	8+ Condividi	4	Altro	Blog successivo»

Code Of The Day

Syed Muhammad Humayun - Senior Developer, United States

$60/hr ★★★★★ (5.0)
75 oDesk Hours|31 Contracts
http://www.smhumayun.com

13 years of industry experience have helped me understand the customer's perspective a lot and with that... more

Hire me on **oDesk**

Google+ Followers

Syed Muhammad Humayun

Add to circles
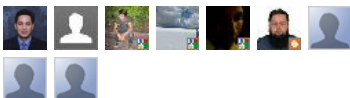
643 have me in circles   View all

FeedBurner FeedCount

Subscribe to our Feed :   **22 readers**
BY FEEDBURNER

Followers

Join this site
with Google Friend Connect

Members (9)

Already a member? Sign in

Cloud

4d (1) actian (1) action (1) adabas d (1) adaptable (1) alfresco (1) algorithm (1) altibase (1) **annotation** (2) anonymous class (1) **apache** (4) **api** (9) architecture (1) as/400 (1) ase (1) association (1) aster ncluster (1) asynchronous (1) bean (1) blogger (1) cache (2) certificate (3) cleardb (1) cloud (2) code reuse (1) connectorj (1) constructor (1) controller (1) convention (1) copyright (1) **cryptography** (4) csql (1) cubrid (1) daffodil (1) daffodildb (2) **database** (8) db (1) db2 (1) default port (8) derby (1) descriptor (1) design patterns (1) design principle (1) django (1) **driver class** (8) drupal (2) dynamodb (1) eclipse (1) embarcadero (1) **embedded** (3) emc (1) empress (1) enterprisedb (1) **example** (7) export (1) exportcert (1) externalize (1) **facebook** (2) filemaker (1) filter (1) firebird (1)

Saturday, 6 July 2013

Paypal Button and Instant Payment Notification (IPN) Integration with Java

DZone   1   0

Integration with Payment Gateways is one of the most common integration in business applications. In this post, I will demonstrate how you can integrate your Java (Servlet/JSP/Struts/Spring/etc.) applications with Paypal using Paypal Button and Instant Payment Notification (IPN).

For those who hate to read long documentation, I've tried to capture "minimum" reading material here from Paypal website that is required to understand Paypal IPN Protocal and Architecture; and which is a pre-requisite for our exampl below.

Besides that, you should also setup following (if not done already):

1. Create Paypal Account
2. Activate Paypal Developer Account by signing in with your Paypal Account credentials
3. Create Paypal Payment Button

Create PayPal payment button

PayPal payment buttons are an easy way to accept payments. Check the PayPal Payments Standard Overview for more information.

Use this page to customize your button and create the HTML you'll need to copy and paste into your website. Learn more.

Having trouble viewing this page?

Step 1: Choose a button type and enter your payment details

Choose a button type   Which button should I choose?
Buy Now ▾

Item name   Item ID (optional) What's this?

Price   Currency
               USD ▾ Need multiple prices?

Customize button   Your customer's view

☐ Add drop-down menu with price/option Example
☐ Add drop-down menu Example   **Buy Now**
☐ Add text field Example

▸ Customize text or appearance (optional)

Shipping
Use specific amount: USD Help

Tax
Use tax rate   %

Merchant account IDs Learn more
◉ Use my secure merchant account ID
○ Use my primary email address smhumayun@gmail.com

Step 2: Track inventory, profit & loss (optional)
Step 3: Customize advanced features (optional)

Create Button

'IpnHandler.java'

```java
1  /**
2   * Paypal Button and Instant Payment Notification (IPN) Integration with Java
3   * http://codeoftheday.blogspot.com/2013/07/paypal-button-and-instant-payment_6.ht
4   */
5  package smhumayun.codeoftheday.PaypalIpnExample;
6
7  import javax.net.ssl.HttpsURLConnection;
8  import javax.servlet.http.HttpServletRequest;
9  import java.io.BufferedReader;
10 import java.io.InputStreamReader;
11 import java.io.PrintWriter;
12 import java.net.URL;
13 import java.net.URLEncoder;
14 import java.util.Enumeration;
15 import java.util.Iterator;
16 import java.util.Map;
17 import java.util.logging.Level;
18 import java.util.logging.Logger;
19
20 /**
21  * Paypal IPN Notification Handler Class
22  *
23  * User: smhumayun
24  * Date: 7/6/13
25  * Time: 5:48 PM
26  */
27 public class IpnHandler
28 {
29
30     private Logger logger;
31     private IpnConfig ipnConfig;
32     private IpnInfoService ipnInfoService;
33
34     /**
35      * This method handles the Paypal IPN Notification as follows:
36      *      1. Read all posted request parameters
37      *      2. Prepare 'notify-validate' command with exactly the same parameters
38      *      3. Post above command to Paypal IPN URL {@link IpnConfig#ipnUrl}
39      *      4. Read response from Paypal
40      *      5. Capture Paypal IPN information
41      *      6. Validate captured Paypal IPN Information
42      *          6.1. Check that paymentStatus=Completed
43      *          6.2. Check that txnId has not been previously processed
44      *          6.3. Check that receiverEmail matches with configured {@link IpnCo
45      *          6.4. Check that paymentAmount matches with configured {@link IpnCo
46      *          6.5. Check that paymentCurrency matches with configured {@link Ipn
47      *      7. In case of any failed validation checks, throw {@link IpnException}
48      *      8. If all is well, return {@link IpnInfo} to the caller for further bu
49      *
50      * @param request {@link HttpServletRequest}
51      * @return {@link IpnInfo}
52      * @throws IpnException
53      */
54     public IpnInfo handleIpn (HttpServletRequest request) throws IpnException {
55         logger.info("inside ipn");
56         IpnInfo ipnInfo = new IpnInfo();
57         try
58         {
59             //1. Read all posted request parameters
60             String requestParams = this.getAllRequestParams(request);
61             logger.info(requestParams);
62
63             //2. Prepare 'notify-validate' command with exactly the same parameter
64             Enumeration en = request.getParameterNames();
65             StringBuilder cmd = new StringBuilder("cmd=_notify-validate");
66             String paramName;
67             String paramValue;
68             while (en.hasMoreElements()) {
69                 paramName = (String) en.nextElement();
70                 paramValue = request.getParameter(paramName);
71                 cmd.append("&").append(paramName).append("=")
72                         .append(URLEncoder.encode(paramValue, request.getParameter
73             }
74
75             //3. Post above command to Paypal IPN URL {@link IpnConfig#ipnUrl}
76             URL u = new URL(this.getIpnConfig().getIpnUrl());
77             HttpsURLConnection uc = (HttpsURLConnection) u.openConnection();
78             uc.setDoOutput(true);
79             uc.setRequestProperty("Content-Type", "application/x-www-form-urlencod
80             uc.setRequestProperty("Host", "www.paypal.com");
81             PrintWriter pw = new PrintWriter(uc.getOutputStream());
82             pw.println(cmd.toString());
83             pw.close();
```

Pages

- Home
- Privacy Policy
- Terms & Conditions

```java
 84
 85            //4. Read response from Paypal
 86            BufferedReader in = new BufferedReader(new InputStreamReader(uc.getInp
 87            String res = in.readLine();
 88            in.close();
 89
 90            //5. Capture Paypal IPN information
 91            ipnInfo.setLogTime(System.currentTimeMillis());
 92            ipnInfo.setItemName(request.getParameter("item_name"));
 93            ipnInfo.setItemNumber(request.getParameter("item_number"));
 94            ipnInfo.setPaymentStatus(request.getParameter("payment_status"));
 95            ipnInfo.setPaymentAmount(request.getParameter("mc_gross"));
 96            ipnInfo.setPaymentCurrency(request.getParameter("mc_currency"));
 97            ipnInfo.setTxnId(request.getParameter("txn_id"));
 98            ipnInfo.setReceiverEmail(request.getParameter("receiver_email"));
 99            ipnInfo.setPayerEmail(request.getParameter("payer_email"));
100            ipnInfo.setResponse(res);
101            ipnInfo.setRequestParams(requestParams);
102
103            //6. Validate captured Paypal IPN Information
104            if (res.equals("VERIFIED")) {
105
106                //6.1. Check that paymentStatus=Completed
107                if(ipnInfo.getPaymentStatus() == null || !ipnInfo.getPaymentStatus
108                    ipnInfo.setError("payment_status IS NOT COMPLETED {" + ipnInfo
109
110                //6.2. Check that txnId has not been previously processed
111                IpnInfo oldIpnInfo = this.getIpnInfoService().getIpnInfo(ipnInfo.g
112                if(oldIpnInfo != null)
113                    ipnInfo.setError("txn_id is already processed {old ipn_info "
114
115                //6.3. Check that receiverEmail matches with configured {@link Ipn
116                if(!ipnInfo.getReceiverEmail().equalsIgnoreCase(this.getIpnConfig(
117                    ipnInfo.setError("receiver_email " + ipnInfo.getReceiverEmail(
118                            + " does not match with configured ipn email " + this.
119
120                //6.4. Check that paymentAmount matches with configured {@link Ipn
121                if(Double.parseDouble(ipnInfo.getPaymentAmount()) != Double.parseD
122                    ipnInfo.setError("payment amount mc_gross " + ipnInfo.getPayme
123                            + " does not match with configured ipn amount " + this
124
125                //6.5. Check that paymentCurrency matches with configured {@link I
126                if(!ipnInfo.getPaymentCurrency().equalsIgnoreCase(this.getIpnConfi
127                    ipnInfo.setError("payment currency mc_currency " + ipnInfo.get
128                            + " does not match with configured ipn currency " + th
129            }
130            else
131                ipnInfo.setError("Inavlid response {" + res + "} expecting {VERIFI
132
133            logger.info("ipnInfo = " + ipnInfo);
134
135            this.getIpnInfoService().log(ipnInfo);
136
137            //7. In case of any failed validation checks, throw {@link IpnExceptio
138            if(ipnInfo.getError() != null)
139                throw new IpnException(ipnInfo.getError());
140        }
141        catch(Exception e)
142        {
143            if(e instanceof IpnException)
144                throw (IpnException) e;
145            logger.log(Level.SEVERE, e.toString(), e);
146            throw new IpnException(e.toString());
147        }
148
149        //8. If all is well, return {@link IpnInfo} to the caller for further busi
150        return ipnInfo;
151    }
152
153    /**
154     * Utility method to extract all request parameters and their values from requ
155     *
156     * @param request {@link HttpServletRequest}
157     * @return all request parameters in the form:
158     *                                        param-name 1
159     *                                          param-value
160     *                                        param-name 2
161     *                                          param-value
162     *                                          param-value (in case of mu
163     */
164    private String getAllRequestParams(HttpServletRequest request)
165    {
166        Map map = request.getParameterMap();
```

```java
167             StringBuilder sb = new StringBuilder("\nREQUEST PARAMETERS\n");
168             for (Iterator it = map.keySet().iterator(); it.hasNext();)
169             {
170                 String pn = (String)it.next();
171                 sb.append(pn).append("\n");
172                 String[] pvs = (String[]) map.get(pn);
173                 for (int i = 0; i < pvs.length; i++) {
174                     String pv = pvs[i];
175                     sb.append("\t").append(pv).append("\n");
176                 }
177             }
178             return sb.toString();
179         }
180
181         public Logger getLogger() {
182             return logger;
183         }
184
185         public void setLogger(Logger logger) {
186             this.logger = logger;
187         }
188
189         public IpnConfig getIpnConfig() {
190             return ipnConfig;
191         }
192
193         public void setIpnConfig(IpnConfig ipnConfig) {
194             this.ipnConfig = ipnConfig;
195         }
196
197         public IpnInfoService getIpnInfoService() {
198             return ipnInfoService;
199         }
200
201         public void setIpnInfoService(IpnInfoService ipnInfoService) {
202             this.ipnInfoService = ipnInfoService;
203         }
204 }
```

IpnHandler.java hosted with ❤ by **GitHub**   view raw

As for 'IpnConfig.java':

1. 'ipnUrl' should be:
    1. For Production/Live - https://www.paypal.com/cgi-bin/webscr
    2. For Sandbox/Testing - https://www.sandbox.paypal.com/cgi-bin/webscr
2. 'receiverEmail' should be the Paypal Account Email
3. 'paymentAmount' should be the 'Price' you setup while defining Paypal Button above.
4. 'paymentCurrency' should be the 'Currency' you mention while defining the Paypal Button above.

Now,

- Test your IPN Listener using Paypal's IPN Simulator
- And, Setup IPN Notifications on Paypal

COMPLETE SOURCE CODE IS AVAILABLE HERE TO DOWNLOAD

DZone   1   0

Posted by Syed Muhammad Humayun at 18:26   g+1 +4 Consiglialo su Google

Labels: instant payment notification, integration, ipn, java, jsp, payment gateway, paypal, servlet, source code, spring struts

3 comments

Add a comment as fiorenzo pizza

Top comments

Syed Muhammad Humayun shared this via Google+ 4 months ago - Shared publicly
+1   [1] · Reply

Vu Nguyen 1 week ago - Shared publicly
very nice, thank you for sharing.

[1] · Reply

Rehan Rana 4 months ago
humayun, This is good affords. Nice one. Rehan Rana

Newer Post   Home   Older Po

Subscribe to: Post Comments (Atom)