

# eBay Search Service

Isabel Giang and Maxwell Wenger

CSS490 Group Y4

January 26, 2021

## Contents

<b>Overview</b>	<b>1</b>
Problem Statement . . . . .	1
Solution . . . . .	2
Current Solution . . . . .	2
Next Steps . . . . .	2
<b>eBay Search Service API Specification</b>	<b>2</b>
createSearchableAuction . . . . .	2
updateSearchableAuction . . . . .	3
findAuctions . . . . .	4
<b>eBay Search Service Intervals</b>	<b>4</b>
<b>Changes to eBay Master Service</b>	<b>4</b>
createAuction . . . . .	4

## 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.

To solve this problem, we have created an additional search database that contains only the necessary auction data required to for search functionality. We have then indexed the database by both

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.

## Solution

### Current Solution

To partially address this problem, we have added an AuctionStatus index to the Auction table so we can immediately access all active auctions instead of search for all auctions every time we query 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, we want to create a separate search service that will handle searching for auctions by keyword and/or category.

## eBay Search Service API Specification

### createSearchableAuction

This creates an entry in the search database for the auction, all of the auction's keywords, and the auction category.

#### Input

```
1 {  
2   "auctionKey": <string>,  
3   "title": <string>,  
4   "closeDate": <string>,  
5   "status": <string>,  
6   "breadcrumb": <string>  
7 }
```

## Output

Scenario	Response
Successfully created search entry.	<pre>{     "success": true }</pre>
TODO: Make error states	Yeah! What he said!

## updateSearchableAuction

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.

### input

```
1 {
2   "auctionkey": <string>,
3   "title": <string>, // optional
4   "closedate": <string>, // optional
5   "status": <string>, // optional
6   "breadcrumb": <string> // optional
7 }
```

### output

scenario	response
successfully updated search entry.	<pre>{     "success": true }</pre>
provided auction key does not exist in the search database.	<pre>{     "success": false,     "exception": "auctionkeynotfound" }</pre>
todo: make error states	yeah! what he said!

## 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

```

1 {
2   "keywords": [ <string>, ... ], // optional
3   "status": <string>, // optional
4   "breadcrumb": <string>, // optional
5   "lastAuctionKey": <string>, // optional
6   "numresults": <string>
7 }
```

## output

scenario	response
Successfully search with found results.	<pre>{   "success": true,   "results": [ auctionKey &lt;string&gt;, ... ] }</pre>
Successfully search with no results.	<pre>{   "success": true,   "results": [ ] }</pre>
todo: make error states	yeah! what he said!

# eBay Search Service Intervals

## Changes to eBay Master Service

### createAuction

Creates an auction as it did before, although now the createAuction service is now responsible for updating the search database.

It does this by first splitting up the title into keywords. The decision on how to identify keywords will be up to the implementation. Then, the createAuction service will create a new entry in the Auction table in the search database for this auction using the same information used to create the auction table in the main service database. Then, it will create an AuctionTitleWord entry with the string text of the word and the auction key for each previously identified keyword. Then, it will create an AuctionCategory entry with the respective auction category.

If three calls to the search service fails, the call will be queued up to be attempted again with other failed calls at a regular interval.