Politecnico di Milano

Software Engineering 2

Bressan G., de Santis S., Di Marco P.

CODE INSPECTION

Professor Elisabetta Di Nitto

Academic Year 2016-2017

Contents

1 Description of the code inspection

2 Assigned classes

The class assigned for the code inspection is only one and it is:

- ProductsExportToEbay

The path of the class is:

/apache-ofbiz-16.11.01/specialpurpose/ebay/src/main/java/org/apache/ofbiz/ebay/ProductsExportToEbay.java

This class belongs to ofbitz project that an open source product for the automation of enterprise processes that includes framework components and business applications for ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), E-Business / E-Commerce, SCM (Supply Chain Management), MRP (Manufacturing Resource Planning), MMS/EAM (Maintenance Management System/Enterprise Asset Management).

3 Functional role of the classes

This class contains most important funcitonalities about Ebay. As described better in the following description of the methods the main functions of this class are:

- Create an onbject to insert in Ebay
- Create a connection with Ebay
- Create a data information using XML

A new object can be create using an apposite method that is **exportToEbay** that can also help you suggesting you the better price of the item. Once this is done, the class needs to create a connection with Ebay server, using apposite APIs for communicate with it sending string or objec (such XML files). For doing this part there is method **postItem**. Last important fucntion is to create an XML document for one or more items and this is done using **buidDataItemsXml** method, while to create an XML that contains information about categories there is a **buidlCategoriesXml**.

4 Methods of the class

4.1 exportToEbay

This method is used to insert into ebay a new item using also the suggestion give from other ebay object, in fact once given the type of object that you want to insert,

this method create a query to execute into Ebay database and this query returns all item to the object in question filtered by date and take the price from the first of this list. With this price you can have a start price for your item and if there are not error, your item will be post on Ebay database using Post Item method.

4.2 Post item

Post Item method is use to create a connection with Ebay server and to write the data items object into it. So more detailed consideration are that this method create a new connection using HTTP protocol each time that it needs to write somethings into database, and for creating a new connection uses a java classes called HttpURLConnection that return a new object (connector). This object connection in this method is used to set some parameters that are the special keywords that permit to use the api with the server to recognise who are doing this request (to add a new data items) and eventually accept or deny the request. At the end of setting of API string (special keywords), this method using connection.getOutputStream() and outputStream.write(dataItems.toString().getBytes()), write a data items inside the Ebay server and this method can also read the aswer of the Ebay server using connection.getInputStream().

4.3 buildDataItemsXml

The main function of this method is to create an XML file using these three methods:

- Document itemDocument = UtilXml.makeEmptyXmlDocument(”AddItemRequest”);
- Element itemRequestElem = itemDocument.getDocumentElement();
- itemRequestElem.setAttribute(”xmlns”, ”urn:ebay:apis:eBLBaseComponents”);

This XML file is created to record the information about a specified item that will be send to Ebay to sell it.
This method has can also create some XML child node and can add the attribute to it such as:

- Product name
- Title
- Description
- Long description
- Quantity
- Price
- Start price

- Buy it Now
- Image
- Product ID
- Category
- Condition

4.4 buildCategoriesXml

This method is used by one object to this class to add one or more categories to an XML file. So this method take some parameters as input and into an XML file create a new child node with these attributes.

4.5 setPaymentMethodAccepted

This private method sets the the payment method accepted from the website, the context attribute contains the actual value of the payment methods accepted appending in the xml the available one.

4.6 setMiscDetails

This method is used to add details to the item Elements from Ebay

4.7 getEbayCategories

This main function of this method is to query Ebay, using the private postItem() mathod, to get categories from its database that are related in some way with the context, here it is used also the private method buildCategoriesXml() to fill the dataItems buffer with proper values and then to perform the query.

4.8 readEbayCategoriesResponse

This private method is used by the getEbayCategories() after the query on the Ebay's Database to read the response about the categories found. In particular this method receives in input the response of the Database and then fill the result with the categories extract from the query.

4.9 exportToEbayResponse

This method is used to validate the query performed by the exportToEbay() method to insert an item in the database. In particular if the insertion is valid this method returns a ServiceUtil.returnSuccess() in the other case a ServiceUtil.returnFailure().

5 Result of the inspection

In this chapter we'll describe the result of the code inspection made on the class assigned to us. The class inspection includes variables/attribute and method, and we checked all the points of the Code Inspection Assignment Task Description.pdf document that describe what are the things to avoid during the software programming.

5.1 Notation

For this part of the project we decide to use a simple notation to distinguish the different issue found during the code inspection.
The notation is:

- The items of the code inspection checklist will be referred as follows: C1, C2,...
- The specific line of code where there is an issue is indicate with L.123
- To indicate an interval of lines of code we will use the notation L.123-L-456

5.2 Notation

5.2.1 ProductsExportToEbay class

1. C23 JavaDoc for a short description of the means of the constant values and methods is missed.

2. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **resource** must be **RESOURCE** L.59

3. C23 JavaDoc for a short description of the means of the constant values is missed L.60

4. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **configFileName** must be `CONFIG_FILE_NAME` L.59

5. C23 JavaDoc for a short description of the means of the constant values is missed L.61

6. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so textbfmodule must be **MODULE** L.61

7. C1 There is a variable called **response** that is used for two different object in the same class, and more precisely at L.89 and L.156

8. C14 This line doesn't not respect the max length of 120 characters and it's length is 173 characters L.75

9. C14 This line doesn't not respect the max length of 120 characters and it's length is 213 characters L.78

10. C14 This line doesn't not respect the max length of 120 characters and it's length is 185 characters L.80

11. C14 This line doesn't not respect the max length of 120 characters and it's length is 330 characters L.89

12. C14 This line doesn't not respect the max length of 120 characters and it's length is 126 characters L.103

13. C14 This line doesn't not respect the max length of 120 characters and it's length is 182 characters L.336

14. C14 This line doesn't not respect the max length of 120 characters and it's length is 146 characters L.362

15. C1 In the L.135 there is an method invoked by final class that is **verboseOn()**. Its name is not explanatory and during the lecture of this part of the code is not clear what is its function. As consequence of the this method is not also clear what is the real faction of the line L.136 so for these reason the lines of code between L.135 and L.137 must be re-write.

16. C1 In this line of code, there is a method called **postItem**, this method is

reported below to explain better the issue of this line of code L.89

postItem(eBayConfigResult.get(”xmlGatewayUri”).toString(), dataItemsXml, eBayConfigResult.get(”devID”).toString(), eBayConfigResult.get(”appID”).toString(),eBayConfigResult.get(”certID”).toString(), ”AddItem”,eBayConfigResult.get(”compatibilityLevel”).toString(), eBayConfigResult.get(”siteID”).toString());

As we can see this method takes eight parameters:

- eBayConfigResult.get(”xmlGatewayUri”).toString()
- sdataItemsXml
- eBayConfigResult.get(”devID”).toString()
- eBayConfigResult.get(”appID”).toString(),
- eBayConfigResult.get(”certID”).toString()
- AddItem
- eBayConfigResult.get(”compatibilityLevel”).toString()
- eBayConfigResult.get(”siteID”).toString()

but is not clear what devID, appID and certID means. The solution of this problem could be to re-write the name of these parameters to explain better what they indicates.

17. C1 Another issue about postItem method is in line L.133, in which during the SCRITTURA of the method because during the definition of input parameters, devID, appID and certID are replicated (so their means is not already clear) but in addition also the parameter callName is an ambiguous name in this context and there is not a related name during the use of the method.

    In buildDataItemsXml(); method:

18. C13 Some lines of this method exceed 80 characters limit, for readablity purpose it’s better to split each line: L. 213 L. 224 L. 230 L. 233 L. 235 L. 236 L. 237 L. 251 L. 263 L. 268 L. 286 L. 287 L. 244 L. 297 L. 306 L. 307 L. 311 L. 315 L. 316 L. 317 L. 318 L. 319 L. 320 L. 321 L. 326 L. 330.

19. C14 Some lines of this method exceed 120 characters limit, for readablity purpose it’s bettere to split each line: L. 238 L. 247 L. 249 L. 253 L. 255 L. 259 L. 261 L. 267 L. 270 L. 272 L. 298 L. 327 L. 331

20. There are some replication of code instead this method, for example in line 210, performs the requesting for minimum price of an item, this is done in some other pieces of code, both in this method and in this class, changing only the minimum with maximum return value. For readability and in order to make a more clear code, it's better to create a method that performs this query on a Product in both minimum and maximum way.

21. Code block that starts in line 247 and ends in line 251 is replicated below.

22. There is a logic error in the formulation if then else that starts at Line 246. briefly it does this:

```
if(b) then
A;
else
A;
B;
```

This can be modified in:

```
A;
if(b) then
nothing
else
B;
```

Or maybe:

```
A;
if(!b) then
B;
```

23. This method is entirely covered by two try catch block, one into the other, and the deeper one contains 140 lines.

24. In this Method there are too much nested if, it's better to divide if and refract conditions.

25. C19 in line 323 there is a block of commented-out lines of code and doesn't contain a date when it will be removed.

26. C18 buildDataItemsXml(); is not commented adequately, of course it requires to be explained, since it's about 160 lines of code.

In setPaymentMethodAccepted() method:

27. C18: The comments above the methods do not explain adequately what the block of code is doing, there are no verbs in the sentences, this happen on L435,L443,L438,L447,L451,L455,L459,L463,L467,L471,L475,L479,L483

28. C13:L420: line length exceed 120 characters

29. C44:From 436 to 485 brute force solution: the long chain of ifs could be replaced with other patterns.

In setMiscDetails() method:

30. C14:L489: line length exceed 120 characters

31. C52: The Exceptions thrown by the method UtilXml.readXmlDocument(customXml) are not catched but simply thrown to the caller

32. C7: L.495: this method receives in input a constant whose name is in lowercase

33. C1:L489: method name is not meaningfull

34. C12:L489: comment required

35. C14:L523,L544,L533,L534: line length exceed 120 characters

In readEbayCategoriesResponse() method:

36. C14: L.523,L533,L534,L544:line length exceed 120 characters

37. C12:L531,L521:Blank lines are useless in this case

38. C33:L532: The declaration of the dataItemsXml does not appear at the beginning of the block
In exportToEbayResponse() method:

39. C14: L.599:line length exceed 120 characters

6 Errors found in other classes

During the code inspection of assigned class, due to the use of objects of other classes we found other "errors" in these class. below the names of classes and their errors are reported.

6.1 ServiceUtil Class

1. C14 This line doesn't not respect the max length of 120 characters and it's length is 200 characters L.116

2. C14 This line doesn't not respect the max length of 120 characters and it's length is 221 characters L.120

6.2 Debug Class

1. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **noModuleModule** must be `NO_MODULE_MODULE`L.36

2. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **emptyParams** must be `EMPTY_PARAMS`L.37

3. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **levelProps** must be `LEVEL_PROPS` L.48

4. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **levelObjs** must be `LEVEL_OBJS` L.49

5. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **levelStringMap** must be `LEVEL_STRING_MAP` L.51

6. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **levelOnCache must** be `LEVEL_ON_CACHE` L.53

7. C7 There is a constant value but doesn't respect the naming convention to use uppercase separated by an underscore so **root** must be **ROOT** L.55

7 Changelog

This the 1.0 version: initial release

8 Hours of work

For this Part of the project we spent 20 hours per person, and we divided the class to inspect in three different parts.