



## WCS Integration with eBay: Using the eBay SDK and Web Services

### **Jaweed Ali Qureshi**

Sr. Software Engineer  
jaweed@royalcyber.com  
Royal Cyber Inc.,

### **Malik M. Sarfaraz**

Software Engineer  
sarfaraz@royalcyber.com  
Royal Cyber Inc.,

### **Abstract**

IBM WebSphere Commerce is a J2EE-based product which provides a platform for developers to write an integrated extensible solution that adds value in customer-centric online business. WebSphere Commerce allows developers to customize its functionality according to current business demands so that the integration of specialized business functionality like auctions, third party shipping rate & tax calculation within WebSphere Commerce is possible by using Web Services & XML. eBay is the world's online marketplace where community of individuals and merchants has equal opportunity to buy and sell new or used goods at fair prices. eBay provides online interface for their customer as well as offer SDK and API (Application Programming Interface) for tool developers to create specialized applications that satisfy business need and market demand of any B2C Stores.

The eBay Auction Management module within WebSphere Commerce Accelerator provides streamlines multi-channel management, integrated listing, and checkout tools within a single interface for your online B2C store.

The core functions this module is offer:

- Manage your eBay Listings.
- Centralized Product Data for Online Store and eBay Listings.
- Winning Bidders Checkout through your B2C Store.
- Scheduled List and Relist Items on eBay.
- Integrated Order Management and Inventory Tracking
- Control listing checkout and shipping preferences.
- Streamline order management by directing buyers to checkout through your web store or use eBay to collect payment.



## Introduction

eBay is a world wide online auction service provider where you can trade together with large retailers without high overheads or start up costs and sale or purchase item/items of your choice at competitive price

eBay provides online interfaces to sell and buy an item as well as API to interact with directly to the eBay databases. you can create your own application to communicate with the eBay databases by using the API such as to develop a java based application, eBay SDK for Java is available and you can download it at <http://developer.ebay.com/java/sdk/> and documentation for using eBay SDK is available with in this SDK.

## Structure of eBay Java SDK

eBay provides SDK both for Windows and Java Technology. SDK for Windows encapsulate the XML API and the SDK for Java is based on SOAP API. In this article we will discuss only Java SDK structure.

eBay SDK for Java is organized into five libraries:

- API Library represents invocations of a particular eBay service. Its include AddItem, VerifyAddItem, RelistItem, ReviceItem, EndItem, GetItem, GetCategories, GetFeedback etc.
- Kernel Library is closest to the SOAP layer and supplies the basic underlying and supporting functionality for API call classes. Its include ApiCall, ApiContext and ApiCredential. ApiCall is the base class that defines the functionality common to all API call classes. ApiContext defines the API context under which an API call is made. ApiCredential provides the mean for authentication data for a caller requesting user.
- Attribute Library it includes sample implementations of the data provider interfaces. Each of these classes downloads the data directly from eBay and caches it in memory.
- Picture Service Library defines the functionality needed for an application to use item picture files hosted on the eBay Picture Service (EPS) server. These include the classes eBayPictureService and PictureInfo.
- Helper Library the rest of the services required for completing the eBay cycle are available on Helper Library like Shipping Service, Payment services etc.

## Functionality Supported by SDK

- Listing an Items
- Monitoring active items
- Post-sales processing (end-of-auction / end-of-transaction)
- Getting and posting feedback
- Checking for category updates, get user information.



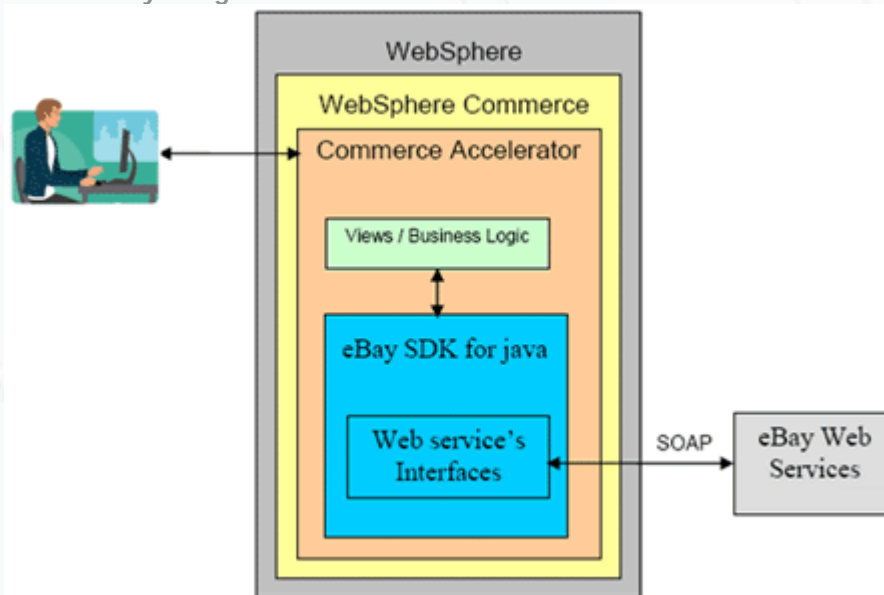
A model Seller Application should support the following activities:

- List an item
- Check Status
- Process Sales
- Relist Items
- Feedback processing

### eBay integration with WebSphere Commerce

We can also use the eBay API inside the WebSphere Commerce environment. Integration of eBay API in Commerce Accelerator allows communication directly with the eBay database. By using this API, you can manage all of your ebay listings form Websphere Commerce Accelerator and also list any product to ebat from your WCS store catalog . The following figure shows the integration of eBay API in WebSphere Commerce.

### WCS — eBay Integration Architecture



### Example using eBay API for Java in WebSphere Commerce Environment

Let's consider a scenario where a user wants to add an item from the wcs store to eBay using commerce accelerator as shown in Figure 1. A user request for add an item by clicking the "add Item" button available . Then he selects a product based on the selected category and than sets additional information related to item for auction as shown in Figure 2. Now the item is ready to send on eBay auction site that is accomplished by the code shown in Listing1. The code utilizes the benefits of web services with the help of eBaySDK to complete the task as add Item.

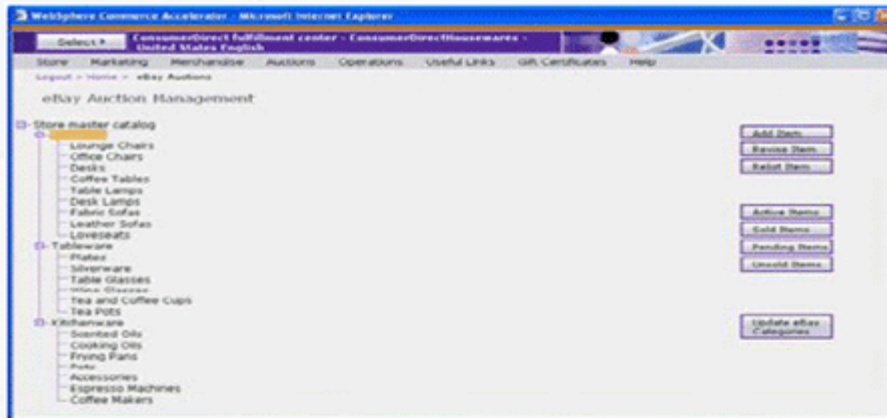


Figure 1

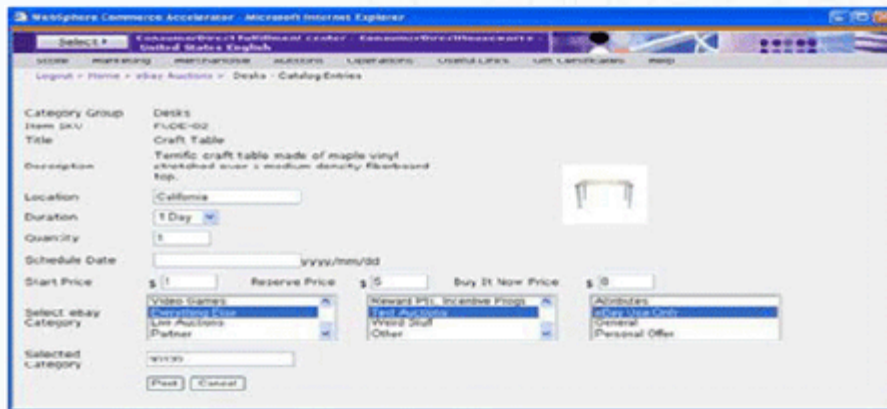


Figure 2

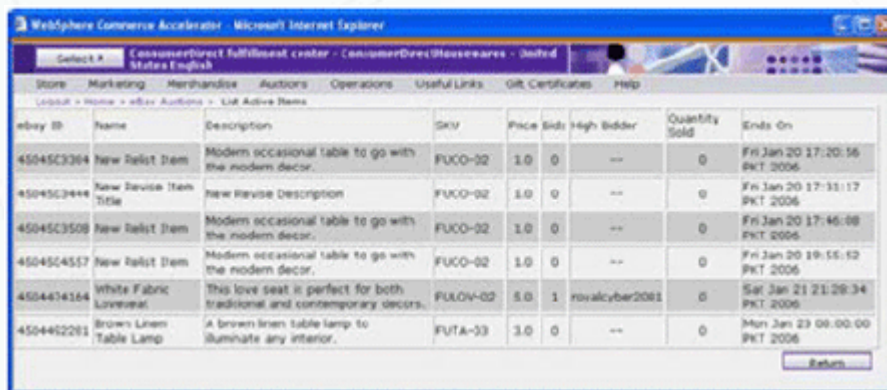


Figure 3

For calling any API call you must need to provide all the connection information that you can get from eBay site by creating your own account. You can either obtain a free individual account that allows you to use the eBay API, or a commercial account. By individual account you can make up to 5,000 calls per day in the Sandbox (test) environment and up to 50 calls per day in the live eBay environment for free. You can create your individual free account by visiting <http://developer.ebay.com/join> and click the join button on this page.



The code fragment at Listing 1, line #1 shows the creation of an ApiContext object, specify values for its properties, and assign the ApiContext object to the ApiContext property of the API call object. By specifying the values in ApiContext object we can:

- ✓ Direct the call to the appropriate eBay API server
- ✓ Pass requesting user authentication credentials to eBay
- ✓ Determine the language in which error messages are passed back to the application
- ✓ Add to the running count of API calls made using that ApiContext object
- ✓ Control the logging of API calls

An application can set up a single ApiContext object and reuse it for multiple API calls made using different API call objects (different API call classes or multiple instances of the same class). By reuse the ApiContext object we can track the number of call made and save the process of specifying authentication credentials again and again.

The code fragment at Listing 1, line #11 shows the creation of AddItemCall object, the API allows you to call or create different types of objects depends on the type of call you want to make. For example list of items that let you know about the status of your item on eBay auction such as sold item, unsold item, active item and pending item can be retrieve by GetItemCall.

Figure 3 show a list of item retrieved by GetItemCall.

AddItemCall object allows you to add an item to eBay auction site and the addItem method in Listing1, line#12 that takes your item as an argument not only makes a call to add an item but also provides you information about fee charged by eBay for auction and the id that is assigned by eBay to your item .



### Listing 1

```
// Create an ApiCallContext object for the call ApiContext
ApiContext apiContext = new ApiContext();

// Specify the authentication token for the requesting user ApiCredential
ApiCredential cred = apiContext.getApiCredential();
ApiAccount ac = cred.getApiAccount();
String soapServer = "https://api.sandbox.ebay.com/wsapi";

    ac.setDeveloper(devId);
    ac.setApplication(appId);
    ac.setCertificate(certId);
    cred.seteBayToken(token);

// Specify the eBay SOAP server URL
apiContext.setApiServerUrl(soapServer);
apiContext.setRouting("default");
// Create the AddItemCall object
11. AddItemCall api = new AddItemCall(apiContext);
// Make the call and get the fee and ebay id for the item submitted
12. FeesType fees = api.addItem(item);
```

This article showed us how we can interact with eBay by using eBaySDK for java within the environment of WebSphere commerce that provides developer the flexibility to work with the outside world of the WebSphere commerce.

For more information ,please dial us on **1-630-355-6292** (Monday - Friday: 8AM to 6PM) , alternatively you can email us at [info@royalcyber.com](mailto:info@royalcyber.com) or for any inquiry [Contact us](#) and our team of consultants will assist you further.

Contact us with your requirements by clicking the link below:  
[WebSphere Commerce Inquiry Form](#)



© Copyright IBM Corporation 2010  
IBM Global Services  
Route 100  
Somers, NY 10589  
U.S.A.  
Produced in the United States of America  
08-10  
All Rights Reserved

IBM, the IBM logo, [ibm.com](http://ibm.com), Lotus®, Rational®, Tivoli®, DB2® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml) Other company, product and service names may be trademarks or service marks of others. The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software. This document illustrates how one organization uses IBM products. Many factors have contributed to the results and benefits described; IBM does not guarantee comparable results elsewhere.