Happiness Meter

Web Widget Integration and Development Guide

Version 2.0

This document is classified as “**Information Classification Label**”

Document Control

Document History

Date	Version	Author(s)	Description
20-11-2016	2.0	Muhammad Akif	

Distribution List

Name	Title	Entity
Jameel Wasfat Ahmed	Smart Solutions Development Section Manager	
Hend Saqer Al Nuaimi	Senior eServices Provisioning Officer	
Ahmad Ali Shobaki	Principal Systems Developer	

Approval List

Date	Name	Title	Signature

Table of Contents

1 Introduction

1.1 About This Guide

This document describes and explains the necessary development activities required by Service Providers (Government Departments) to integrate with the new “Happiness Meter Systems” by adding the Happiness Meter Web Widget into their portals and applications.

The audience of this guide are Developers, Software Specialist and Software Application vendors who are assigned to integrate Service Providers’ existing web applications and portals with the Happiness Meter System using the Web Widget developed by Smart Dubai Government.

1.2 Purpose

This guide helps the implementer to understand the required development actions and steps required to integrate with the Happiness Meter System from web applications and mobile portals.

1.3 Prerequisites

Before proceeding with this Development Guide you should have a good knowledge of the Java programming language or the Microsoft .Net Framework

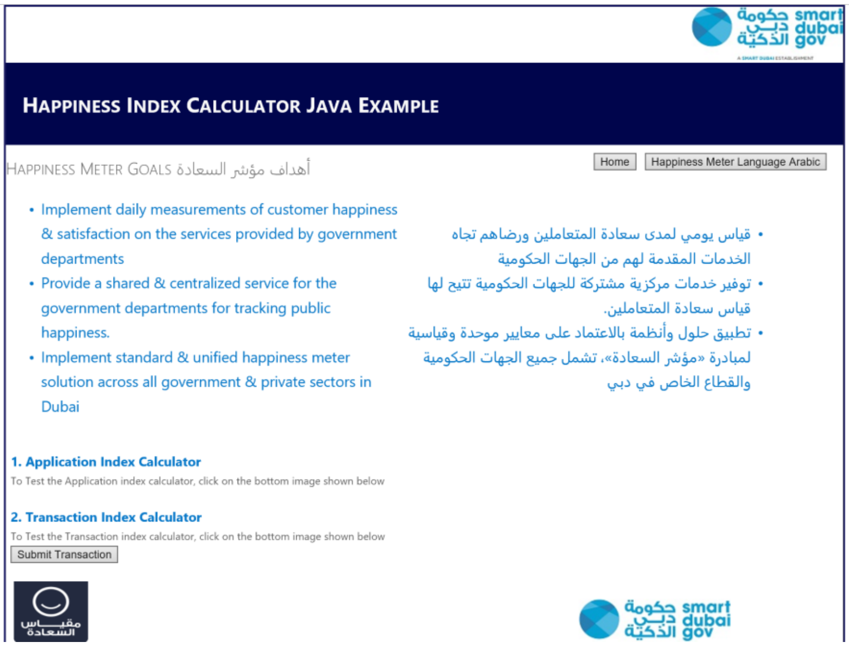
You should be familiar with Web Applications and Online Services Development using the Java JEE or the .Net Platforms with basic understating to the HTTP "Hyper Text Transfer Protocol".

2 Sample Application

2.1 How it works?

This section describes the workflow and how the sample application is working.

The Sample application stars at the Home Page of the Department.



HAPPINESS INDEX CALCULATOR JAVA EXAMPLE

HAPPINESS METER GOALS أهداف مؤشر السعادة

- Implement daily measurements of customer happiness & satisfaction on the services provided by government departments
- Provide a shared & centralized service for the government departments for tracking public happiness.
- Implement standard & unified happiness meter solution across all government & private sectors in Dubai

- قياس يومي لمدى سعادة المتعاملين ورضاهم تجاه الخدمات المقدمة لهم من الجهات الحكومية
- توفير خدمات مركزية مشتركة للجهات الحكومية تتيح لها قياس سعادة المتعاملين.
- تطبيق حلول وأنظمة بالاعتماد على معايير موحدة وقياسية لمبادرة «مؤشر السعادة»، تشمل جميع الجهات الحكومية والقطاع الخاص في دبي

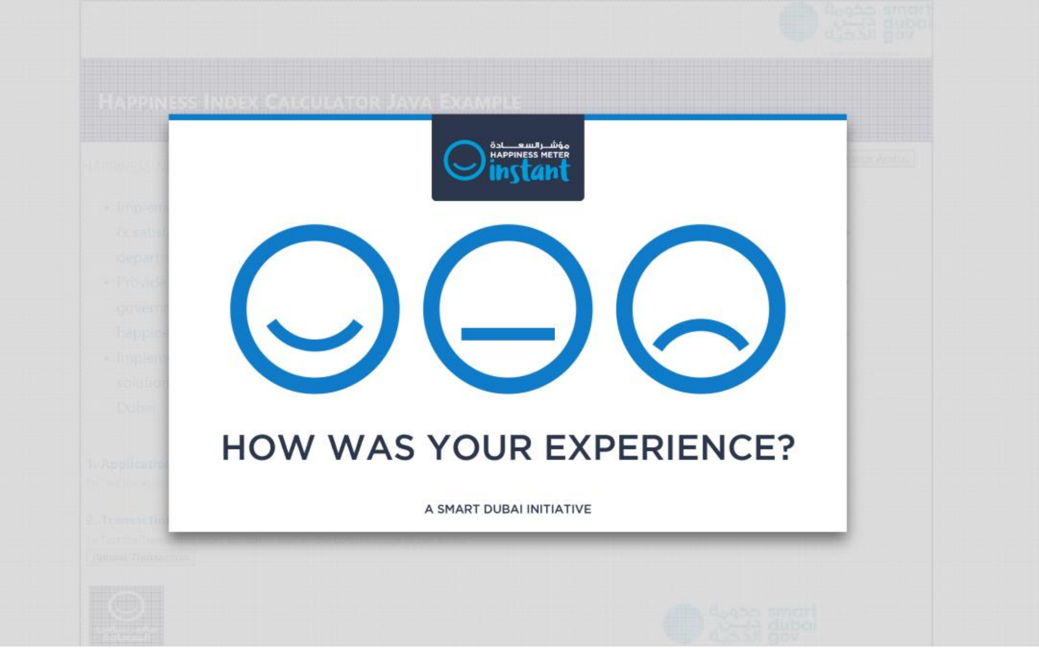
1. Application Index Calculator

To Test the Application index calculator, click on the bottom image shown below

2. Transaction Index Calculator

To Test the Transaction index calculator, click on the bottom image shown below

Once the end user clicks on the "Happiness Gadget Button" (i.e. this button must be added to all pages in the department portal), which is located at bottom corner of the page, a model dialogue window under which the survey/happiness form will be opened

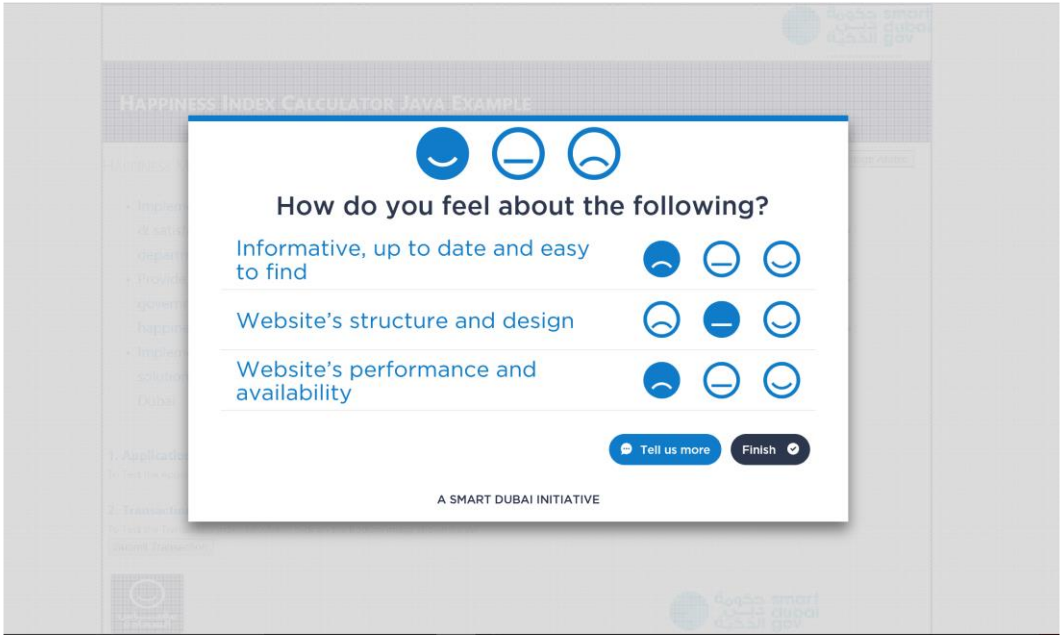


HOW WAS YOUR EXPERIENCE?

A SMART DUBAI INITIATIVE

Once the user select the desired face icon based on his experience.

Survey page appears having some questions regarding the service and department.



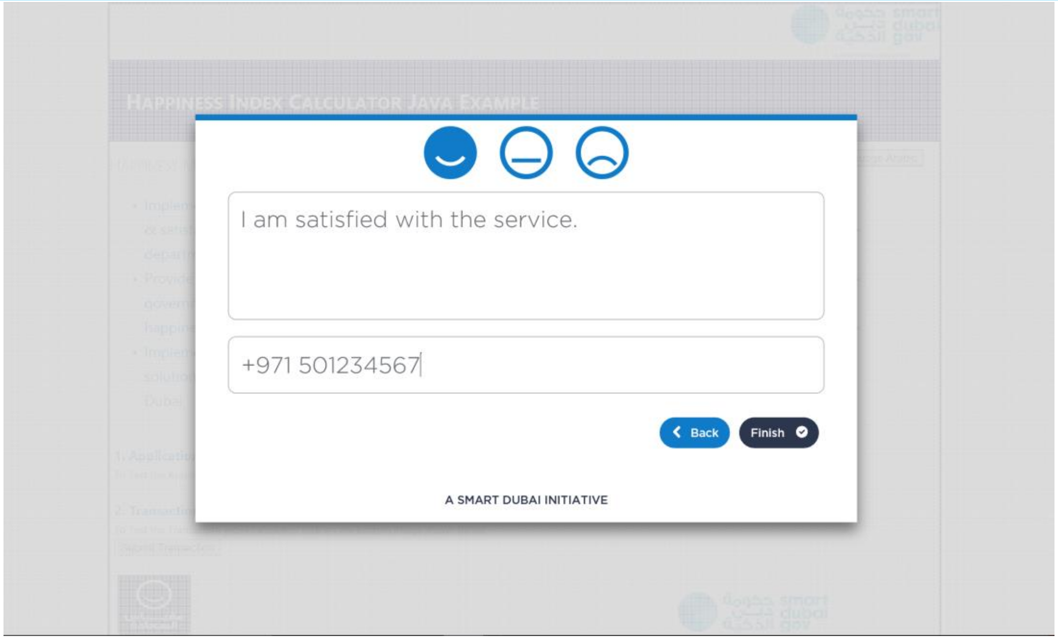
How do you feel about the following?

Informative, up to date and easy to find	●	○	○
Website's structure and design	○	●	○
Website's performance and availability	●	○	○

A SMART DUBAI INITIATIVE

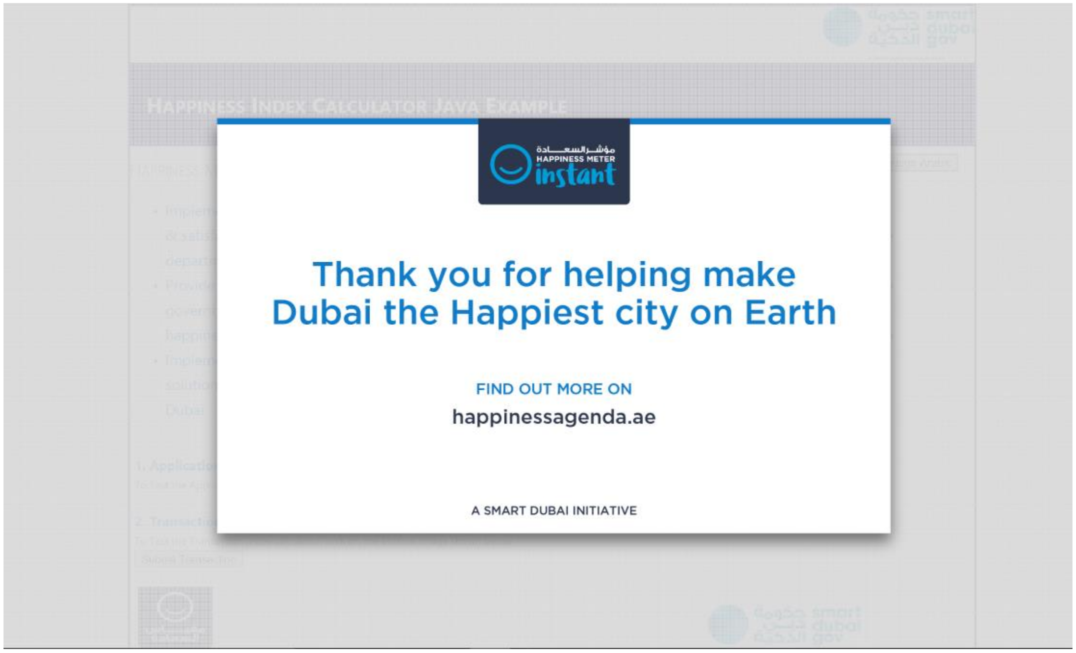
User has to fill the desired response against each asked survey options.

He / She may enter some comments and write it down mobile number for any correspondence in the below screen, when user clicked on “Tell us more” button.


I am satisfied with the service.

+971 501234567

Once the survey has been completed, user can Press “Finish” button to complete it or can press “Back” to change the survey options.

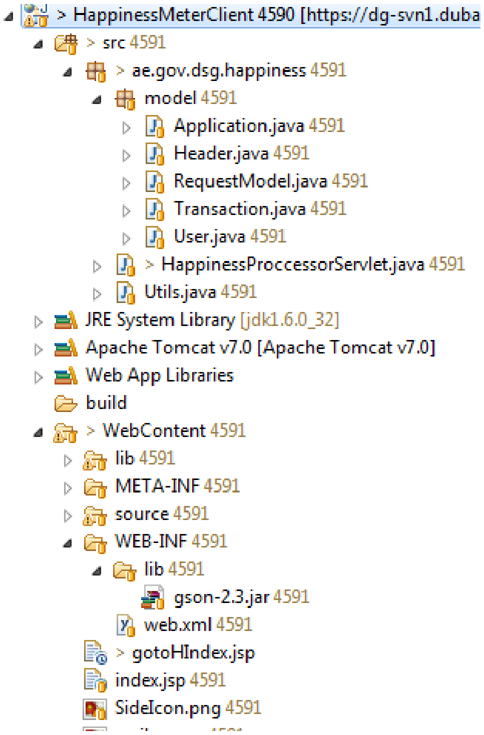

Thank you for helping make Dubai the Happiest city on Earth

FIND OUT MORE ON happinessagenda.ae

User can navigate out of the survey dialog by clicking any place outside the shown dialog.

2.2 Sample Application Building Blocks “Java”

The Sample Application is a J2EE Application that contains has the following structure, We are using eclipse to build this sample application and this is how the Web application is shown the Project Explorer view from the JEE perspective.



```
◢ > HappinessMeterClient 4590 [https://dg-svn1.duba
   ◢ > src 4591
      ◢ > ae.gov.dsg.happiness 4591
         ◢ model 4591
            ▷ Application.java 4591
            ▷ Header.java 4591
            ▷ RequestModel.java 4591
            ▷ Transaction.java 4591
            ▷ User.java 4591
         ▷ > HappinessProccessorServlet.java 4591
         ▷ Utils.java 4591
   ▷ JRE System Library [jdk1.6.0_32]
   ▷ Apache Tomcat v7.0 [Apache Tomcat v7.0]
   ▷ Web App Libraries
     build
   ◢ > WebContent 4591
      ▷ lib 4591
      ▷ META-INF 4591
      ▷ source 4591
      ◢ WEB-INF 4591
         ◢ lib 4591
              gson-2.3.jar 4591
           web.xml 4591
        > gotoHIndex.jsp
        index.jsp 4591
        SideIcon.png 4591
```

Next we will explain the main building blocks

2.2.1 Index Page (i.e. Home Page)

The Index page is the Welcome page of the application, this is the page where the end user agent is redirected when he access the Sample application URL http://[IP]:[Port]/demo/

In order to add to your page you need to do the following:

1. Copy lib, source, and SideIcon.png to your application
2. On the webpage head section “<head>” add / update the following code

```
 <!-- Add jQuery library -->

<script type="text/javascript"
src="https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/source/jquery-
1.10.1.min.js"></script>

      <script type="text/javascript"
src="https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/source/jquery.
fancybox.js?v=2.1.5"></script>

      <link rel="stylesheet" type="text/css"
href="https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/source/jquery
.fancybox.css?v=2.1.5" media="screen" />
```

3. Add the following HTML code to your webpage body.

```
 <div id="foo">

      <a class="fancybox fancybox.iframe"
   href="<%=request.getContextPath()%>/HappinessProccessor?type=2"><img
   src="Images/SideIcon.png" border="0" alt=""></a>
```

Note :-

When clicking the ☺ button, application should post to local URL, that will be responsible of populating the request parameters and sign the request using the department secret key.

2.2.2 Happiness Processor Servlet

1. Sample application comes with a sample servlet “ae.gov.dsg.happiness. HappinessProccessorServlet” that will handle populating the request parameters and sign it.
2. In order to integrate this servlet in your application, please follow the following steps:
   a. Copy the source folder to your application source folder
   b. From sample application copy WEB-INF/lib/gson-2.3.jar, gotoHIndex.jsp to your application
   c. Add the following servlet mapping to your “WEB-INF/web.xml” file

```
     <servlet>

       <servlet-name>HappinessProccessorServlet</servlet-
   name>

       <servlet-
   class>ae.gov.dsg.happiness.HappinessProccessorServlet</se
   rvlet-class>

     </servlet>

     <servlet-mapping>

       <servlet-name>HappinessProccessorServlet</servlet-
```

3. The above servlet will populate the following request parameters:

Parameter Name	Description
json_payload	Request payload in JSON (i.e. this value must be URL encoded)
client_id	Service provider code
signature	Calculated signature for the request
lang	This will be the survey language accepted values are (en | ar)
random	16 digit random number (i.e. see below sample code) (I.e. please refer to Appendix C - Security Requirements)
timestamp	System Timestamp (DD/MM/YYYY HH24:MI:SS) always in UTC time zone (i.e. please refer to Appendix C - Security Requirements)
nonce	Secure value calculated for each request (i.e. please refer to Appendix C - Security Requirements)

themeColor	e.g. #0000ff [service provider theme color to be provided in request]

```
      public static String generate16DigitRandom() {
            SecureRandom rand = new SecureRandom();
            long accumulator = 1 + rand.nextInt(9);

// ensures that the 16th digit isn't 0

            for (int i = 0; i < 15; i++) {
                  accumulator *= 10L;

                  accumulator += rand.nextInt(10);

            }
```

Sample code to generate random number

```
      public static String hash(String text) {
            MessageDigest md;

            try {

                  md = MessageDigest.getInstance("SHA-512");
                  md.update(text.getBytes());

                  byte byteData[] = md.digest();

                  // convert the byte to hex format method 1
                  StringBuffer sb = new StringBuffer();

                  for (int i = 0; i < byteData.length; i++) {
                        sb.append(Integer.toString((byteData[i] &

0xff) + 0x100, 16).substring(1));

                  }
```

Sample code to calculate hash value

After the servlet populate the parameters it will redirect the request to a post back page (i.e. gotoHIndex.jsp) which will submit the final request for the survey, the following code shows this page content.

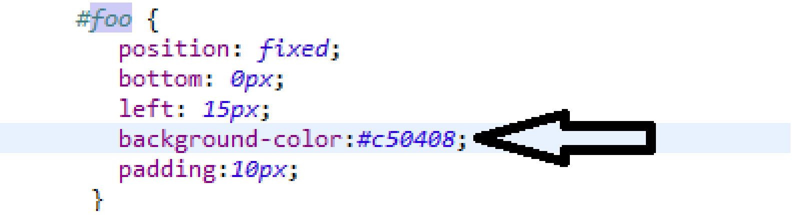
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<%@ page pageEncoding="UTF-8"%>
<html>
<head>
</head>
<body>
<form action="
https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/PostDataService"
method="post">
```

```
            <input type="hidden" name="json_payload"
value="<%=request.getAttribute("json")%>" />
            <input type="hidden" name="client_id"
value="<%=request.getAttribute("clientId")%>"/>
            <input type="hidden" name="signature"
value="<%=request.getAttribute("signature")%>"/>
            <input type="hidden" name="lang"
value="<%=request.getAttribute("lang")%>"/>
            <input type="hidden" name="random"
value="<%=request.getAttribute("random")%>"/>
            <input type="hidden" name="timestamp"
value="<%=request.getAttribute("timestamp")%>"/>
            <input type="hidden" name="nonce"
value="<%=request.getAttribute("nonce")%>"/>


       </form>

 </body>

 <script type="text/javascript">
```

2.2.3 Change the default implementation

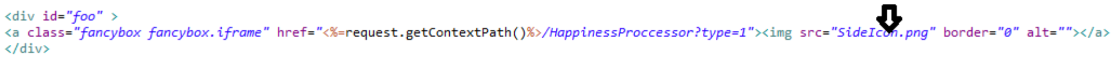
Sample application shipped with default implementation, so if you to change the existing behavior (i.e. change secret key, client_id, etc.), please do the following:

If you want to change the background color of image go to source\jquery.fancybox.css file and go to the end of file and change background-color value.



```
#foo {
   position: fixed;
   bottom: 0px;
   left: 15px;
   background-color:#c50408;
   padding:10px;
 }
```

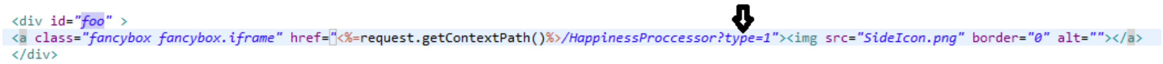
1- May be you need to change path of the image icon if the image and the jsp not in the same folder for example change it to ../common/SideIcon.png



```
<div id="foo" >
<a class="fancybox fancybox.iframe" href="<%=request.getContextPath()%>/HappinessProccessor?type=1"><img src="SideIcon.png" border="0" alt=""></a>
</div>
```

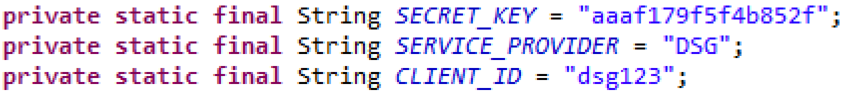
2- By default happiness feedback will be at application level if you want to do transaction based feedback.

You have to change the type of feedback in the needed footer pages to 1



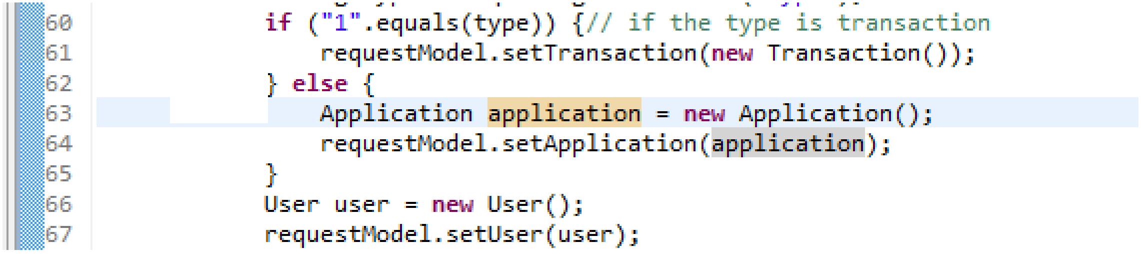
```
<div id="foo" >
<a class="fancybox fancybox.iframe" href="<%=request.getContextPath()%>/HappinessProccessor?type=1"><img src="SideIcon.png" border="0" alt=""></a>
</div>
```

Goto ae.gov.dsg.happiness.HappinessProccessorServlet class and change below values Set service_provider to your name and for secret_key and client_id you will receive new values for them from us.



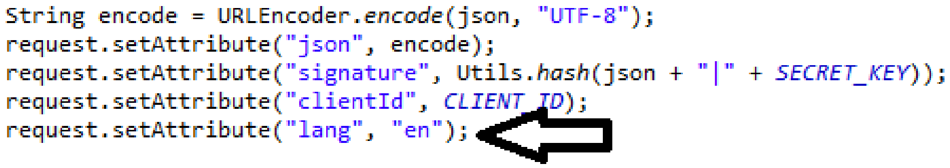
```
private static final String SECRET_KEY = "aaaf179f5f4b852f";
private static final String SERVICE_PROVIDER = "DSG";
private static final String CLIENT_ID = "dsg123";
```

3- In the same class you have set the values needed for the feedback based on your application



```
        if ("1".equals(type)) {// if the type is transaction
            requestModel.setTransaction(new Transaction());
        } else {
            Application application = new Application();
            requestModel.setApplication(application);
        }
        User user = new User();
        requestModel.setUser(user);
```

You can change the text language by setting either **ar** or **en**



```
String encode = URLEncoder.encode(json, "UTF-8");
request.setAttribute("json", encode);
request.setAttribute("signature", Utils.hash(json + "|" + SECRET_KEY));
request.setAttribute("clientId", CLIENT_ID);
request.setAttribute("lang", "en");
```

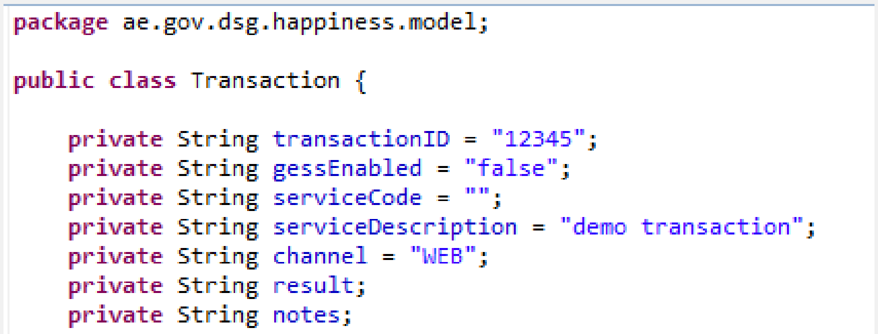
2.2.4 Application Configuration

You have to change default values inside Application, Transaction and Users classes based on your logic except result field keep it empty.

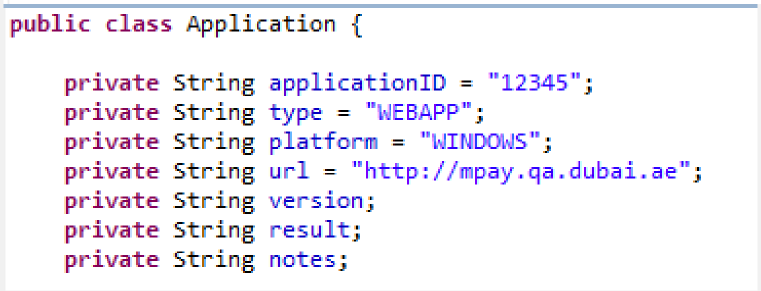
Default Values

```
       private String applicationID = "12345";
        private String type = "WEBAPP";
       private String platform = "WINDOWS";
        private String url = "http://mpay.qa.dubai.ae";
        private static final String SECRET_KEY = "aaaf179f5f4b852f";
        private static final String SERVICE_PROVIDER = "DSG";
        private static final String CLIENT_ID = "dsg123";
        private String transactionID = "12345";
        private String gessEnabled = "false";
        private String serviceCode = "";
        private String serviceDescription = "demo transaction";
```

```
       private String channel = "WEB";
```



```
package ae.gov.dsg.happiness.model;

public class Transaction {

    private String transactionID = "12345";
    private String gessEnabled = "false";
    private String serviceCode = "";
    private String serviceDescription = "demo transaction";
    private String channel = "WEB";
    private String result;
    private String notes;
```

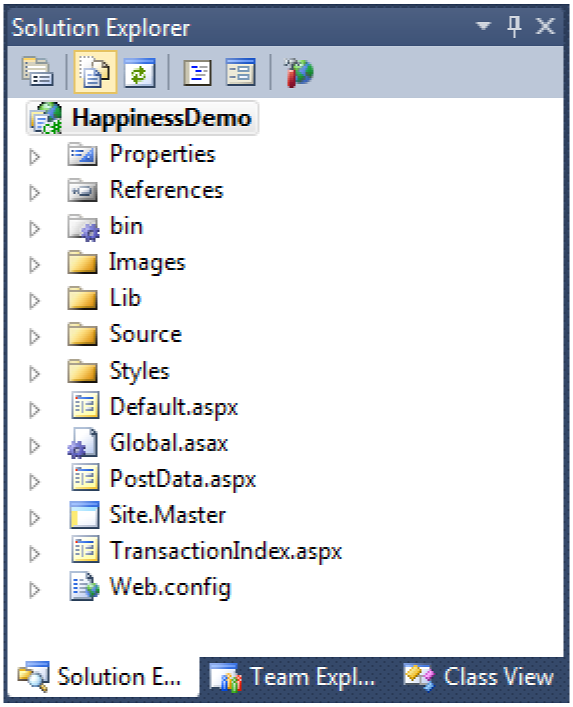


```
public class Application {

    private String applicationID = "12345";
    private String type = "WEBAPP";
    private String platform = "WINDOWS";
    private String url = "http://mpay.qa.dubai.ae";
    private String version;
    private String result;
    private String notes;
```

2.3 Sample Application “.NET”

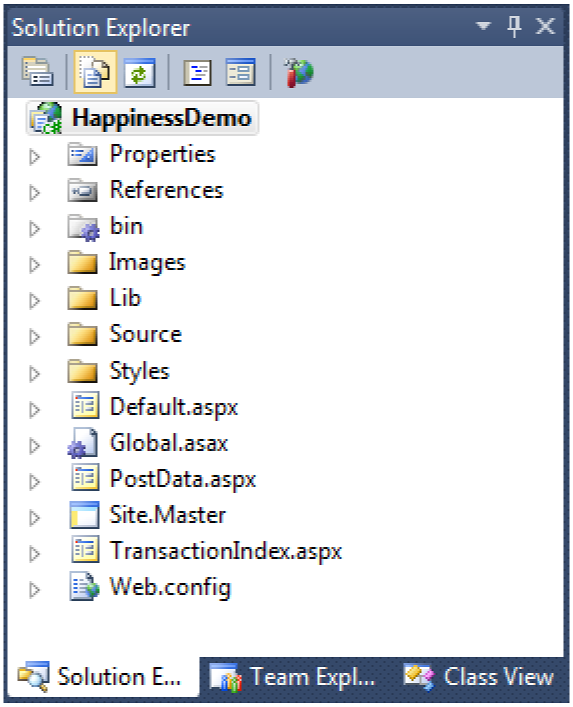
2.3.1 Application Structure

The sample application is developed in VS 2010 using c#. Below is the application structure. You need to configure the application on IIS server using .net 4 application pools. Set the Default.aspx as home page of the application.



Solution Explorer

HappinessDemo
- Properties
- References
- bin
- Images
- Lib
- Source
- Styles
- Default.aspx
- Global.asax
- PostData.aspx
- Site.Master
- TransactionIndex.aspx
- Web.config



2.3.2 Files overview

Below are the description of files/folders that requires to integrate in the application

File/Folder Name	Required	Description
Images	Yes	It contains the image SideIcon.png which is showing the smiley icon at the bottom of the page. It requires to copy the image folder and keep in the similar project hierarchy.
Lib	Yes	It contains the JS files and requires copying the
Source	Yes	It requires copying this folder and keeping in the similar level in the integrated application. These Libraries should be included on master page.
Styles	Optional	It contain CSS file, and used only in the sample application.

Default.aspx	Optional	It is the starting page of the application and is used to measure the application level happiness index.
PostData.aspx	Yes	This page is used to compose the web service and post the data to the server. You need to copy the page in your root directory of the application without any modification. It is required to set the language parameter in this page. Happiness index page open in two languages depending upon the value of the integrated application. You need to set “en” for English and “ar” for Arabic.
TransactionIndex.aspx	Optional	This page is used to measure the Transaction happiness index in the application.
Site.Master	Yes	Call JavaScript and libraries files on master page, If application shared the Master page. In this case, it is not required to call the libraries file on each transaction and application pages.
Web.config	Yes	All the parameter values are defined in the configuration file.

2.3.3 Master Page Format (Transaction/Application)

Below code is required to add / update in the Master Page of the application. It should be added in the <head> </head> section of the master page. It calls the required JavaScript libraries and CSS file.

to show the happiness div. You can change the position and appearance of smiley icon by modifying the #foo CSS class.

Master page format

```
        <head>
        <title>Happiness Widget</title>
         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
         <link href="Styles/Site.css" rel="stylesheet" type="text/css" />

         <!-- Add jQuery library -->
         <script type="text/javascript"
         src="https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/source /jquery-1.10.1.min.js">
         </script>
         <script type="text/javascript"
         src="https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/source/jquery.fancybox.js?v=2.1.5">
         </script>
         <link rel="stylesheet" type="text/css"
href=" https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/source/jquery.fancybox.css?v=2.1.5"
media="screen" />

         <script type="text/javascript">
           $(document).ready(function () {
              $('.fancybox').fancybox();
           });

    // get the page name
              var  pageName  =  (function  ()  {  var  a  =
              window.location.href,            b            =
              a.lastIndexOf("/");
              return a.substr(b + 1); } ());

    // it will open the happiness div on transaction page. You can specify the transaction page name
           function autoClick() {
              if (pageName == 'TransactionIndex.aspx') { document.getElementById('onload').click();
              }
    }
         </script>
         <style type="text/css">
                  .fancybox-custom .fancybox-skin { box-
                           shadow: 0 0 50px #222;
                  }

                  body {
                           /*max-width: 700px;*/ margin: 0
                           auto;
```

```
                    #foo {

                              position:
                              fixed;
                              bottom:
                              0px;  left:
                              15px;

                              background-
                              color:#c50408;
                              padding:10px;

                     }

          </style>

      </head>
```

Below code need to add anywhere in the body tag of the master page. It will always take the transaction type as application. It will show the smiley icon on left side of the page.

Master Page Format

```
 <body>
   <div id="foo"><a class="fancybox fancybox.iframe"
href="PostData.aspx?type=application"><img src="Images/SideIcon.png" border="0" alt=""
/></a></div>
</body>
```

2.3.4 Transaction index

You need to add the below code in all your Transaction pages or where you want to calculate the transaction index in your application. It will show the smiley icon on left side of the page. When user clicks, it will open the PostData.aspx page passing the type as Transaction.

Transaction Index format

Transaction index calculator requires information related to Transaction and users, who performed this transaction. So you have to pass the available values to the PostData.aspx page. Below you can find the sample data adding in the TransactionIndex.aspx.cs. You need to replace with the actual transaction values.

```
 <body>
```

Transaction/ User sample data

```
protected void Page_Load(object sender, EventArgs e)
        {
         // Service Provider unique Transaction ID
            Session["transactionID"] = "1001";

         // Below parameter need to set from the logged in User.
            Session["emiratesID"] = "789-1998-87689898";


         // Username in case available
            Session["username"] = "admin";

         // Email in case available Session["email"]
         = "test@test.com";

// User Mobile in case available and Format should be “9715XXXXXXXX” Session["mobile"]
= "971500000000";  }
```

2.3.5 Application Configuration

Below keys are required to set the values in the configuration file of the application. You need to copy the below keys in the web.config file of your application.

Configuration settings

```
protected void Page_Load(object sender, EventArgs e)
        {
         // Service Provider unique Transaction ID
            Session["transactionID"] = "1001";

         // Below parameter need to set from the logged in User.
            Session["emiratesID"] = "789-1998-87689898";


         // Username in case available
            Session["username"] = "admin";

         // Email in case available Session["email"]
         = "test@test.com";

// User Mobile in case available and Format should be “9715XXXXXXXX” Session["mobile"]
= "971500000000";  }
```

3 Appendix A – Request Parameters

The following table provides a description of all parameters.

KeyName	Values
serviceProvider	DSG will provide the value
client_id	DSG will provide the value
PostURL QA	https://happinessmeterqa.dubai.gov.ae/HappinessMeter2/PostDataService
PostURL PROD	https://happinessmeter.dubai.gov.ae/HappinessMeter2/PostDataService
SecretKey	DSG will provide the value
serviceCode	DSG will provide the value
channel	WEB
source	User Source Any of LOCAL | MYID | ANONYMOUS Where LOCAL to be used with Departments Local User Profile.
serviceDescription	Provide the detail of your service
applicationID	Provide the name of your application
gessEnabled	Indicates if the department service is exist on GESS system or not Allowed Values {true, false}
Applicationtype	Provide the type of your application like WEBAPP|SMARTAPP|DESKTOP
platform	Any of IOS|ANDROID|BLACKBERRY|WINDOWS |OTHERS *Filed Required when type is SMARTAPP
Applicationurl	Web application URL
version	Application version
themeColor	e.g. #0000ff

	[service provider theme color to be provided in request]

4 Appendix B – Sample Request Parameters

The following table shows a sample values for each request parameter

KeyName	Values
Parameter Name	Sample Value

json_payload	{
	"user":
	{"source":"ANONYMOUS"},
	"transaction":
	{"transactionID":"12345", "gessEnabled":"false", "serviceCode":"", "serviceDescription":"demo transaction", "channel":"WEB"},
	"header":
	{"timestamp":"22/01/201508:19:06", "serviceProvider":"DSG"}
	}
	Note:- In actual request you should encode json_payload using URLEncoder

client_id	dsg123
signature	b13e873f22e537f7978ef420f16d864a40633200ca45bb8797d058c9953c87f868 aa20ce287f399f8bf16086f87bb0eb680a8900d5890896672104facc40d6fc
lang	en

5 Appendix C – System Parameters

DSG will provide the following credential for every service provider; the Service provider will use these credentials from Happiness Index Web gadget and Smart Application SDK.

1.Service Provider Code

2. Username

3. Service Provider Secret

The following http headers should be passed to access the Happiness Index APIs

Timestamp, The client's current system timestamp using the following format, DD/MM/YYYY HH24:MI:SS (i.e. UTC timezone)

Random, this is a 16 Hexadecimal digits String

Nonce, This header is used to prevent reply attacks and is calculated by using the following formula

Nonce=HASH("SHA512", Random header| Timestamp header| Service Provider Secret)

Signature , This header is used to assure data integrity and that Message body was not tempered during transmission "Man in the middle attack" and calculated by the using the following formula

Signature= HASH("SHA512", Message Body | Service Provider Secret)

Example

A client with the following credentials "Provided by DSG to the client" want to invoke the API on 30/12/2014 12:25:11

Service Provider Code: DEWA

Username (i.e. Client_ID): dewa_user_happiness_index

Service Provider Secret: 123FA34CD1223

5.1 How to calculate the Timestamp http header

The Timestamp http header is equal to the client current system

date Timestamp: 30/12/2014 12:25:11

5.2 How to calculate the Random http header

The Random http header is a 16 Hexadecimal digits random string

Random: random (“HEX”, 16 digits) Random: CD35545FDAB33CDF

5.3 How to calculate the Nonce http header

Nonce:
HASH(“SHA512”, Random header| Timestamp header| Service Provider Secret)
Nonce:
HASH(“SHA512”,”CD35545FDAB33CDF|30/12/2014 12:25:11|123FA34CD1223”)
Nonce:
6283DA2466B9C4A7085CD7A6BF7942AD7B854F33233132AB4734A2B48DB8F223
68993373479308642F39B004B7B010A124A844A2CF04DC5B0C0859B0F6F8C5DB

5.4 How to calculate the signature http parameter

Signature: HASH(“SHA512”, Message Body | Service Provider Secret)