**(By Ranavir Dash)**

1 – Link from Android device browser to open the Application

( Use a new scheme )

* In web page href give the link to a unique scheme.

Ex. <a href=”bluegreen://stl.com”>click me</a>

* Give the handler for that scheme in your android application activity with intent filter

Ex.

<**activity  
 android:name=".SplashActivity"  
 android:label="@string/app\_name"  
 android:exported="true"**> //This activity can rcv msgs from outside of the application  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 <**intent-filter**>  
 <**data android:scheme="bluegreen"** />  
 <**action android:name="android.intent.action.VIEW"** />  
 <**category android:name="android.intent.category.BROWSABLE"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
 </**intent-filter**>  
</**activity**>

2- **SMS Gateway Integration**

(sandeshlive gateway)

* Prerequisite:- Internet connection Availability
* Input:- Destination mobile number, text Message
* Process:-
* The sendSms() method uses Gateway URL,Registration details(User Id,Password,senderName) & Input details to generate a final Request URL.
* The message is to be checked for special characters as it is appended with the URL itself using ScriptEngine present in javax.script package.
* Then then the URL to be requested using Classes of java.net package.
* Output:-

One response value is to be retrieve during request to get the message sending details

Ex-

a:7:{s:6:"msg\_id";s:51:"7467-125889-1339f37-14b290cab44-56e28753-1821569358";s:8:"SenderId";s:6:"STLIND";s:9:"linecount";s:1:"1";s:10:"billcredit";s:4:"1.00";s:7:"message";s:12:"Your+message";s:10:"sendondate";s:19:"2016-03-11 14:22:00";s:6:"seq\_id";a:1:{i:1;a:10:{s:5:"valid";s:4:"true";s:10:"billcredit";s:4:"1.00";s:14:"originalnumber";s:12:"919658602048";s:11:"validnumber";s:13:"+919658602048";s:5:"mnpID";s:3:"645";s:11:"providerkey";s:2:"AC";s:11:"id\_provider";s:1:"1";s:9:"regionKey";s:2:"OD";s:7:"dlr\_seq";i:1;s:12:"simpleNumber";s:13:"+919658602048";}}}

**3- PAYMENT GATEWAY INTEGRATION**

( HDFC PAYMENT GATEWAY )

* **Prerequisite**:-
* Internet connection Availability
* HDFC pg URL
* HDFC account id and secret key

1. **Implementation using Servlet & JSP** :-

* **Input**:-
* Unique application number(\*)
* Amount to pay(\*)
* Billing Address( can be hardcoded inside servlet )
* Delivery Address( can be hardcoded inside servlet )
* Return URL( After successful / failure transaction Merchat URL to be hard coded inside servlet )
* Hashing method( can be MD5/SHA1/SHA512 )
* **Process**:-
* Get all the Above information
* For each payment request generate one order no ( same to be referenced and returned by the payment gateway for reference )
* Arrange all infos in dictionary Ascending order of the keys
* Generate one concatenated string as below

Ex. Secretkey | value1 | value2...etc

* Make the hashing data using the hash method (as **secure\_hash**)
* Save one payment initiation record in database
* Make request for payment to Gateway URL with request parameters as above all values along with the secure\_hash value.
* Then pg website opened for making the payment and after successful / failure transaction one payment Id generated and link available to go to merchant website(same as return URL)
* Then our own response page / servlet called and we can get the response values from PG website request parameters( ex.MerchantRefNo same as order no )
* PG also sends one secure\_hash value using the same hash method of the parameters that it has sent(same process as done by you)
* So first secure\_hash value of the response parameters to be prepared as done for the request parameters
* Then that value matched with the response secure\_hash value if it is same then it is the transaction you are looking for
* Then get the response code as 0/1/2 as success/failure/error etc. And get the status,paymentId etc. And update your payment database with above response details.
* Then you can navigate to your payment details page.

1. **Reusing above implementation in Android WebView** :-

* Give same input application\_no and amount to pay value and press pay button
* As request parameter give the above two values (optionally address details), then post the request to /initiatePayment Servlet in a WebView.
* Then the entire process of web can be done inside WebView then after clicking the return to merchant URL override the /PostPayment.jsp URL and move to your any response details activity.
* In your response activity send request to a servlet with application number and amount that your have requested for payment and get the details of the payment and show it cheerfully.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*