removeSearchableAuction

Remove searchable auction does simply that. It will delete the keywords and category entries from each table matching the auction key that was requested to be deleted. In general, its not good practice to delete things out of the database, but in the case, its okay because this the search database is treated more like a cache than a datastore. The original data will still exist in the main database.

findAuctions

Find auctions builds a query based on the provided information. It generates this query as an “and” operation, so results passed back are only those that fit all the criteria you passed to findAuction. It handles pagination by having the user keep track of the last auction it saw, and request a certain number after that. So from our ordered query, we will pull numResults of results back, starting from one after the last auction key. If the last auction key was deleted/does not exist at the time of the call, it will return numResults number of results starting from the first value again. This helps get around the issue of missed auctions that were added during the search session.

Changes to eBay Master Service

Master DB Schema Changes

The sequence number field has been added to the auction table to count and uniquely identify the order and number of changes made to the auction table.

createAuction

Creates an auction as it did before, although now the createAuction service is responsible for calling createSearchableAuction to create a search entry for the auction. Sequence number is set to zero.
Create auction will then receive a response from the search service. If the response is anything other than success with the returned sequence number larger than or equal to the one the service sent, it will fail the update and alert the user.
The impact on the database from createAuction is documented in the master api documentation. The impact on the database from createSearchableAuction is documented in section saveSearchableAuction.

updateAuction

Update auction will work as it did before. Except, now, update auction will call saveSearchableAuction with the updated parameters it will receive. The sequence number is incremented by one.
Update auction will then receive a response from the search service. If the response is anything other than success with the returned sequence number larger than or equal to the one the service sent, it will fail the update and alert the user.
The impact on the database from updateAuction is documented in the master api documentation. The impact on the database from saveSearchableAuction is documented in section saveSearchableAuction.

findOpenAuctions

This service in the master DB now simply calls the findAuctions service from the search service, passing through the keywords and category it was passed, but only looking for auctions with a status of open.

Logging

This section describes a suggested format for this service's application logs.

```
{
    "start": <string>,
    "duration": <string>,
    "api": <string>,
    "params":
        {
            "original": {
                "auctionkey": <string>,
                "title": <string>,
                "closeDate": <string>,
                "status": <string>,
                "category": [<string>, ...],
                "seqNum": <int>
            },
            "updated": {
                "auctionkey": <string>,
                "title": <string>,
                "closeDate": <string>,
                "status": <string>,
                "category": [<string>, ...],
                "seqNum": <int>
            }
        }
}
```

- **start** The time the service first receives the call.
- **duration** The amount of time it takes from when the service receives the call to when the service responds.
- **api** The name of the API that is called.
- **params** The original and updated data that the API changed. Optional data that was not changed is not included in either the updated or original logs.