



## PesaPal Integration - .Net

### 1. Add reference to PesaPal.API.dll

Download the version of **PesaPal.APIHelper.dll** most suitable for you from the downloads section at <http://developer.pesapal.com/how-to-integrate/samples-downloads>.

### 2. Create a page to display the PesaPal payment processing form

Example: Payment.aspx

You can now embed PesaPal directly in your site, providing a seamless experience to your customers.

You can do this by inserting an *iframe* on the page on your site customers land on when they checkout:

```
<iframe src="<%=GetPesapalUrl() %>" width="100%" height="620px" frameborder="0" scrolling="auto" />
```

In the code behind file, **Payment.aspx.cs**, add the method `GetPesapalUrl()`. This is where most of the work happens.

```
using Pesapal.API;
using System.Linq;

...

protected string GetPesapalUrl()
{
    Uri pesapalPostUri = new Uri("http://demo.pesapal.com/API/PostPesapalDirectOrderV4");
    /* change to https://www.pesapal.com/API/PostPesapalDirectOrderV4 when you
       are ready to go live! */
    Uri pesapalCallbackUri = new Uri(/* link to the page on your site users will be
       redirected to when the payment process has been completed */);

    // Setup builder
    IBuilder builder = new APIPostParametersBuilderV2()
        .ConsumerKey(/* Your PesaPal Consumer Key
                       Register a merchant account on demo.pesapal.com and use the
                       merchant key for testing. When you are ready to go live make
                       sure you change the key to the live account registered on
                       www.pesapal.com!*/)
        .ConsumerSecret(/* Your PesaPal Consumer Secret
                       Use the secret from your test account on demo.pesapal.com.
                       When you are ready to go live make sure you change the
                       secret to the live account registered on www.pesapal.com! */)
        .OAuthVersion(EOAuthVersion.VERSION1)
        .SignatureMethod(ESignatureMethod.HMACSHA1)
        .SimplePostHttpMethod(EHttpMethod.GET)
        .SimplePostBaseUri(pesapalPostUri)
        .OAuthCallbackUri(pesapalCallbackUri);

    // Initialize API helper
```



```
APIHelper<IBuilder> helper = new APIHelper<IBuilder>(builder);

// Populate line items
var lineItems = new List<LineItem> { };

// For each item purchased, add a lineItem.
// For example, if the user purchased 3 of Item A, add a line item as follows:

var lineItem =
    new LineItem
    {
        Particulars = /* description of the item, example: Item A */,
        UniqueId = /* some unique id for the item */,
        Quantity = /* quantity (number of items) purchased, example: 3 */,
        UnitCost = /* cost of the item (for 1 item) */
    };

lineItem.SubTotal = (lineItem.Quantity * lineItem.UnitCost);
lineItems.Add(lineItem);

// Do the same for additional items purchased

...

// Compose the order
PesapalDirectOrderInfo webOrder = new PesapalDirectOrderInfo()
{
    Amount = (lineItems.Sum(x => x.SubTotal)).ToString(),
    Description = /* [required] description of the purchase */,
    Type = "MERCHANT",
    Reference = /* [required] a unique id, example: an order number */,
    Email = /* [either email or phone number is required]
             email address of the user making the purchase */,
    FirstName = /* [optional] user's first name */,
    LastName = /* [optional] user's last name */,
    PhoneNumber = /* [either email or phone number is required]
                  user's phone number */,
    LineItems = lineItems
};

// Post the order to PesaPal, which upon successful verification,
// will return the string containing the url to load in the iframe
return helper.PostGetPesapalDirectOrderUrl(webOrder);
}
```

### 3. Create a page to display after the user has completed the payment process on PesaPal.

Example: PaymentBeingProcessed.aspx

Note: this is the same page whose url was specified in the `pesapalCallBackUri` parameter in the `GetPesapalUrl()` method above.

PesaPal will redirect to this page with the following query parameters:



**pesapal\_merchant\_reference:** this is the reference (a unique order id), that you passed to PesaPal when posting the transaction.

**pesapal\_transaction\_tracking\_id:** this is the unique tracking id for this transaction on PesaPal.

```
string reference = Request.QueryString["pesapal_merchant_reference"];
string pesapal_tracking_id = Request.QueryString["pesapal_transaction_tracking_id"];
```

Store the *pesapal\_tracking\_id* in your database against the order matching the *reference* as you will need it for subsequently querying for the status of the payment from PesaPal.

## 4. Query for the Status of a Transaction

Once a transaction has been posted to PesaPal, you can query for its status using the following API methods:

**QueryPaymentStatus**

**QueryPaymentStatusByMerchantRef**

**Query PaymentDetails**

You will have to query PesaPal periodically for the status of the payment until you no longer receive a PENDING status.

### **QueryPaymentStatus**

```
string reference = /* this is the Reference you sent PesaPal when posting a transaction
                    (Note: this has to be unique for each transaction) */
string pesapal_tracking_id = /* this is the tracking id returned by pesapal when you
                              posted a transaction */

// Setup builder
IBuilder builder = new APIPostParametersBuilder()
    .ConsumerKey(WebManager.ConsumerKey)
    .ConsumerSecret(WebManager.ConsumerSecret)
    .OAuthVersion(EOAuthVersion.VERSION1)
    .SignatureMethod(ESignatureMethod.HMACSHA1)
    .SimplePostHttpMethod(EHttpMethod.GET)
    .SimplePostBaseUri("http://demo.pesapal.com/api/querypaymentstatus"); /*When you
    are ready to go live change to https://www.pesapal.com/api/querypaymentstatus*/

// Initialize API helper
APIHelper<IBuilder> helper = new APIHelper<IBuilder>(builder);

try
{
    // query pesapal for status >> format of the result is
    // pesapal_response_data=<PENDING|COMPLETED|FAILED|INVALID>
    string result = helper.PostGetQueryPaymentStatus(pesapal_tracking_id, reference);
```



```
string[] resultParts = result.Split(new char[] { '=' });
string paymentStatus = resultParts[1]; /* Possible values:
                                         PENDING, COMPLETED, FAILED or INVALID*/
}
catch (Exception ex)
{
    // Handle error
}
```

The following result will be returned to you:

**pesapal\_response\_data =<PENDING|COMPLETED|FAILED|INVALID>**

### ***QueryPaymentStatusByMerchantRef***

Same as *QueryPaymentStatus* above, but *pesapal\_tracking\_id* is not required.

```
string reference = /* this is the Reference you sent PesaPal when posting a transaction
                    (Note: this has to be unique for each transaction) */

// Setup builder
IBuilder builder = new APIPostParametersBuilder()
    .ConsumerKey(WebManager.ConsumerKey)
    .ConsumerSecret(WebManager.ConsumerSecret)
    .OAuthVersion(EOAuthVersion.VERSION1)
    .SignatureMethod(ESignatureMethod.HMACSHA1)
    .SimplePostHttpMethod(EHttpMethod.GET)
    .SimplePostBaseUri("http://demo.pesapal.com/api/querypaymentstatusbymerchantref");
    /*When you are ready to go live change to
    https://www.pesapal.com/api/querypaymentstatusbymerchantref */

// Initialize API helper
APIHelper<IBuilder> helper = new APIHelper<IBuilder>(builder);

try
{
    // query pesapal for status >> format of the result is
    // pesapal_response_data=<PENDING|COMPLETED|FAILED|INVALID>
    string result = helper.PostGetQueryPaymentStatus(reference);

    string[] resultParts = result.Split(new char[] { '=' });
    string paymentStatus = resultParts[1]; /* Possible values:
                                             PENDING, COMPLETED, FAILED or INVALID*/
}
catch (Exception ex)
{
    // Handle error
}
```



The following result will be returned to you:

**pesapal\_response\_data =<PENDING|COMPLETED|FAILED|INVALID>**

### ***QueryPaymentDetails***

Same as QueryPaymentStatus, but additional information is returned.

```
string reference = /* this is the Reference you sent PesaPal when posting a transaction
                    (Note: this has to be unique for each transaction) */
string pesapalTrackingId = /* this is the tracking id returned by PesaPal when you
                           posted a transaction */

// Setup builder
IBuilder builder = new APIPostParametersBuilder()
    .ConsumerKey(WebManager.ConsumerKey)
    .ConsumerSecret(WebManager.ConsumerSecret)
    .OAuthVersion(EOAuthVersion.VERSION1)
    .SignatureMethod(ESignatureMethod.HMACSHA1)
    .SimplePostHttpMethod(EHttpMethod.GET)
    .SimplePostBaseUri("http://demo.pesapal.com/api/querypaymentdetails");
    /*When you are ready to go live change to
    https://www.pesapal.com/api/querypaymentdetails */

// Initialize API helper
APIHelper<IBuilder> helper = new APIHelper<IBuilder>(builder);

try
{
    /* query pesapal for status >> format of the result is
       pesapal_response_data= pesapalTrackingId,paymentMethod,paymentStatus,reference
    */

    string result = helper.PostGetQueryPaymentDetails(pesapalTrackingId, reference);

    string[] resultParts = result.Split(new char[] { '=' });
    string paymentDetails = resultParts[1];
    string[] paymentDetailsParts = result.Split(new char[] { ',' });
    string paymentMethod = resultParts[1]; /* example, MPESA, VISA, etc.
    string paymentStatus = resultParts[2]; /* Possible values:
                                           PENDING, COMPLETED, FAILED or INVALID*/
}
catch (Exception ex)
{
    // Handle error
}
```

The following result will be returned to you:

**pesapal\_response\_data= pesapalTrackingId,paymentMethod,paymentStatus,reference**



**pesapalTrackingId:** this is the same as the parameter you sent when making the query

**paymentMethod:** the payment method used by the user to make the payment

**paymentStatus:** one of <PENDING|COMPLETED|FAILED|INVALID>

**reference:** this is the same as the parameter you sent when making the query