

McMaster University

Faculty of Engineering

Department of Computing and Software

COMP SCI 4ZP6

Market Identifier

By

Ryan Ticlo (ticlor)

Kenny Kim (kimm54)

Rickesh Mistry (mistrrp)

Muyideen Jimoh (jimohma)

Supervisor

Dr. Archer

Supervisor

Dr. Anand

April 2019

**Project Abstract**

Market Identifier is a business analytics application that helps small businesses and new entrepreneurs identify profitable niche opportunities. Therefore, allowing new entrepreneurs to strategically penetrate markets with low budgets. By helping small businesses identify these niche markets, we can provide a new stream of income to entrepreneurs, stimulate the economy, and provide a better selection of goods to consumers. Market identifier’s algorithms analyze the results of the queries by the user and output an opportunity score, based on various marketing and economic indicators.

***Acknowledgement***

**Dr. Anand**

**Dr. Archer**

**Table of Contents**

1. **Introduction**
2. **Problem Formulation**

2.1 Problem Statement

2.2 Problem Solution

1. **Specification**
   1. Product Functions
2. **Requirement**

4.1 Server-Side Requirements

1. **System Diagram**
2. **Value proposition**
3. **Cost Analysis**
4. **Conclusion**
5. **References**
6. **Introduction**

One of the biggest factors determining a new business’s success is timing, which is closely related to luck. This is one of the biggest reasons people avoid starting their own businesses: the risk of failure, and uncertainty of not knowing if their idea is needed and viable. Big data can help solve this problem. When start-ups are provided with in-depth market analysis, they can better identify profitable gaps to capitalize on. In order for entrepreneurs to generate income, in the presence of large corporations, they must fill markets in which big corporations are not selling in. These markets are not big enough and therefore not as desirable for larger corporations. However, there may be sufficient demand within these niche markets for individual entrepreneurs to make a sizable income off of. By helping small businesses identify these niche markets, we can provide a new stream of income to entrepreneurs and a better selection of goods to consumers.

1. **Problem Formulation**
   1. **Problem Statement**

Uncertainty and risk associated with starting a new business venture.

* 1. **Problem Solution**

Providing business analytics to small businesses and entrepreneurs which will reduce risk and uncertainty. The application will analyze sales, rating, brand dominance, and review count to help gauge the success probability of a market.

1. **Specifications**
   1. **Product Functions.**

|  |  |
| --- | --- |
| **Package** | **Description** |
| APIScript | Contains all functions required to query data from external servers. Queries are made using API’s from Keepa (Amazon), eBay, and Walmart.  Main functions include:   * queryEbayData * queryAmazonData * queryWalmartData |
| CriteriaFilter | Contains functions which can be used to reduce the number of rows in the data set by applying upper and lower bound filters.  This functionality has been removed from the final solution because of conflicts with the MI Score functionality which is considered more valuable to the users.  Main functions include:   * FilterData * quicksort * binarySearchLow * binarySearchHigh |
| ModifyData | Contains all functions required to transform raw data from each e-commerce platform into a standard form for processing.  Main functions include:   * reduceEbay * reduceAmazon * reduceWalmart |
| MIScore | Contains all functions used to calculate the MI Score of each data set based on brand dominance, rating, and reviews.  Main functions include:   * calcScore * addBrand |

1. **Requirements**
   1. **Server-Side Requirements**

API’s are provided by Amazon, eBay, and Walmart to retrieve data for products. Queries will be initiated from Market identifier’s AWS server to servers from the companies mentioned above, via their APIs. Quicksort and binary search algorithms are implemented for sorting and searching through the results from the queries. The result from a sort procedure will display top results using a rank field returned byAmazon. The price, sales, and feedback are not required for the algorithm but will be displayed tobenefit the user.

1. **System Diagram:**

The code is hosted on an AWS plan that consists of a ubuntu server. A user interacts with the web application using their local PC, our server makes the relevant queries to retrieve data from Amazon, eBay and Walmart servers, sorts and analyzes this data, then displays it to the user in a meaningful way.

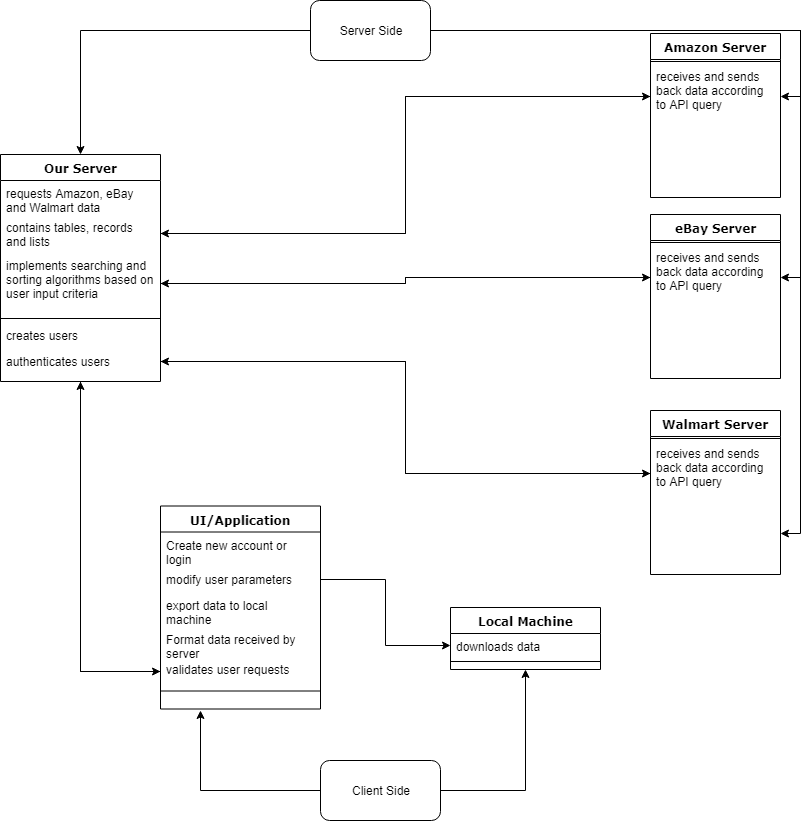


Figure (a) System Flowchart

Server location: ec2-3-91-79-241.compute-1.amazonaws.com

1. **Value Proposition**

Market Identifier provides the user with data from 3 different e-commerce platforms. Whereas, similar solutions such as Helium10, focus just on 1 platform (Amazon). Our solution also provides the user with an MI score: a score out of 4. A product that has an MI score of 1 is a good opportunity to capitalize on. While an MI score of 4 is bad. The score is determined by analyzing various key indicators, such as review count, rating, brand dominance, and price.

1. **Cost Analysis**

API access - $99/month

AWS server cost - $12.99/month

Domain name cost - $10/year

1. **Significance of data**

* Review count: If there are too many reviews, it is indicative of a well-established, saturated market, with little room for differentiation.
* Brand dominance: If there is one brand with the majority of sales, it suggests that there is brand loyalty, and consumers will be reluctant to purchase elsewhere. Thus, making it a harder niche to penetrate.
* Price: For startups in the e-commerce space, it is difficult to profit off an item that is below $10, given the increase in marketing costs.

1. **Conclusions**

The final product is capable of retrieving data from Amazon, Walmart, and eBay via their respective APIs. The algorithms are capable of analyzing the results and producing an opportunity score for a particular market based on various indicators such as sales, reviews, rating, and brand dominance.

1. **References**

*All API documentation*. [Online]. Available: https://developer.ebay.com/docs. [Accessed: 18-Apr-2019]

“Using eBay RESTful APIs,” *eBay Developers Program*. [Online]. Available: https://developer.ebay.com/api-docs/static/ebay-rest-landing.html. [Accessed: 18-Apr-2019]

“keepa Documentation¶,” *keepa Documentation - keepa 0.15.0 documentation*. [Online]. Available: https://keepaapi.readthedocs.io/en/latest/. [Accessed: 18-Apr-2019].

“Product Lookup API,” *Walmart Open API - Product Lookup API*. [Online]. Available: https://developer.walmartlabs.com/docs. [Accessed: 18-Apr-2019].

Amazon Web Services. ec2-3-90-176-69.compute-1.amazonaws.com

1. **Appendix**

# External Interface Requirements

## User Interfaces

*API’s are provided by Amazon, eBay, and Walmart to retrieve data for products. OpenWeatherMap provides an API* which will be used by our solution to retrieve *current* weather data.Queries will be processed using Amazon and Walmart API’s when users search for keywords orproducts. We will use a sorting algorithm to display top results using a rank field returned byAmazon. The price, sales, and feedback is not required for the algorithm but will be displayed tobenefit the user. Weather and average income filters can be selected by users. Historical datawill be searched for dates with the selected values and new queries to Amazon will be processto find sales data.

## API Documentation

### Weather Data

#### Overview

OpenWeatherMap.org provides a free to use current weather data API. The free version will allow us to make up to 60 calls per minute and provides a 5 day/3 hour forecast API at a 95% availability. The free features are expected to be sufficient for the startup usage rates. If more calls or availability is required in the future we can update our subscription to a paid service which can provide up to 200,000 calls per minute at a 95% availability.

Official API Documentation

<https://openweathermap.org/current>

Company API Key

43644e4e8baf019c3a323353f3b594ee

#### Queries

Our solution will be calling weather data by city name to get current weather for a broad geographical area. The accuracy of a call by geographic coordinates or by ZIP code is not necessary for our requirements. Our solution will be expecting a JSON response to parse the data. The API call format follows:

*api.openweathermap.org/data/2.5/weather?q={city name},{country code}&mode={modeType}&units={unitType}&appid={APIkey}*

An example call for current weather in Hamilton, ON follows:

*api.openweathermap.org/data/2.5/weather?q=Hamilton,CA&mode=json&units=metric&appid=43644e4e8baf019c3a323353f3b594ee*

#### Responses

Expected Positive Response

<current>

*<city id="5969785" name="Hamilton">*

*<coord lon="-79.87" lat="43.26"/>*

*<country>CA</country>*

*<sun rise="2019-01-30T12:36:41" set="2019-01-30T22:29:33"/>*

*</city>*

*<temperature value="-17.75" min="-20" max="-16" unit="metric"/>*

*<humidity value="64" unit="%"/>*

*<pressure value="1009" unit="hPa"/>*

*<wind>*

*<speed value="12.3" name="Strong breeze"/>*

*<gusts value="15.4"/>*

*<direction value="230" code="SW" name="Southwest"/>*

*</wind>*

*<clouds value="20" name="few clouds"/>*

*<visibility value="6437"/>*

*<precipitation mode="no"/>*

*<weather number="600" value="light snow" icon="13d"/>*

*<lastupdate value="2019-01-30T16:45:00"/>*

*</current>*

Negative Response Incorrect city name or country code

*<ClientError>*

*<cod>404</cod>*

*<message>city not found</message>*

*</ClientError>*

Negative response for incorrect API key:

*{"cod":401, "message": "Invalid API key. Please see http://openweathermap.org/faq#error401 for more info."}*

Negative response for other incorrect parameters:

* If the mode parameter is not valid then the response will default to a JSON format.
* If the units parameter is not valid then the response will default to show the temperature in Kelvin.

### Ebay Data

#### Overview

eBay has several API packages available for accessing item data. With the item\_summary API, we can query on an item name and retrieve a list of item identification codes with some additional details such as price. Item id can be used with the Item API for additional information. RESTful calls using eBay APIs require authorization to be set in the request header. The required ID changes according to the API. See the official API for more details on which ID to use.

App ID (Client ID)

RickeshM-MarketId-PRD-4393cab7d-e5a6ffdf

Dev ID

30f859b7-87ae-42ca-830b-8c1c31b8e6da

Cert ID (Client Secret)

PRD-393cab7d81f4-9329-4a25-9cda-1a4f

Base64-encoded Credentials

Umlja2VzaE0tTWFya2V0SWQtUFJELTQzOTNjYWI3ZC1lNWE2ZmZkZjpQUkQtMzkzY2FiN2Q4MWY0LTkzMjktNGEyNS05Y2RhLTFhNGY=

Official API Documentation

*All API Docs:*

<https://developer.ebay.com/docs>

*Useful getting started documentation for making calls:*

<https://developer.ebay.com/api-docs/static/ebay-rest-landing.html>

*Item\_Summary API documentation used for getting item listings:*

<https://developer.ebay.com/api-docs/buy/browse/resources/item_summary/methods/search>

*Item API documentation used for specific item data:*

<https://developer.ebay.com/api-docs/buy/browse/resources/item/methods/getItem>

*Client Credentials Grant Flow API required for Item\_Summary:*

<https://developer.ebay.com/api-docs/static/oauth-client-credentials-grant.html>

#### Queries

The following query was made in javascript. The headers required as specified in the official documentation have been set. A POST call is made to create an access token with the client credentials grant flow. The access token is then used as the authorization for the item\_summary call.



#### Response

The JSON response can be parsed to retrieve the required information for our solution.



#### Dataset

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| title | The title of the product. |
| epid | An EPID is the eBay product identifier of a product from the eBay product catalog. This indicates the product in which the item belongs.  Occurrence: Conditional |
| categories | This container returns the primary category ID of the item (as well as the secondary category if the item was listed in two categories).  Occurrence: Always |
| marketingPrice | This container is returned if the item is eligible for a seller discount and contains the item's original price, and the seller discount amount and percentage.  Occurrence: Conditional |
| shortDescription | This text string is derived from the item condition and the item aspects (such as size, color, capacity, model, brand, etc.). Sometimes the title doesn't give enough information but the description is too big. Surfacing the shortDescription can often provide buyers with the additional information that could help them make a buying decision. |
| estimatedAvailabilityQuantity | The estimated number of this item that are available for purchase. Because the quantity of an item can change several times within a second, it is impossible to return the exact quantity. So instead of returning quantity, the estimated availability of the item is returned.  Occurrence: Conditional |
| primaryProductReviewRating | The container that returns the product rating details, such as review count, rating histogram, and average rating.  Occurrence: Conditional |

### Walmart Data

#### Overview

Walmart Open API specifications provides several API tools including the "Trending API", "Reviews API", "Data Feed API" and "Product Lookup API". These tools require an active account with an API Key in order to be used freely by any developer. The API provides special feeds of items that are hot selling on the Walmart website. These items range from rollback and bestsellers category. Their current feeds are updated every 24 hours. The lookup API provided allows viewing of important parts of a product page to the end customers, along with the pricing status in real time. Trending API gives information on what is bestselling on Walmart.com right now. The items are curated on the basis of user browse activity and sales activity, and updated multiple times a day. A total of 5 calls can be made per second while a total of 5,000 calls can be made per day using the production API key assigned.

*Dev Application ID*

mktIdentif\_

*Dev API Key*

qufjxk28t4su7mr8sgxhpd7n

Official API Documentation

*All API Docs:*

<https://developer.walmartlabs.com/docs>

*I/O Docs: Interactive API Tool For Making Calls:*

[*https://developer.walmartlabs.com/io-docs*](https://developer.walmartlabs.com/io-docs)

*Product Lookup API documentation being used in our solution:*

<https://developer.walmartlabs.com/docs/read/Home>

#### Queries

The following query was made for an item with id “12417832” and format of response is in json! The only required query parameter needed to make a successful query is an apiKey which is the API access key of any developer.

Sample Lookup for item id 12417832

http://api.walmartlabs.com/v1/items/12417832?apiKey={qufjxk28t4su7mr8sgxhpd7n }&lsPublisherId={Your LinkShare Publisher Id}&format=json

#### Response



#### Dataset

Walmart API useful fields

Reviews API

Name (Name of product (string type)

Reviews (Actual reviews submitted by people with upvotes and downvotes (string and integer type))

availableonline (Online availability of product (Boolean type)

salePrice (Market price of product (integer type)

reviewStatistics (ratingValue (i.e 1,2,3,4,5) and number of counts such a rating has)

Search API

totalResults (number of search results related to item searched for)

numItems (This is the actual number of items returned that matches the search field)

Items API

numReviews (

Stock (Availablity of item on the website. Value is of type boolean (True or False))

salePrice (Market selling price of product on website (integer type))

customerRating (This gives the overall rating of item based on the ratings posted by customers (Rating scale 1 to 5))

### Amazon Data

#### Overview

Amazon Product Advertising API tool enables to request items on Amazon and returns the result. It can be sorted by popularity, price, condition, etc. The request can be written in any preferred languages, and since it uses RESTful calls, we can set the authorization in the header, and request for accessing the database by URLs. In our product, Amazon APIs will be used to access customer reviews by setting ResponseGroup header to Reviews, seller reviews, price of the products, popularity of the products, product promotions, and use those information to display the lists of information for users to see and decide whether he/she wants to participate in the competition.

Amazon access key ID: AKIAIJPGALEGINWEHUDQ

Amazon Secret Access Key: ZbIQQjue7kGJB1UNjHgmgnisOVJgaRKYOYar9yqM

Associate ID: sensoryparent-20

Official API Documentation

*Getting started: Making your first request:*

**-    Open source code in eclipse**

**-    Change parameters Operations to one of :  [‘ItemSearch’, ‘SimilarityLookup’, ‘ItemLookup’, ‘CartCreate’, ‘CardAdd’, ‘CardModify’, ‘CartClear’, ‘CartGet’, ‘BrowseNodeLookup’]**

**-    Change parameter search index to one of : ['All','Wine','Wireless','ArtsAndCrafts','Miscellaneous','Electronics','Jewelry','MobileApps','Photo','Shoes','KindleStore','Automotive','Vehicles','Pantry','MusicalInstruments','DigitalMusic','GiftCards','FashionBaby','FashionGirls','GourmetFood','HomeGarden','MusicTracks','UnboxVideo','FashionWomen','VideoGames','FashionMen','Kitchen','Video','Software','Beauty','Grocery',,'FashionBoys','Industrial','PetSupplies','OfficeProducts','Magazines','Watches','Luggage','OutdoorLiving','Toys','SportingGoods','PCHardware','Movies','Books','Collectibles','Handmade','VHS','MP3Downloads','HomeAndBusinessServices','Fashion','Tools','Baby','Apparel','Marketplace','DVD','Appliances','Music','LawnAndGarden','WirelessAccessories','Blended','HealthPersonalCare','Classical']**

**-       Change parameter Response group to : ['Tags', 'Help', 'ListMinimum', 'VariationSummary', 'VariationMatrix',**

**'TransactionDetails', 'VariationMinimum','VariationImages',**

**'PartBrandBinsSummary', 'CustomerFull', 'CartNewReleases',**

**'ItemIds', 'SalesRank', 'TagsSummary', 'Fitments',**

**'Subjects', 'Medium', 'ListmaniaLists',**

**'PartBrowseNodeBinsSummary', 'TopSellers', 'Request',**

**'HasPartCompatibility', 'PromotionDetails', 'ListFull',**

**'Small', 'Seller', 'OfferFull', 'Accessories',**

**'VehicleMakes', 'MerchantItemAttributes', 'TaggedItems',**

**'VehicleParts', 'BrowseNodeInfo', 'ItemAttributes',**

**'PromotionalTag', 'VehicleOptions', 'ListItems', 'Offers',**

**'TaggedGuides', 'NewReleases', 'VehiclePartFit',**

**'OfferSummary', 'VariationOffers', 'CartSimilarities',**

**'Reviews', 'ShippingCharges', 'ShippingOptions', 'EditorialReview',**

**'CustomerInfo', 'PromotionSummary', 'BrowseNodes',**

**'PartnerTransactionDetails', 'VehicleYears', 'SearchBins',**

**'VehicleTrims', 'Similarities', 'AlternateVersions',**

**'SearchInside', 'CustomerReviews', 'SellerListing',**

**'OfferListings', 'Cart', 'TaggedListmaniaLists',**

**'VehicleModels', 'ListInfo', 'Large', 'CustomerLists',**

**'Tracks', 'CartTopSellers', 'Images', 'Variations',**

**'RelatedItems','Collections']**

<https://docs.aws.amazon.com/AWSECommerceService/latest/GSG/SubmittingYourFirstRequest.html>

*Components of a request:*

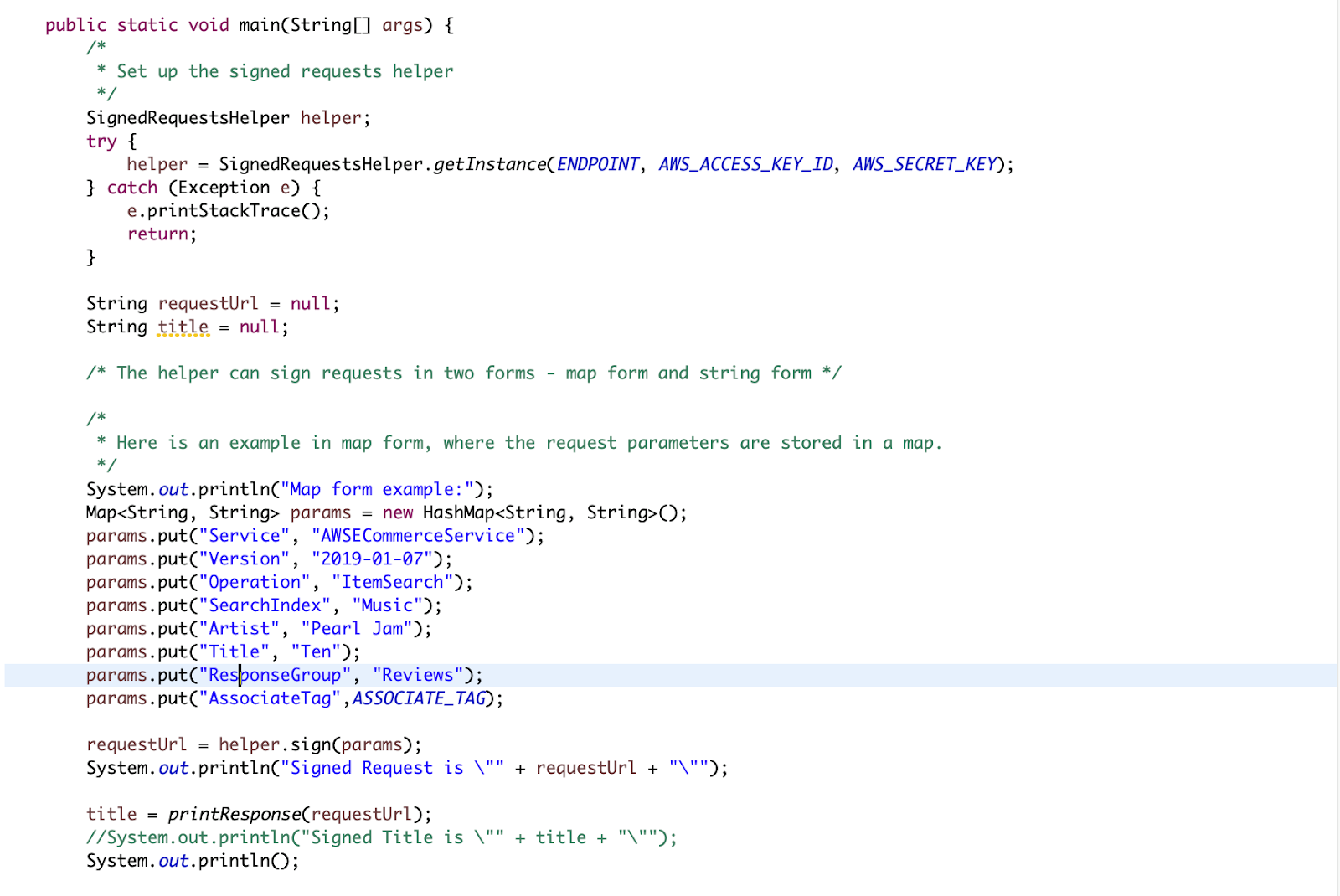
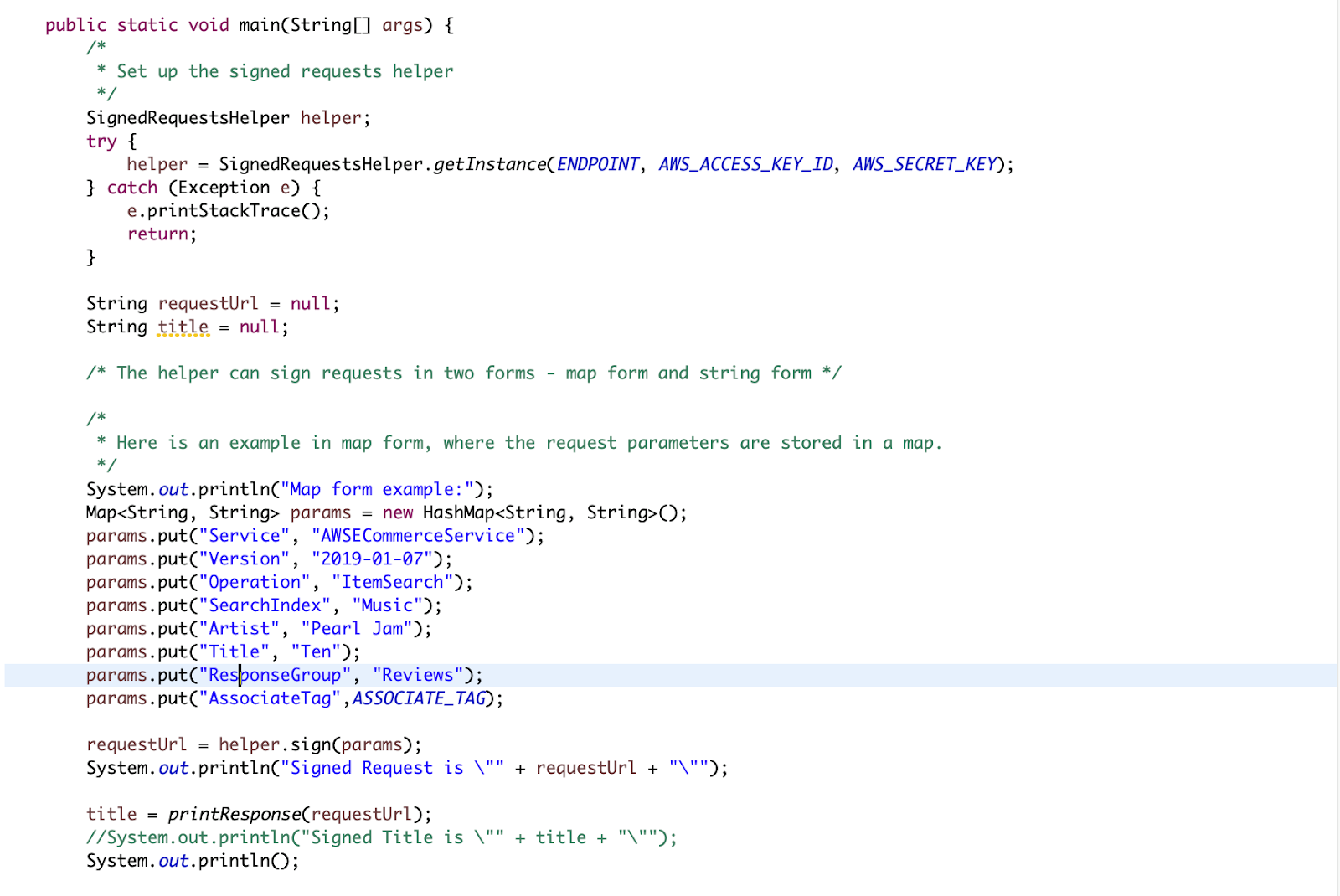
<https://docs.aws.amazon.com/AWSECommerceService/latest/GSG/PartsofaRequest.html>

*Implementing a request:*

<https://docs.aws.amazon.com/AWSECommerceService/latest/GSG/ImplementinganA2SRequest.html>

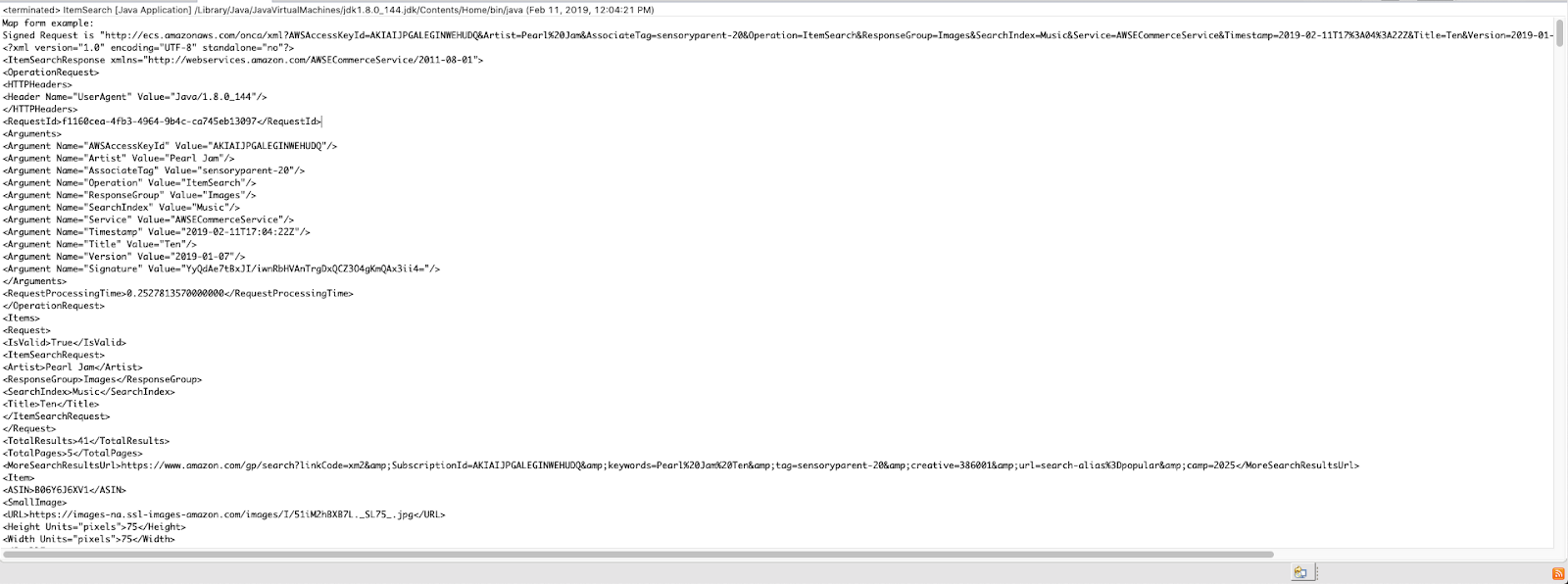
#### Queries

The query was sent from java and JavaScript and the responses were in xml. When sending a request using java, we can build a parameters with all the information we need to send over (Operation, Title, ResponseGroup, etc) and receives responds. When we are using a JavaScript, we also build an URL that contains all the headers of information we want to retrieve, and receives response.

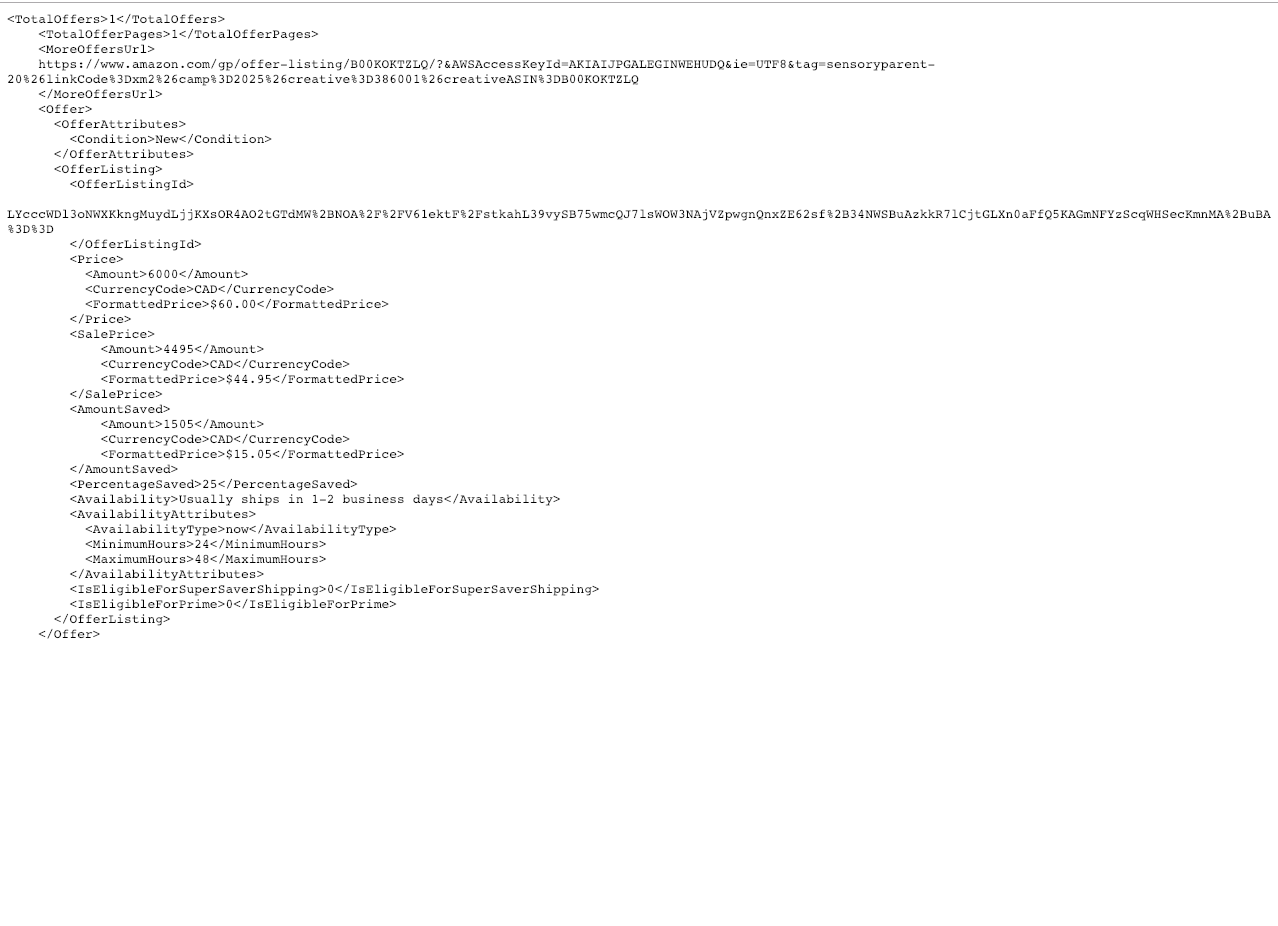


#### Response

**Response from JAVA applet**



**Sample XML response**



#### Dataset

ItemLookup API

* Reviews (Reviews that are written by people who actually purchased the product (xml/json type))
* SalesRank (Rank of item ranked along all items on Amazon(integer type))
* ReviewSort (Sort the reviews by ascending or descending order of rating/submission date/etc(xml/json type) )
* OfferSummary (description of the summary of the item  (xml/json type))

ItemSearch API

* SalesRank (Rank of item ranked along items in the search pool (integer type))
* MaximumPrice/MinimumPrice (field that will can be used to return items in the price range (xml/json type))
* Availability (description of availability of items in the search pool (xml/json type))
* SearchInside (from the search pool, returns search results with additional parameters (xml/json type)
* VariationSummary( provides lowest price, highest price, lowest sale price, and highest sale price for all child item in search pool (xml type))
* SearchBins(Narrowing search results by Brand Name, Price Range, Subject, Percentage Off)