

# Automated Testing of eBay's Search Functionality

---

## Test Report

09/19/2024

## **Table of Contents**

### **1.0 Introduction**

1.1 Purpose

1.2 Specification

### **2.0 Test Summary**

2.1 Search Functionality

2.2 Search Suggestions

2.3 Search Pagination

### **3.0 Test Results**

2.1 Search Functionality

2.2 Search Suggestions

2.3 Search Pagination

### **4.0 Discussion**

## **1.0 Introduction**

### **1.1 Purpose**

The purpose of this test report is to outline the tests that were performed on eBay's search functionality and UI elements. This report will also discuss the results of the tests outlined in the test summary portion of the document.

### **1.2 Specification**

Create an automation framework in Java using Cucumber and Selenium WebDriver to test a web application's search functionality. The framework should include feature files, step definitions, and page objects. Write test scenarios to cover different search scenarios such as valid/invalid search terms, search suggestions, and search results pagination. Provide a report of the test results in a readable format.

## **2.0 Test Summary**

### **2.1 Search Functionality**

These tests were meant to test eBay's base search functionality and UI using a variety of valid/invalid search terms.

### **2.2 Search Suggestions**

These tests were meant to test eBay's search suggestion relevancy as well as the UI elements that make up the search suggestions.

### **2.3 Search Pagination**

These tests were meant to test eBay's search pagination. These included tests interacting with UI elements as well whether the UI elements accomplish their purpose.

## 3.0 Test Results

### 3.1 Search Functionality

Test 1: This test scenario is meant to test the search feature of eBay and see if the resulting page is relevant to the search. By sending a search term to the search bar, we can check the title of the resulting page to determine whether the page is relevant.

Result 1.1: Searching for “Nerf Toys” : PASS

Result 1.2: Searching for “Baseball” : PASS

Result 1.3: Searching for “Gherkins” : PASS

Test 2: This test scenario is meant to test searching for something that does not exist. We can check whether the search is null by looking for a “null search” header that appears when a search result has zero matches.

Result 2.1: Searching for “POISDZhjfgvvPA;EROIKFN” : PASS

Test 3: This test scenario is meant to test a search that will result in an error. This can be done by misusing eBay's search terms, in this case the ‘-’ character. Using this character in front of a word without any other words in the search will cause the search to result in an error and bring the user to a specific error page.

Result 3.1: Searching for “-baseball” : PASS

### 3.2 Search Suggestions

Test 1: This test is meant to determine whether search suggestions are visible when nothing has been typed in the search bar. This can be done by looking for whether the unsorted suggestion list HTML element is visible or not. In this case we are confirming that it is not visible.

Result 1: Suggestions not visible with nothing in search bar : PASS

Test 2: This test is meant to determine whether search suggestions are visible after something has been typed into the search bar. This test follows the same methodology as Test 1, however in this case we are looking for the opposite.

Result 2: Suggestions are visible with a term in the search bar : PASS

Test 3: This test is meant to determine whether search suggestions are relevant to the terms in the search bar. This can be done by getting the suggestion text out of each list element and checking that the suggestion contains the search term.

Result 3.1: Suggestions for "Soccer" : PASS

Result 3.2: Suggestions for "Nerf" : PASS

### 3.3 Search Pagination

Test 1: This test determines whether the pagination UI elements exist on a common search. This can be done by locating the ordered list that makes up the search pagination and checking that it has different pages.

Result 1: **Pagination UI elements exist on common search : PASS**

Test 2: This test determines whether the forward arrow UI element moves the user to the next page. This can be determined by keeping track of the page the user started on before clicking the forward arrows and the page the user ends on.

Result 2: **Forward arrow brings user to next page : PASS**

Test 3: This test determines whether the backwards arrow UI element moves the user to the previous page. This test uses the same methodology as Test 2, however we are testing the backwards arrow and are checking that it brings the user one page backwards.

Result 3: **Backwards arrow brings the user to the previous page : PASS**

Test 4: This test confirms whether a search will place the user on the first page of the results pagination. This can be done by checking the user is on page 1 after a search.

Result 4: **Check whether the user is on page 1 after a search : PASS**

Test 5: This test confirms the ability to move to specific pages by clicking that page's number, in this case the furthest visible page or last page. This can be determined by keeping track of the largest visible page number in the pagination list and then attempting to move to that page. If the user ends on that page, then we have moved to it.

Result 5: **Check whether user can move to specific pages : PASS**

## 4.0 Discussion

The test results above demonstrate the functionality of eBay's search feature. We accomplished the specification by testing eBay's search function, search suggestion functionality, and search pagination functionality. A notable challenge that was encountered along the way was having to deal with the latency of eBay's auto suggestions. When the user types something into the search bar, the suggestions are not immediately present, and the underlying HTML needs to be updated. This presented a problem when testing as we needed to wait on eBay's suggestion list to update before proceeding with further testing. This was accomplished using the 'WebDriverWait' class as part of Selenium.