Bank Alfalah

Alfa Payment Gateway

Merchant Integration Guide V 1.1

Table of Contents

Overview

Alfa Payment Gateway (APG) provides simple and seamless integration which is designed to enhance customer experience of paying online while keeping security standards as a corner stone. Keeping this in view APG offers its merchants to collect payments online via three payment methods

a. Alfa Wallet

b. Alfalah Bank Account

c. Credit/Debit Card.

Modes of Integration

Following are the modes of integration respective to the payment methods offered

Payment Method	Mode Of Integration
Alfa Wallet	REST API (Redirection also supported)
Alfalah Bank Accounts	REST API (Redirection also supported)
Credit/Debit Card	Redirection

Key Points:

- For Alfa Wallet payment method merchants can use REST API's to collect payments without redirecting their customers from their Platforms to APG checkout page.

- For Alfalah Bank account payment method merchants can use REST API's to collect payments
  without redirecting their customers from their Platforms to APG checkout page.

- For Credit/Debit Card payment method merchants will have to redirect their customers to APG secure checkout page from their respective platform to collect payments.

Getting Started

Please find below a step by step explanation of how you can start the integration process with APG Sandbox

Once your request for APG is approved, you will receive a system generated email from Bank Alfalah. This email contains your Username and a One-time password (OTP) along with a link to your merchant portal. Following are the steps you need to perform on receiving the said email.

a. Go to the Merchant portal URL

b. Enter your username

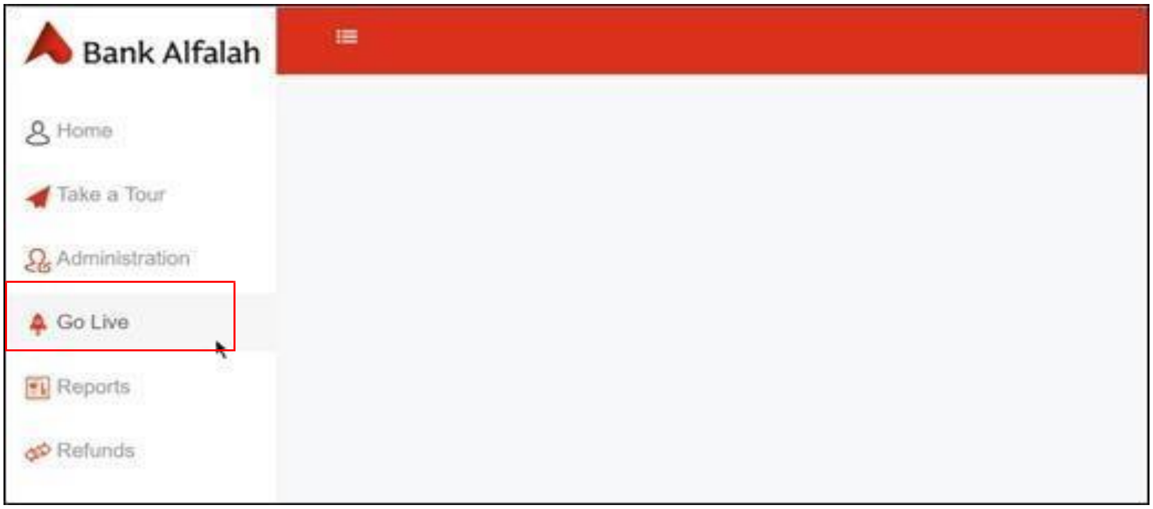
c. Enter OTP and set a new password

d. Login to the Merchant portal

Now that you have logged in your APG merchant account follow the steps below

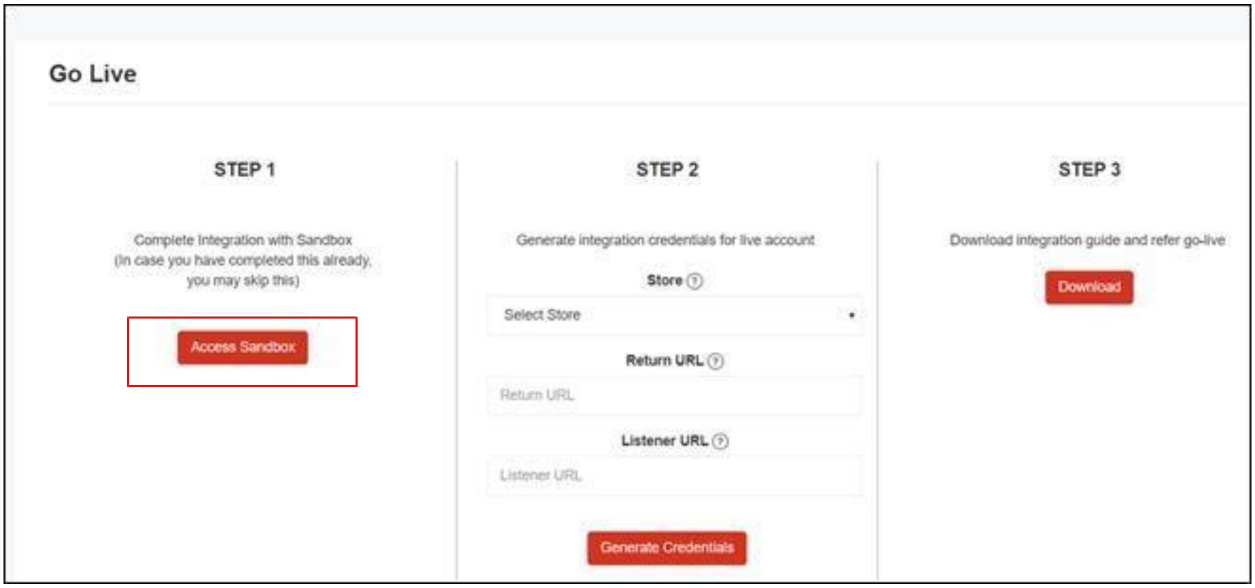
Step 1:

Click on “Go Live” button on the left side menu



Step 2:

Click on “Access Sandbox” button and follow the instruction provided



Go Live

STEP 1

Complete Integration with Sandbox
(In case you have completed this already, you may skip this)

STEP 2

Generate integration credentials for live account

Store: Select Store

Return URL: Return URL

Listener URL: Listener URL

STEP 3

Download integration guide and refer go-live

Step 3:

Once you have accessed your sandbox. You will have to generate your credentials.

Once done, go to Integration>Page Redirection>Scroll down to the end to find the HTML Body in which all of your integration credentials are given.

Security Features to be Enabled from Merchant:

- Merchants website should follow the https protocol in order to complete integration with APG
- Enable your URL Referrer, this can mostly be done by contacting your Hosting service. Your URL Referrer should be the same as your Return URL (set in merchant portal in step 3 above)

  You can get more information on this by visiting the link below.

  https://www.pixelstech.net/article/1561789561-A-mini-guide-to-HTTP-referer

REST API for Alfa Wallet and Alfa Bank Account Payment Method

APG provides REST based API integration for Alfa Wallet and Alfa Bank account integrations. This allows the merchants to collect payments from within their platforms and not having to redirect customer to APG checkout page. This greatly enhances the customer experience of paying online and is recommended to be used for the said payment methods by our tech team.

Integrating with REST API’s involve calling three API’s in the order given below

1. **Initiate Handshake**: Creates a connection b/w Merchant platform and APG

2. **Initiate Transaction Request**: Creates a Request from merchant’s platform to APG for creating a transaction

3. **Process Transaction Request**: Creates a Request from merchant’s platform to APG for processing payment

Initiate HandshakeRequest URL

Following request URL is to be used to POST request parameters given in the next section

Staging Environment:

URL: https://sandbox.bankalfalah.com/HS/api/HSAPI/HSAPI

Production Environment:
URL: https://payments.bankalfalah.com/HS/api/HSAPI/HSAPI Request Parameters:

Field Name	Description	O/M	Possible Values
HS_ChannelId	Channel id of the integration channel. API = 1002	M	1002
HS_MerchantId	Unique Merchant id issued by APG to each merchant. Found in Merchant portal	M	String
HS_StoreId	Unique Store id issued by APG to each merchant Found in Merchant portal	M	String
HS_ReturnURL	Return URL of the merchant on which customer will be redirected after payment success/failure.	M	String
HS_MerchantHash	Unique Merchant hash generated by system for each merchant credentials. Found in merchant portal	M	String
HS_MerchantUsername	Unique Merchant Username generated by system for each merchant credentials. Found in Merchant portal	M	String
HS_MerchantPassword	Unique Merchant Password generated	M	String

	by system for each merchant credentials. Found in Merchant portal		
HS_TransactionReferenceNumber	Unique Order reference number generate by merchant for each order/transaction	M	String
HS_RequestHash	Generated Encrypted Hash of the request. See Encryption method in the section below	M	String

How to get RequestHash Parameter Value?

RequestHash is an important parameter which is being used in all API's and Redirection calls. This parameter is being used by APG as an assurance that no data tampering has been made while communicating with merchant Platform.

This parameter requires encryption whose implementation is explained below.
Step 1. Get the field name from the map

```
var mapString = '', hashName = 'RequestHash';
```

Step 2. Create map of all the fields that are part of the request by passing the form name in the submitRequest(formName) function and assign it to the defined variable:

```
$("#" + formName +" :input").each(function ()
{
if ($(this).attr('id') != '')
{
mapString += $(this).attr('id') + '=' + $(this).val() + '&';

}
});
```

Step 3. This will create a string in following format.

```
HS_ChannelId=10027&HS_MerchantId=197&HS_StoreId=000001&HS_ReturnURL:http://sample.com/SampleP
age&HS_MerchantHash=olqgSxwnlSgwFFtjDXqrshxR6zl782RvXkJud3PelFYnY/WGnDucmQzrHHAvnzbDqiqbFID
ek9moI/UAlACgUdQPZCIiFMJa&HS_MerchantUsername=tyoiqwy&HS_MerchantPassword=m8Kqw@8Nep+Yr
aUg==&HS_TransactionReferenceNumber=a100
```

Step 4. Use AES/CBC/PKCS7Padding algorithm to encrypt the string produced in the previous step

```
$("#" + hashName).val(CryptoJS.AES.encrypt(CryptoJS.enc.Utf8.parse(mapString.substr(0, mapString.length -
1)), CryptoJS.enc.Utf8.parse($("#Key1").val()),
{
keySize: 128 / 8,

iv: CryptoJS.enc.Utf8.parse($("#Key2").val()), mode: CryptoJS.mode.CBC,
padding: CryptoJS.pad.Pkcs7

}));
}
```

Sample Request of Initiate Handshake API:

```
{
"HS_ChannelId": "1002",

"HS_MerchantId": "197",
"HS_StoreId": "000001",
"HS_ReturnURL": " http://sample.com/SamplePage ",
"HS_MerchantHash":"olqgSxwnlSgwFFtjDXqrshxR6zl782RvXkJud3PelFYnY/WGnDucmQzrHHAvnzb
DqiqbFIDek9moI/UAlACgUdQPZCIiFMJa", "HS_MerchantUsername": "tyoiqwy",
"HS_MerchantPassword": "m8Kqw@8Nep+YraUg==", "HS_TransactionReferenceNumber": "a100",
"HS_RequestHash":"BgfY8+t5rCC+oXhl5mrLuJ434+kb+76t6Ju6w1gXPvjWNjHMIltfC4N1M26WWV
ZrElutixOLePTwqarYEMOiDRxkMMXuEQRHSE2GFxgtmMYez1YQRa5kCsvsOU4GlRUqElv+BgLkA6uFNWAhq
Sg5Sq7IR59rnKk2MngAUF+6NO3aMkNEx4FtoxjwADfSa2EIFHYdBq5Vxi1DTIa0zfz1OCfB54wB52RvV49wrN
cgfSn0FfK1bC5NkjjGf9UYCjd6BsjuMCLDaN8Fj/gtgI34kCZzP4PKPKI9pqCMOgy3mnkaBDrvn73f3GUTvms95
EyV1ZL1pcz1gxjb7r4pfXOhztWtX28TbtLTCP4HuePdXGAEbzlNrorUDJHvQ3MBXkj8dPQRQA1gANNe2JIgq60
1lwDIadR7tBybf5M/mh846bP7y/hv/KeEx6UBjA9w8sKnsE5LCwqSNGYsnGd6zdJMFA=="
}
```

Sample Response of Initiate Handshake API

Success:

```
{
“success”: "true",
“AuthToken”:
“76t6Ju6w1gXPvjWNjHMIltfC4N1M26WWVZrElutixOLePTwqarYEMOiDRxkMMXuEQRHSE2GFxgt
mMYez1YQRa5kCsvsOU4GlRUqElv+BgLkA6uFNWAhqSg5Sq7IR59rnKk2MngAUF+6NO3aMkNEx4Ft
oxjwADfSa2ElFHYdBq5Vxi1DTla0zfz1OCfB54wB52RvV49wrNcgfSn0FfK1bC5NkjjGf9UYCjd6BsjuMC LDaN8Fj”,
“ReturnURL”: “http://sample.com/SamplePage”
}
```

Failure:

```
{
        "success": "false",
        "AuthToken": "",
        "ReturnURL": "https://Sandbox.com/APITesting", "ErrorMessage": "Invalid Request"



}
```

Initiate Transaction Request URL

Staging Environment:

URL: https://sandbox.bankalfalah.com/HS/api/Tran/DoTran

Production Environment:

URL: https://payments.bankalfalah.com/HS/api/Tran/DoTran

Request Parameters:

Field Name	Description	O/M	Possible Values
ChannelId	Channel id of the integration channel. API = 1002	M	String

MerchantId	Unique Merchant id issued by APG to each merchant	M	String
StoreId	Unique Store id issued by APG to each merchant	M	String
MerchantHash	Unique Merchant hash generated by system for each merchant credentials	M	String
MerchantUsername	Unique Merchant Username generated by system for each merchant credentials	M	String
MerchantPassword	Unique Merchant Password generated by system for each merchant credentials	M	String
ReturnURL	Return URL of the merchant on which customer will be redirected after payment success/failure	M	String
Currency	Currency of the transaction	M	String (Value used will always be PKR)
AuthToken	Auth Token generated by system in response of successful Handshake request	M	String
TransactionTypeId	Transaction type id through which transaction will be performed. Alfa Wallet = 1, Alfalah Account = 2	M	String 1,2,3
TransactionReferenceNumber	Unique Order reference number generate by merchant for each order/transaction	M	String
AccountNumber	Account Number of the customer through which transaction will be committed	M	String

Country	Country of the customer	M	String
EmailAddress	Email address of the customer upon which payment notification will be sent	M	String
MobileNumber	Mobile Number of the customer upon which payment notification will be sent	M	String
RequestHash	Generated Encrypted Hash of the request. See Encryption method in the section below	M	String

How to get RequestHash Parameter Value?

RequestHash is an important parameter which is being used in all API's and Redirection calls. This parameter is being used by APG as an assurance that no data tampering has been made while communicating with merchant Platform.

This parameter requires encryption whose implementation is explained below.

Step 1. Get the field name from the map

```
var mapString = '', hashName = 'RequestHash';
```

Step 2. Create map of all the fields that are part of the request by passing the form name in the submitRequest(formName) function and assign it to the defined variable:

```
$("#" + formName +" :input").each(function ()
{

    if ($(this).attr('id') != '')
    {

        mapString += $(this).attr('id') + '=' + $(this).val() + '&';

    }

});
```

Step 3. This will create a string in following format.

```
ChannelId=1002&MerchantId=170&StoreId=000001&MerchantHasho=lqgSxwnlSgwFFtjDXqrshxR6zI782RvXkJud3PelFYnY/WGnDucmQzrHHAvnzbDqiqbFIDek9moI/UAlACgUdQPZCIiFMJa&MerchantUsername=tyvasy&MerchantPassword=m8KePI8Nep+YrRUrObmaUg==&ReturnURL=http://202.59.254.153/APGQA/Sandbox/PageRedirectionTesting&Currency=PKR&AuthToken=""&TransactionTypeId=1&TransactionReferenceNumber=a100&TransactionAmount=100&AccountNumber=987654321987100&Country=164&EmailAddress=test@test.com&MobileNumber=034512345678
```

Step 4. Use AES/CBC/PKCS7Padding algorithm to encrypt the string produced in the previous step

```
$("#" + hashName).val(CryptoJS.AES.encrypt(CryptoJS.enc.Utf8.parse(mapString.substr(0, mapString.length - 1)), CryptoJS.enc.Utf8.parse($("#Key1").val()),
    {

        keySize: 128 / 8,
        iv: CryptoJS.enc.Utf8.parse($("#Key2").val()),
        mode: CryptoJS.mode.CBC,
        padding: CryptoJS.pad.Pkcs7
    }));
}
```

Sample Request of Initiate Transaction API:

```
{
        "ChannelId":  "1002",
        "MerchantId": "170",
        "StoreId": "000001",
        "MerchantHash":
"olqgSxwnlSgwFFtjDXqrshxR6zl782RvXkJud3PelFYnY/WGnDucmQzrHHAvnzbDqiqbFIDek9moI/UAlACgUdQPZCIiFMJa",
        "MerchantUsername": "tyvasy",
        "MerchantPassword": "m8KePI8Nep+YrRUrObmaUg==",
        "ReturnURL": "http://202.59.254.153/APGQA/Sandbox/PageRedirectionTesting",
        "Currency": "PKR",
        "AuthToken": "",
        "TransactionTypeId": "1",
        "TransactionReferenceNumber": "a100",

        "TransactionAmount": "100",

        "AccountNumber": "987654321987100",

        "Country": "164",

        "EmailAddress": "test@test.com",


         “MobileNumber”: “034512345678”

        "RequestHash":
"3QC1elErnLVID+K11qMWx3ddbh4dmVUoSfqlin3FQBOHFUUB7Dg4AsLeRman8zp2rxz4KVfQSqV8IbZQeq
2i9QI3+jinZnVZOMSnRoL6Za4Y3lBun1n6AZgh3tIPuwplOIDV4Wz5chitXmnf5FkPyyY5XQJ3268i/CUNT/7ztxE
NTuEQ8/9ftNiVEyMMD2GbYcqef1FuX0DBdLPLTRbSoQ6iDc+RdmmOofKoHpPNwWX3wuSgntytwdOkD52D
eD4gJhEjNAUsmUY8DXclTuzBKtPrLjeYluWu987D0tC//EYCfqkr7cvSt+Nf8ZjgWySbiwXT9/nGoSg+sBP6hmIK
7kPb34hkN+QE7WLk4T8tUZXwxpJeXvxUxuzwi/gbjun3zya6MngoaOCVv0i8fOqqGDp4kkQNMfdwl+TqU0R2
1nawmB0Koy2yW5WzLpjuVguN0mVwgyPIDoLdRYUNw7cKtZvuq3MJDGvJBi5aCf1H5zMMvBFH2d/VELqdA
a29n4e7hlLHLEfqWLyTLcESazMnHSxgJLAI/kOveuC1MfEHzaA="

}
```

Sample Response of Initiate Transaction API:

Success:

```
{

        "success" : "true",

        "AuthToken" : "
        3QC1elErnLVID+K11qMWx3ddbh4dmVUoSfqlin3FQBOHFUUB7Dg4AsLeRman8zp2rxz4KV
        fQSqV8IbZQeq2i9QI3+jinZnVZOMSnRoL6Za4Y3lBun1n6AZgh3tIPuwplOIDV4Wz5chitXmnf
        5FkPyyY5XQJ3268i/CUNT/7ztxENTuEQ8/9ftNiVEyMMD2GbYcqef1FuX0DBdLPLTRbSoQ6iD
        c+RdmmOofKoHpPNwWX3wuSgntytwdOkD52DeD4gJhEjNAUsmUY8DXclTuzBKtPrLjeYlu
        Wu987D0tC//EYDBV61wuatK2yCTHi1es1W6WLG4RnzwcoTM920W49whyFqTlbexlk41ra2
        x8kSTg4jhZtn4m6VF5zSEHQnz+FfpYQzC+F04s+IWvjKZQdNfijqskZJcQdV86C8RmSQgDMj
        MEMuIXKo+wJOKR4k2GgYqH2JSTNoplZNMuKK4JOfMwoONTw3uOLcSaWjjL4AEMFfDib/3
        wOf0irrWt9PxcVEg1XclQ72UwpOycyB2nvCCA== ",

        "MerchantId" : "170",

        "StoreId" : "000001",

        "TransactionTypeId" : "1",

        "TransactionReferenceNumber" : "123",

        "order_datetime" : "03-07-2018 09:49:33",
        "HashKey" :
        "wuSgntytwdOkD52DeD4gJhEjNAUsmUY8DXclTuzBKtPrLjeYluWu987D0tC//EYDBV61wuat
        K2yCTHi1es1W6WLG4Rn ",

        "IsOTP" : "true"

}
```

Failure:

```
{

          "success": "false",

          "AuthToken": "",

          "ReturnURL": "https://Sandbox.com/APITesting",

          "ErrorMessage": "Invalid Request"

}
```

Process Transaction RequestURL

Following request URL is to be used to POST request parameters given in the next section

Staging Environment:

URL: https://sandbox.bankalfalah.com/HS/api/ProcessTran/ProTran

Production Environment:

URL: https://payments.bankalfalah.com/HS/api/ProcessTran/ProTran

Request Parameters:

Field Name	Description	O/M	Possible Values
ChannelId	Channel id of the integration channel. API = 1002	M	String
MerchantId	Unique Merchant id issued by APG to each merchant	M	String
StoreId	Unique Store id issued by APG to each merchant	M	String

MerchantHash	Unique Merchant hash generated by system for each merchant credentials	M	String
MerchantUsername	Unique Merchant Username generated by system for each merchant credentials	M	String
MerchantPassword	Unique Merchant Password generated by system for each merchant credentials	M	String
ReturnURL	Return URL of the merchant on which customer will be redirected after payment success/failure	M	String
Currency	Currency of the transaction	M	String
AuthToken	Auth Token generated by system in response of successful Transaction request API	M	String
TransactionTypeId	Transaction type id through which transaction will be performed. Alfa Wallet = 1, Alfalah Account = 2	M	String
TransactionReferenceNumber	Unique Order reference number generate by merchant for each order/transaction	M	String
SMSOTAC	SMS OTAC sent on the Mobile number tagged with the account of the customer. To be an input field for customer on merchant’s interface and passed in this parameter. Will only	O/M	String SMS OTAC of length 4

	be sent in case “IsOTP” flag is “false” in previous call’s response. Mandatory in case Alfalah Account is used.		
EmailOTAC	Email OTAC sent on the Email Address tagged with the account of the customer. To be an input field for customer on merchant’s interface and passed in this parameter. Will only be sent in case “IsOTP” flag is “false” in previous call’s response. Mandatory in case Alfalah Account is used.	O/M	String Email OTAC of length 4
SMSOTP	SMS OTP sent on the Mobile number tagged with the wallet of the customer. To be an input field for customer on merchant’s interface and passed in this parameter. Will only be sent in case “IsOTP” flag is “true” in previous call’s response. Mandatory in case Alfa wallet is used.	O/M	String SMS OTP of length 8
HashKey	Transaction Hash generated by system passed in Transaction request response	M	String
RequestHash	Generated Encrypted Hash of the request. See Encryption method in the section below	M	String

How to get RequestHash Parameter Value?

RequestHash is an important parameter which is being used in all API's and Redirection calls. This parameter is being used by APG as an assurance that no data tampering has been made while communicating with merchant Platform.

This parameter requires encryption whose implementation is explained below.

Step 1. Get the field name from the map

```
var mapString = '', hashName = 'RequestHash';
```

Step 2. Create map of all the fields that are part of the request by passing the form name in the submitRequest(formName) function and assign it to the defined variable:

```
  $("#" + formName +" :input").each(function ()
{
        if ($(this).attr('id') != '')
        {
            mapString += $(this).attr('id') + '=' + $(this).val() + '&';
        }
});
```

Step 3. This will create a string in following format.

ChannelId=1002&MerchantId=170&StoreId=000001&MerchantHasho=lqgSxwnlSgwFFtjDXqrshxR6zl782RvXkJud3PelFYnY/WGnDucmQzrHHAvnzbDqiqbFIDek9moI/UAlACgUdQPZCIiFMJa&MerchantUsername=tyvasy&MerchantPassword=m8KePI8Nep+YrRUrObmaUg==&ReturnURL=http://202.59.254.153/APGQA/Sandbox/PageRedirectionTesting&Currency=PKR&AuthToken=""&TransactionTypeId=1&TransactionReferenceNumber=a100&SMSOTAC=123&EmailOTAC=ABC&SMSOTP=12345678&HashKey=wuSgntytwdOkD52DeD4gJhEjNAUsmUY8DXcl
TuzBKtPrLjeYluWu987D0tC//EYDBV61wuatK2yCTHi1es1W6WLG4Rn

Step 4. Use AES/CBC/PKCS7Padding algorithm to encrypt the string produced in the previous step

```
$("#" + hashName).val(CryptoJS.AES.encrypt(CryptoJS.enc.Utf8.parse(mapString.substr(0,
mapString.length - 1)), CryptoJS.enc.Utf8.parse($("#Key1").val()),

     {

        keySize: 128 / 8,

        iv: CryptoJS.enc.Utf8.parse($("#Key2").val()),

        mode: CryptoJS.mode.CBC,

        padding: CryptoJS.pad.Pkcs7

     }));

   }
```

Sample Request of Process Transaction Request API:

```
{

"ChannelId":  "1002",
"MerchantId": "170",
"StoreId": "000001",

"MerchantHash":
"olqgSxwnlSgwFFtjDXqrshxR6zl782RvXkJud3PelFYnY/WGnDucmQzrHHAvnzbDqiqbFIDek9moI/UAlACgUd
QPZCIiFMJa",
"MerchantUsername": "tyvasy",
"MerchantPassword": "m8KePI8Nep+YrRUrObmaUg==",

"ReturnURL": "http://localhost:58934/Test/TestMerchantReturnPage", "Currency": "PKR",
"AuthToken":
"gntytwdOkD52DeD4gJhEjNAUsmUY8DXclTuzBKtPrLjeYluWu987D0tC//EYDBV61wuatK2yCTHi1es1W6WL
G4RnzwcoTM920W49whyFqTlbexlk41ra2x8kSTg4jhZtn4m6VF5zSEHQnz+FfpYQzC+F04s+IWvjKZQdNfijqsk
ZJcQdV86C8RmSQgDMjMEMu",
"TransactionTypeId": "1", "TransactionReferenceNumber": "A100", "SMSOTAC": "",
"EmailOTAC": "", "SMSOTP": "12345678",
"HashKey": "wuSgntytwdOkD52DeD4gJhEjNAUsmUY8DXclTuzBKtPrLjeYluWu987D0tC//EYDBV61wuatK2yCT
Hi1es1W6WLG4Rn ",
"RequestHash":
"3QC1elErnLVID+K11qMWx3ddbh4dmVUoSfqlin3FQBOHFUUB7Dg4AsLeRman8zp2rxz4KVfQSqV8IbZQeq
2i9QI3+jinZnVZOMSnRoL6Za4Y3lBun1n6AZgh3tIPuwplOIDV4Wz5chitXmnf5FkPyyY5XQJ3268i/CUNT/7ztxE
NTuEQ8/9ftNiVEyMMD2GbYcqef1FuX0DBdLPLTRbSoQ6iDc+RdmmOofKoHpPNwWX3wuSgntytwdOkD52D
eD4gJhEjNAUsmUY8DXclTuzBKtPrLjeYluWu987D0tC//EYDBV61wuatK2yCTHi1es1W6WLG4RnzwcoTM920
W49whyFqTlbexlk41ra2x8kSTg4jhZtn4m6VF5zSEHQnz+FfpYQzC+F04s+IWvjKZQdNfijqskZJcQdV86C8RmS
QgDMjMEMuIXKo+wJOKR4k2GgYqH2JSTNoplZNMuKK4JOfM3Z1WYOIa/g/q0ydbOvm9hq0Ygx5QjHrJMNk
3218jvkVQ3FzvGiL2IuRgPnl41Ub6NrUHnPcjyRSeA5ve2ipD/k="
}
```

Sample Response of Process Transaction Request API:

<u>Success:</u>

```
{
        "merchant_id": "170",
        "merchant_name": "Test Merchant",
        "response_code": "00",
        "order_id": "A100",
        "order_datetime”: "03-07-2018 09:49:33",
        "paid_datetime": "03-07-2018 09:49:33",
        "transaction_status": "PAID",
        "transaction_amount": "10",
        "mobile_number": "03331234567",
        "payment_method": "1",
        "account_number": "48xxx58xx600",
        "description": "Success",
        "unique_tran_id": "1782631892367"
}
```

Failure:

```
{
        "success": "false",
        "AuthToken": "",
        "ReturnURL": "https://Sandbox.com/APITesting",
        "ErrorMessage": "Invalid Request"
}
```

Key Points:

- As, APG REST API's Request URL is a publicly accessible URL, you should first try to access the URL by simply copy pasting the URL into your browser and press enter. If you do get a response it means that your network does support publicly accessible Request URL, otherwise you will have to check your network security protocols first.

Integration for Credit/Debit Card Payment Method:

Since, credit/debit card has the customer sensitive financial details, merchants cannot process credit/debit card payments on their own platform. To cater this, APG offers a secure checkout page to the merchants on which their customers can provide their card details and Pay securely

In order to integrate and redirect customers to APG secure checkout page, following steps need to be followed.

STEP 1:

The merchant needs to POST below parameters in a form to APG on the following URL

Staging Environment:

URL: https://sandbox.bankalfalah.com/HS/HS/HS

Production Environment

URL: https://payments.bankalfalah.com/HS/HS/HS

Request Parameters:

Parameter	O/M	Explanation	Possible Values
HS_RequestHash	M	Generated Encrypted Hash of the request. See Encryption method in the Sample code attached	String
HS_IsRedirectionRequest	M	Handshake mode, This defines if a merchant wants to first redirect customers on a page where merchants will get authentication token (0) Or Merchants wants to handle the authentication token on the same page (1)	0,1
HS_ChannelId	M	Channel id of the integration channel. Page redirection = 1001	1001
HS_ReturnURL	M	Return URL of the merchant on which authentication token will be sent. This should be a public URL	String
HS_MerchantId	M	Unique Merchant id issued by APG to each merchant	String
HS_StoreId	M	Unique Store id issued by APG to each merchant	String
HS_MerchantHash	M	Unique Merchant hash generated by system for each merchant credentials	String
HS_MerchantUsername	M	Unique Merchant Username generated by system for each merchant credentials	String
HS_MerchantPassword	M	Unique Merchant Password generated by system for each merchant credentials	String
HS_TransactionReferenceNumber	M	Unique Order reference number generate by merchant for each order/transaction	String

<u>Success response</u>

```
{
        "success": "true",
            "AuthToken":
"uYtmxwFtQxUpp2uL6x4Echgk3WpfpzZcZi/N9k9P58Uv83WqpvuvNMjglFCC15LIDP2UTwJ3iBl
bwvMIMca 20BzagNY/92TtQIO9Zs1AvAWQCDb3U1kNUnUpLzW/EFVHB1SUmxDKnQk=",

            "ReturnURL": "https://google.com",

            "ErrorMessage": ""

}
```

Failure Response

```
{
   "success": "false",
   "AuthToken": null,
   "ReturnURL": "https://Sandbox.com/PageRedirectionTesting",
   "ErrorMessage": "Invalid Request"
}
```

STEP 2:
After completing the form in Step 1 the customer will be pressing the Proceed Button and lands on the merchant URL given in HS_ReturnURL variable in the first step. This will be a confirmation screen on merchant’s website to perform a handshake between APG and merchant’s website.

Now, APG sends back a parameter named auth_token to the HS_ReturnURL which is sent as a GET parameter. Now the merchant needs to post back below parameters again to the following URL:

Staging URL:

https://sandbox.bankalfalah.com/SSO/SSO/SSO

Production URL:

https://payments.bankalfalah.com/SSO/SSO/SSO

Request Parameters.

Parameter	O/M	Name	Explanation	Possible Values
AuthToken	M	AuthToken	Auth Token generated by system in response of successful Handshake	String
RequestHash	M	RequestHash	Generated Encrypted Hash of the request	String
ChannelId	M	ChannelId	Channel id of the integration channel. Page redirection = 1001,API=1002	String
Currency	O	Currency	Currency of the transaction	String
ReturnURL	M	ReturnURL	Return URL of the merchant on which customer will be redirected after payment success/failure	String
MerchantId	M	MerchantId	Unique Merchant id issued by APG to each merchant	String
StoreId	M	StoreId	Unique Store id issued by APG to each merchant	String
MerchantHash	M	MerchantHash	Unique Merchant hash generated by system for each merchant credentials	String
MerchantUsername	M	MerchantUsername	Unique Merchant Username generated by system for each merchant credentials	String
MerchantPassword	M	MerchantPassword	Unique Merchant Password	String

			generated by system for each merchant credentials	
TransactionTypeId	O	TransactionTypeId	Transaction type id through which transaction will be performed. Debit/Credit Card = 3	String 3
TransactionReferenceNumber	M	TransactionReferenceNumber	Unique Order reference number generate by merchant for each order/transaction	String
TransactionAmount	M	TransactionAmount	Transaction amount of the order placed by the customer through the merchant	String

Response:

Successful Response would result in redirection to the payment page. In case of failure, user will be redirected to an invalid request page.

Post method Sample Code (Encryption Mechanism Embedded):

https://drive.google.com/open?id=141dRIb6qvSxwROQicLCvOrvEVThDERyq

Sample Postman Collection Of All Discussed API's:

https://www.getpostman.com/collections/63197f313d2975a07446

Instant Payment Notification (IPN):

APG provides the Transaction Status to merchant platform in real time so that merchants can update their Database as soon as the customer has PAID for their order or the transaction has Failed due to any reason.

In order to do so, merchants have to follow the following steps:

Inquire Transaction Response from APG

- Merchants will have to inquire the order financial status from APG

- Once the final Success/Failure page of APG checkout is shown to the customer, customer is then redirected back to merchant Return URL/website (this URL is defined by the merchant while doing integrations)

- Once the customer is redirected, APG will return the Order ID on merchants Return URL with alias ‘O’ e.g www.google.com/TS=P/RC=00/RD=/O=A10 . Now merchant has the order ID against which a transaction is performed.

- Now merchants can inquire a transaction status by using the IPN URL i.e.

  https://sandbox.bankalfalah.com/HS/api/IPN/OrderStatus/123/000456/A10 – for Sandbox
  https://payments.bankalfalah.com/HS/api/IPN/OrderStatus/123/000456/A10 – for Production

  where 123= Merchant ID ; 000456= Store ID ; A10 = Order ID

- Merchant ID and Store ID in the IPN URL are merchants own ID’s and will never change and all a merchant has to do is append the Order ID as returned in the return URL (as explained in point 3) and initiate a GET call
- The above steps will return the following values to the merchant

```
ResponseCode = "00",
Description = "Success",
MerchantId = "123",
MerchantName = "Test merchant",
StoreId = "000456",
StoreName = "Test Store",
TransactionTypeId = "1",
TransactionReferenceNumber = "A10",
OrderDateTime = "09-10-2019 12:55:39 AM",
TransactionId = "1263781929",
TransactionDateTime = "09-10-2019 12:55:57 AM",
AccountNumber = "930003331234567",
TransactionAmount = "10",
MobileNumber = "03331234567",
TransactionStatus = "Paid"
```

The yellow highlighted parameter is the final status of the transaction and merchants can complete their order on the basis of this parameter value. i.e. if value is PAID order complete else order failed.

Or you can follow the below steps and configure your own IPN URL

Step1: Create your Listener URL

Listener URL Example:

www.sample.com/listener

Step 2: Configure this URL in your merchant portal.

Merchant portal > Login>GoLive>Access Sandbox>Credentials Generator>Listener URL

IPN Call from APG:

A POST call will be made from APG to the provided listener URL with the “url” parameter as follows:

www.sample.com/listener?url=https://sandbox.bankalfalah.com/HS/api/IPN/OrderStatus/123/000456/ A10

How To Get Response from IPN Call:

Merchant needs to initiate a GET call to the URL provided in “url” parameter. The order status will be
returned in JSON response as follows:

IPN Order Status response:

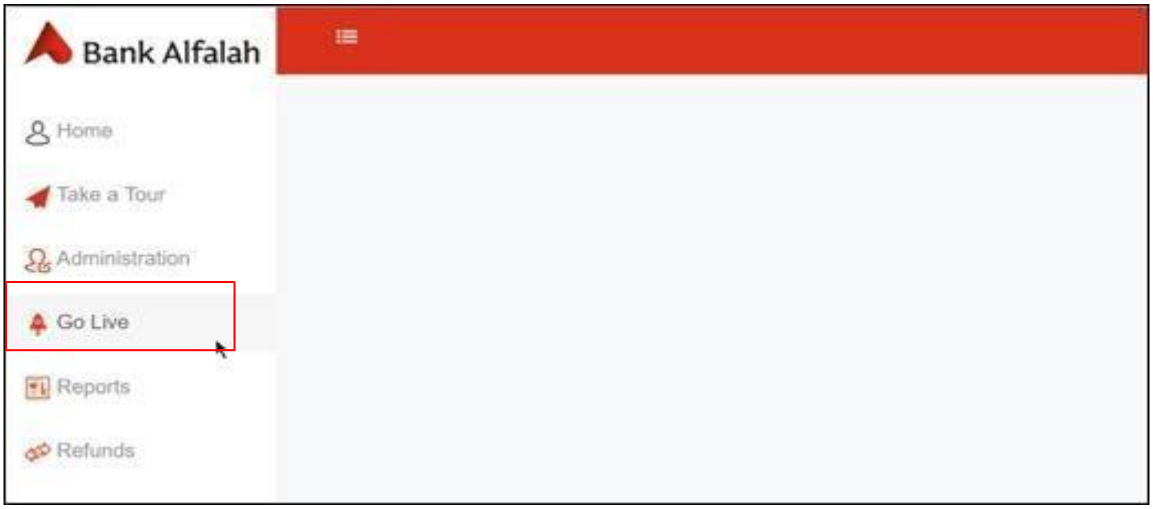
```
{
        ResponseCode =
        "00", Description =
        "Success",
        MerchantId = "123",
        MerchantName = "Test
        merchant", StoreId = "000456",
        StoreName = "Test Store",
        TransactionTypeId = "1",
        TransactionReferenceNumber =
        "A10",
        OrderDateTime = "09-10-2019 12:55:39 AM",
        TransactionId = "1263781929",
        TransactionDateTime = "09-10-2019 12:55:57 AM",
        AccountNumber = "930003331234567",
        TransactionAmount = "10",
        MobileNumber =
        "03331234567",
        TransactionStatus = "Paid"
}
```

Please note that once you configure your IPN URL. You will have to inform your business owner with your created IPN details as it has to be Whitelisted on Bank Alfalah Network, before you can start receiving transactions status

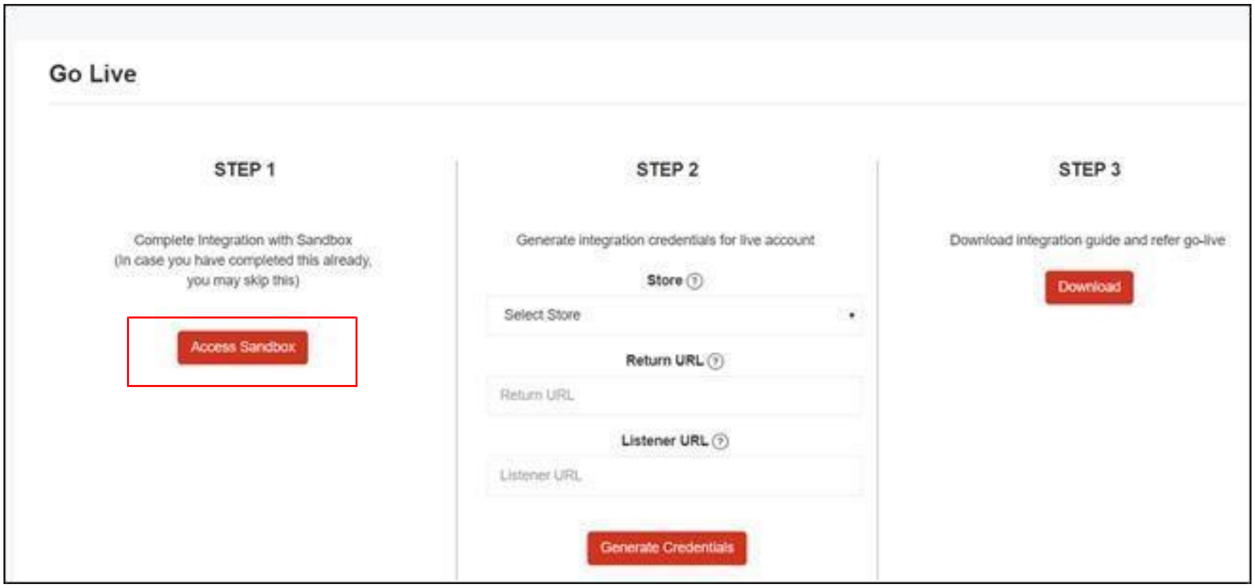
How to Go Live:

Once all your sandbox testing is completed and you want to move live. Kindly follow the below mentioned steps:

1. Login to your merchant portal
2. Click on “Go Live” button on the left side menu



3. Complete Step 2 and click on Generate Credentials



Go Live

STEP 1

Complete Integration with Sandbox
(In case you have completed this already, you may skip this)

STEP 2

Generate integration credentials for live account

Store: Select Store

Return URL: Return URL

Listener URL: Listener URL

STEP 3

Download integration guide and refer go-live

This will provide you with your Merchant Username, Merchant Password and Merchant Hash for production environment.

4. Once step 3 is completed, you will have to inform your concerned Business Owner from APG and they will provide you with your production Encryption Keys.