

## 1. SPECIFICATION OF WEBSERVICES

### To list a new item for auction.

Operation: Add new item

Accessible by this URL: /abay

HTTP method used: POST

parameters: seller, title, stprice, description,dateTime

Status code and the result returns on a successful request: 201 (Created)

How the result is presented: This webservice returns a string with the message "Item saved."

From the point of view of the user's interface the New item appears in the table with all items saved highlighted in green colour. Take in account in this point that AJAX put in the cache some information and may be the table seems that doesn't refresh the information. In this case the problem is the browser's cache. Regarding the code in made for no caching any information with the instruction:

```
$.ajaxSetup({cache:false});
```

Status code returned on a failure: 400 (Some error)

### To retrieve all items for auction.

Operation: Retrieve all items for auction

Accessible by this URL: /abay

HTTP method used: GET

parameters: any

Status code and the result returns on a successful request: 200 (OK)

How the result is presented: This webservice returns a string following json format as follows:

```
[{"id":123,"title":"some_title","description":"some_description","seller":"some_seller","currentPrice":"9.99","deadline":"1435363620000"}, {"id":124,"title":"some_title1","description":"some_description1","seller":"some_seller1","currentPrice1":"19.99","deadline":"1435363620001"}]
```

Status code returned on a failure: 400 (Some error)

### To retrieve a single item given its ID.

Operation: Retrieve a single item given its ID

Accessible by this URL: /abay/123456789

That's /abay/itemID

HTTP method used: GET

parameters: any

Status code and the result returns on a successful request: 200 (OK)

How the result is presented: This webservice returns a string following json format as follows:

```
{"id":123,"title":"some_title","description":"some_description","seller":"some_seller","currentPrice":"9.99","deadline":"1435363620000"}
```

Status code returned on a failure: 400 (Some error)

### **To place a bid on an item.**

Operation: Retrieve a single item given its ID

Accessible by this URL: /abay

HTTP method used: POST

parameters: idItem, bidder, amount

Status code and the result returns on a successful request: 201 (Created)

How the result is presented: This webservice returns a string with the message "Bid saved."

Status code returned on a failure: 400 (Some error)

### **To retrieve all bids on an item (given an item ID).**

Operation: Retrieve all bids on an item(given an item ID)

Accessible by this URL: /abay/123456789

That's /abay/itemID

HTTP method used: GET

parameters: attribute="bids"

Status code and the result returns on a successful request: 200 (OK)

How the result is presented: This webservice returns:

In case that this itemID have bids: a string following json format as follows:

```
[{"bidder":"some_bidder","amount":"99.99","date":"1435363620000"},
```

```
{"bidder": "some_bidder1", "amount": "49.99", "date": "1435363620001"}}
```

In case that this itemID haven't bids: a string following text/plain format with the message "The item with id=xxx have not bids"

Status code returned on a failure: 400 (Some error)

## 2. STATEMENT OF COMPLIANCE

Functional Requirements. aBay has the following major functions:

- List an item for auction.
- Refresh and display all items on auction, including their current prices.
- Clicking an item in the list shows the item details.
- While reading the details of an item, you can place a bid.

### Listing an Item

Each item has a unique ID which is generated by the system. The seller does not enter the item ID. Instead, the user needs to enter the following details:

Seller: The seller's name/ID.

Title: A short description of the item to show in the item list.

Description: A more detailed description of the item.

Starting price: The starting price of the auction. When an item is freshly listed, its current price is set to its starting price.

Deadline: The date and time of the bidding deadline.

### Showing the Items

The home page shows a list of items on auction, including all open and closed ones. For each item, you only need to show the deadline, item title, and current price.

Take in account that information last a short time to appear in the web page.

### Showing an Item's Details

Clicking an item in the list shows the details of the selected item.

### Placing a Bid

On the item details page (see section 3.1.2 above), a user can place a bid by entering a bidder name/ID and a maximum offer. Placing a bid triggers a series of operations:

If the new bidding price is not higher than the current price of the item, the bid is rejected as it will surely lose.

A bid placed after the auction deadline is rejected. A bid is given a time-stamp when it reaches the server. See the API document of the Bid class for details.

If there is no bid on the item yet, the new bid wins for the moment. The current price of the item remains unchanged (i.e. still the starting price), as the bidder needs to pay no more than the starting price to **win the item. Bid accepted.**

### **Bibliography:**

---

Google (2012) *Vallalarblogs*, [online] Available at: <https://code.google.com/p/vallalarblogs/downloads/list> [Accessed 26th Apr 2015].

Google (2015) *Google Cloud Platform*, [online] Available at: <https://cloud.google.com/appengine/docs/> [Accessed 26th Apr 2015].

IETF (1999) *RFC 2616. Hypertext Transfer Protocol – HTTP/1.1*. [online] Available at: <https://tools.ietf.org/html/rfc2616> [Accessed 26th Apr 2015].

jQuery Foundation (2015) *jQuery API Documentation*, [online] Available at: <http://api.jquery.com> [Accessed 26th Apr 2015].

jQuery Foundation (2015) *jQuery UI*, [online] Available at: <http://jqueryui.com/> [Accessed 26th Apr 2015].

Stack Overflow (2015) *Stack Overflow*, [online] Available at: <http://stackoverflow.com> [Accessed 26th Apr 2015].

Study Trails (2014) *Java Google Json (Gson) Introduction*, [online] Available at: <http://www.studytrails.com/java/json/java-google-json-introduction.jsp> [Accessed 26th Apr 2015].

Oracle and/or its affiliates (2013) *API JAVA 7.Response Status*. [online] Available at:

<http://docs.oracle.com/javase/7/api/javax/ws/rs/core/Response.Status.html> [Accessed 26th Apr 2015].

W3Schools (2015). *W3Schools*. [online] Available at: <http://www.w3schools.com> [Accessed 26th Apr 2015].