# **FD Connect**

Merchant Integration Guide Version 1.4

# Contents

1	OVERVIEW	3
2	TECHNICAL PREREQUISITES	3
3	TRANSACTION FLOW	4
4	STEPS FOR INTEGRATION	5
	4.1 JAVA integration	5
	4.2 .Net integration	6
	4.3 PHP integration	9
	4.4 Integration Types	9
5	TRANSACTION REQUEST API	11
	5.1 Request Parameters for Transaction Request	11
	5.2 Response Parameters for Transaction Request	14
6	MERCHANT RESPONSE HANDLING	21
	6.1 Request Parameters for Decrypt message	21
	6.2 Response Parameters for Decrypt message	22
7	TRANSACTION INQUIRY API	25
	7.1 Request Parameters for Transaction Inquiry	26
	7.2 Response Parameters for Transaction Inquiry	27
8	TRANSACTION REFUND API	30
	8.1 Request Parameters for Refund Transaction	30
	8.2 Response Parameters for Refund Transaction	31
9	STANDING INSTRUCTION TRANSACTION API	34
	9.1 Request Parameters for Standing Instruction Transaction	35
	9.2 Response Parameters for Standing Instruction Transaction	35

#### 1 OVERVIEW

This document describes the steps for technical integration process between merchant website / application and FD Connect.

Through FD Connect, your customers can initiate an online payment through any of the following available payment modes:

- Credit cards
- Debit cards
- ➤ EMI
- Standing instruction on cards
- Saved Cards
- Debit Card +ATM PIN/OTP
- Net banking
- ▶ UPI
- Cash Cards/Wallets
- Bharat QR

FD Connect also offers you a business UI (https://oceanui.firstdata.com) where you have access to all your prior transaction/payment details, settlement details, analytics, etc.

You can also use this UI to create invoices singly or in bulk, set reminders, recurring billing, and many more features.

### 2 TECHNICAL PREREQUISITES

### Java integration

- 1. Merchants web application should be in Java environment
- 2. JDK version should be 1.8 or more
- 3. Utilization of rest web service using FDConnect.jar, this will be shared by FISERV
- 4. MID(Merchant Id) and Page Id(Functional & Theme configuration), this will be shared by FISERV
- 5. Key and IV pair for encryption and decryption, this will be shared by FISERV
- 6. Error property file to be shared by FISERV

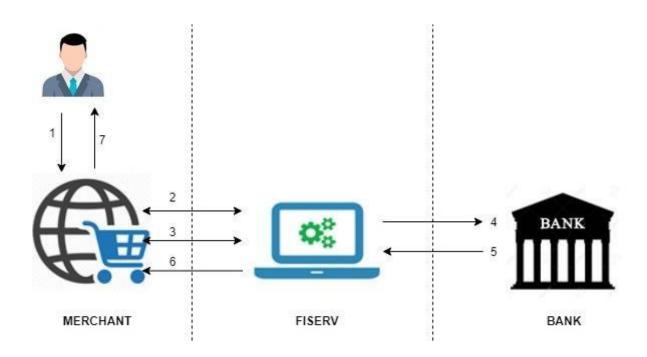
### .Net integration

- 1. MID(Merchant Id) and Page Id(Functional & Theme configuration), this will be shared by FISERV
- 2. Visual Studio version should be 2012 or above
- 3. Key and IV pair for encryption and decryption, this will be shared by FISERV
- 4. PaymentGatewayDLL.dll, this will be shared by FISERV
- 5. Error property file to be shared by FISERV

#### **PHP** integration

- 1. MID(Merchant Id) and Page Id(Functional & Theme configuration), this will be shared by FISERV
- 2. PHP version should be 7.2.14 or above
- 3. Key and IV pair for encryption and decryption, this will be shared by FISERV
- 4. Merchant should consume restful web services
- 5. Error property file to be shared by FISERV

### 3 TRANSACTION FLOW



Step 1: Customer selects items from Merchant web site and click on Check out button.

Step 2: Merchant initiate's Transaction Request API (Refer 5) call and receives session token.

**Step 3:** Merchant do the browser redirection to

https://www.test.fdmerchantservices.com/Pay/?sessionToken=RDDI27SEPEHZFDR3&configId=PageId2019080820099 then Customer redirects to hosted page. Customer selects payment mode and enter required details. Customer redirected to Bank /OTP page to authenticate transaction.

Step 4 & 5: Bank will verify customer transaction details and approve/reject the transaction.

**Step 6:** Transaction status is given to merchant on Result URL in encrypted form. Merchant uses decrypt merchant response API (Refer 8) to decrypt the message.

Based on transaction response the following scenarios are possible.

- 1. If transaction is successful then response given to merchant.
- **2.** If transaction is failed and merchant is not enabled for retry payment, then failure response is returned to merchant.
- **3.** If transaction is failed and merchant is enabled for retry payment, then will be redirected to hosted page to select payment and complete the transaction.

**Step 7:** Merchant displays transaction status to Customer.

### 4 STEPS FOR INTEGRATION

When you integrate with FD Connect, the customer will be re-directed from your merchant website to the FD Connect payment page. After completion of the transaction, FD Connect will direct the customer back to the merchant website.

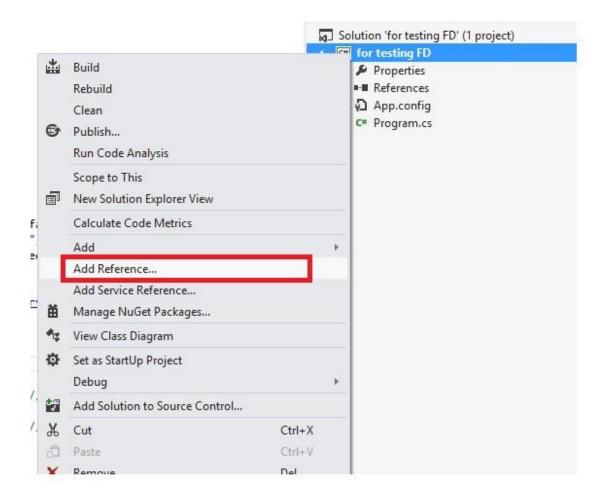
### 4.1 JAVA integration

- 1. Add FdConnect jar dependency in the project shared by FISERV.
- 2. In order to initiate a transaction request from your (merchant) check out page, create FDConnectSaleRequest Object and set mandatory fields using constructor or setter methods to set data.
- 3. Call FDConnectUtils.saleTxn(fdConectRequest) and this method returns FDConnectSaleResponse object(It contains SessionTokenId or ErrorCode & ErrorMessage)
- If session token received successfully then do the browser redirection to https://www.test.fdmerchantservices.com/Pay/?sessionToken=RDDI27SEPEHZFDR3&configI d=PageId2019080820099

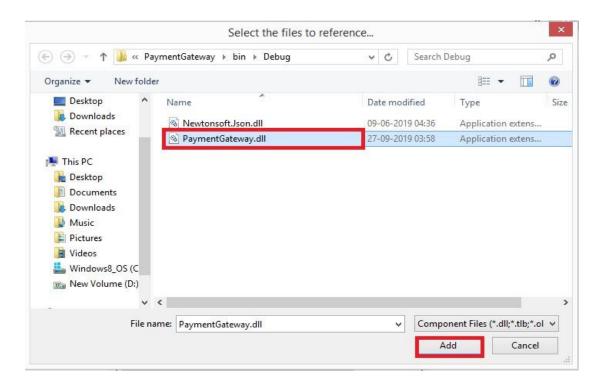
   Session token is received in Step no 3 and Config Id shared by FISERV
- 5. Browser redirection will happen to FD Connect payment page

### 4.2 .Net integration

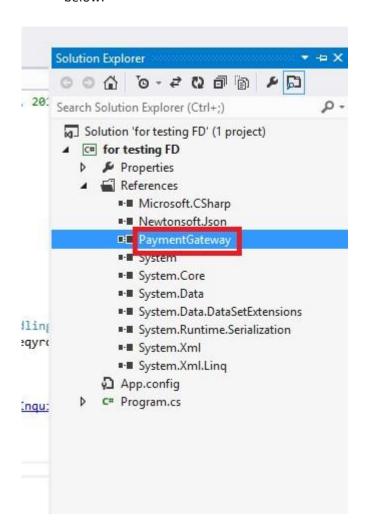
- 1. Open Visual Studio then select "File" -> "New" -> "Project." then select "Visual C#" -> " ASP.NET Web Forms Application ".
- 2. Add a reference for the dll file, "PaymentGatewayDLL.dll", which was created earlier. Rightclick on the project and then click on "Add reference" as shown below.



3. Select the DLL file and add it to the project as shown below



4. After adding the file, "Payment Gateway" namespace will get added (in references) as shown below:



5. Add the namespace ("using Payment Gateway;") as shown below

```
∃using Newtonsoft.Json.Linq;
 using Newtonsoft.Json;
 using System;
 using System.Collections.Generic;
 using System.Linq;
 using System.Text;
 using System. Threading. Tasks;
 using System.Net;
 using System. TO:
 using PaymentGateway;
   sing System Security reryptography;
 using System.Security.Cryptography.X509Certificates;
 using System.Text.RegularExpressions;

    □ namespace for testing FD

 {
     class Program
     {
         static void Main(string[] args)
Ė
              FristPayRequest FRObj = new FristPayRequest();
              FristPayResponse frobject = new FristPayResponse();
              FirstPayInquiryRequest frpayeqyrqt = new FirstPayInquiryRequest();
              RootObject frpayeqyresponse = new RootObject();
              FirstPayRefundRequest frpayrefreq = new FirstPayRefundRequest();
              FirstPayRefundResponse frpayrefres = new FirstPayRefundResponse();
```

- 6. In order to initiate a transaction request from your (merchant) check out page, create FDConnectSaleRequest Object and set mandatory fields using constructor or setter methods to set data.
- 7. Call FDConnectUtils.saleTxn(fdConectRequest) and this method returns FDConnectSaleResponse object(It contains SessionTokenId or ErrorCode & ErrorMessage)
- If session token received successfully then do the browser redirection to
   https://www.test.fdmerchantservices.com/Pay/?sessionToken=RDDI27SEPEHZFDR3&configId=PageId2019080820099

   Session token is received in Step no 7 and Config Id shared by FISERV
- 9. Browser redirection will happen to FD Connect payment page

### 4.3 PHP integration

- 1. Add php files in the project shared by FISERV.
- 2. In order to initiate a transaction request from your (merchant) check out page, create firstPayForm Object and set mandatory fields using constructor or setter methods to set data.
- 3. Call sale method of firstPayForm class and this method returns firstPayResponse object(It contains SessionTokenId or ErrorCode & ErrorMessage)
- 4. If session token received successfully then do the browser redirection to <a href="https://www.test.fdmerchantservices.com/Pay/?sessionToken=RDDI27SEPEHZFDR3&configId=PageId2019080820099">https://www.test.fdmerchantservices.com/Pay/?sessionToken=RDDI27SEPEHZFDR3&configId=PageId2019080820099</a> . Session token is received in Step no 3 and Config Id shared by FISERV
- 5. Browser redirection will happen to FD Connect payment page

### 4.4 Integration Types

FDConnect is a single integration which covers the following scenarios

### 1. Complete payment experience at FDConnect

In this integration Payment mode selection, Card details capturing and bank selection is done using FDConnect payment page only.

### 2. Only Payment mode selection at Merchant end

In this integration, Payment mode selection is at merchant end and Card details capturing and bank selection is done using FDConnect payment page. The following parameters need to set in GetToken API call.

- a. integrationType = MERCHANT\_PAYMENT\_MODE\_INTEGRATION
- b. paymentMethodType = CREDITCARD

### 3. Bank/Wallet selection at Merchant end but Card capture at FDConnect

In this integration, Payment mode selection is at merchant end and Bank/Wallet selection is at merchant side and Card details capturing in FDConnect payment page.

The following parameters need to set in GetToken API call.

- a. integrationType=MERCHANT\_PAYMENT\_MODE\_BANKORWALLET\_INTEGRATION
- b. paymentMethodType = NETBANKING
- c. bankId, bankName/ walletId, walletName

### 4. Bank/Wallet and Card details capturing at Merchant end

In this integration, Payment mode selection, Bank/Wallet selection and Card details capturing at merchant side and FDConnect payment page is pass through only.

The following parameters need to set in GetToken API call.

- a. integrationType=MERCHANT\_PAYMENT\_MODE\_CARD\_INTEGRATION
- b. paymentMethodType=CREDITCARD
- c. cardNumber, expMonth, expYear, cVV and nameOnCard

### 5. Link Pay integration

In this integration, merchant generates tiny payment URL and share with the customers through SMS or Email. Once customer clicks on the tiny url, it will redirect to FDConnect hosted page.

The following parameters need to set in GetToken API call.

- a. integrationType = MERCHANT\_LINK\_PAYMENT\_MODE\_INTEGRATION
- b. configld = Page id provided by Fiserv
- c. sessionTokenExpireDate = link expiry date(dd-mm-yyyy)

#### Note:

#### **Merchant Transaction ID:**

- When you submit your transaction request to FD Connect, you need to submit a merchant transaction ID as part of the request. This ID can be used by you as a universal reference number for all transaction requests submitted by you.
- Merchant transaction ID acts as a "merchant reference number". We strongly recommend that you maintain uniqueness of this ID, to avoid confusion and conflicts while retrieving transaction details subsequently.

#### **FPTransaction ID:**

• When your customer clicks the "Pay" button on the FD Connect payment page, a unique transaction ID is assigned to the transaction.

## 5 TRANSACTION REQUEST API

Request Type	Transaction(GetToken) API Call(Server2Server call)
Description	This API sends transaction and customer details and receive session token. This sessiontoken used to do the transaction
Event/Trigger of Reque st Type	This is Server2Server call. Merchant need to initiate this call when customer clicks on Checkout page.
List of Parameters in Request & Response	Refer 5.1 and 5.2

## 5.1 Request Parameters for Transaction Request

URL: https://www.test.fdmerchantservices.com/FirstPayL2Services/getToken

Parameter Name	Description	Optional/ Mandatory	Sample value
merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	Mandatory	M123456789
key	First Pay would assign a unique key to the merchant. This will part of sharing the integration kit with the merchant	Mandatory	
iv	IV value is associated with the key shared with the merchant	Mandatory	
apiURL	API URL	Mandatory	
merchantTxnld	Merchant needs to pass this value for every transaction request and this can be referred for future interaction. It must be unique for every transaction.	Mandatory	TR1234
resultURL	This is the url in which transaction response has to be posted back with	Mandatory	http://firstdata/ result.jsp

	some set of parameters once the transaction is complete		
transactionType	This has to be "sale" in this case	Mandatory	sale
paymentMethodT ype	This value will be passed only if the payment selection is done at merchants end	Conditional	creditcard/debit card/netbanking /wallet/upi
amount	This is the payment amount	Mandatory	100.00
currencyCode	This is the 3 digit currency code as defined in the master DB	Mandatory	INR
hostedId	This will be passed by the merchant if the hosted IDs for the saved card are managed by the merchant for future transactions	Optional	KRS1220ESFFTR 849
customerId	This has to be the customer id if the merchant wants to pass	Optional	CA12345
firstName	First Name of customer	Optional	Abc
lastName	Last Name of customer	Optional	Xyz
middleName	Last Name of customer	Optional	Tbc
suffix	Suffix to be added along with the name	Optional	Mr/Mrs/Miss
emailld	Customer email address.	Optional	test@gmail.com
mobileNo	Customer phone number	Optional	1234567891
billStreet1	Customer billing address1	Optional	Aaaa
billStree2	Customer billing address2	Optional	Bbbb
billCity	Customer billing city	Optional	Mumbai
billState	Customer billing state	Optional	Maharashtra
billCountry	Customer billing country	Optional	India
billZipcode	Customer billing zip code	Optional	400001
shipStreet1	Customer billing address1	Optional	Aaaa
shipStree2	Customer billing address2	Optional	Bbbb
shipCity	Customer billing city	Optional	Mumbai
shipState	Customer billing state	Optional	Maharashtra
shipCountry	Customer billing country	Optional	India
shipZipcode	Customer billing zip code	Optional	400001
productData	This needs to be passed if single transaction is initiated for multiple products with different price. The sum of price of all the products will be	Optional	

	equal to the total transaction amount of the request		
airlineData	This field is for the merchants who captures airline booking transactions	Optional	
integrationType	This filed indicates integration type.	Conditional	Refer 51. Integration types
cardData	This field is required if merchant captures the card detail in its own page and pass as part of transaction request	Conditional	Card no, expiry month, expiry year,  CVV, Name on
			card
bankData	This field is required if merchant wants to capture the bank selection at its end	Conditional	bankId, bankName
walletData	This field is required if merchant wants to capture the wallet selection at its end	Conditional	walletId, walletName
upiData	This field is required if merchant wants to capture the UPI details (VPA) at its end	Conditional	abc@icick
configld	In case of Link pay this is mandatory. Page Id shared by Fiserv	Conditional	
sessionTokenExpir	In case of Link pay this is mandatory.	Conditional	
eDate	Link expiry date (dd-mm-yyyy)		
udf1		Optional	
udf2	User defined field	Optional	
udf3	User defined field	Optional	
udf4	User defined field	Optional	
udf5	User defined field	Optional	
udf6	User defined field	Optional	
udf7	User defined field	Optional	
udf8	User defined field	Optional	
udf9	User defined field	Optional	
udf10	User defined field	Optional	

# 5.2 Response Parameters for Transaction Request

Parameter Name	Description	Sample value
sessionToken	This is the unique id assigned to the transaction.	RDDI27SEPEHZFDR3
errorCode	Error code	400 BAD_REQUEST
errorMessage	Error description	FPDR4013:Merchant_ txn_id cannot be blank
udf1	User defined field	
udf2	User defined field	
udf3	User defined field	
udf4	User defined field	
udf5	User defined field	
udf6	User defined field	
udf7	User defined field	
udf8	User defined field	
udf9	User defined field	
udf10	User defined field	_

## Sample code:

Language JAVA	
---------------	--

```
public class SaleTest {
     public static void main(String[] args)
          //Using Constructor
          FDConnectSaleRequest request = new
FDConnectSaleRequest("470000000332252", "CNy+HimxmI4PmrJWrpLarBfbo6
jIY/CHcezg2VQ8u5o=","9T4hd3Nx0b0sMgYuyWLCTg==","https://www.test.f
dmerchantservices.com/FirstPayL2Services/gettoken","10","INR","TRA
123456", "sale", "https://merchantresposneurl");
           //Using Setter methods
           /*
FDConnectSaleRequest request = new FDConnectSaleRequest();
          request.setMerchantId("47000000332252");
          request.setKey("CNy+HimxmI4PmrJWrpLarBo6jIY/CH8u5o=");
          request.setIv("9T4hd3Nx0b0sMgYuyWLCTg==");
     request.setResultURL("http://localhost:8080");
     request.setAmount("10");
     request.setCurrencyCode("INR");
     request.setMerchantTxnId("TRA123456");
     request.setTransactionType("sale");
```

```
request.setCustomerId("CUST1234");
     request.setFirstName("D'costa-Flavin");
     request.setLastName("Last Name");
     request.setMiddleName("Middle Name");
     request.setSuffix("Mr");
     request.setMobileNo("9769970828");
     request.setEmailId("Prashanth@gmail.com");
           request.setBillStreet1("billStre'et1address");
     request.setBillStree2("billStree2");
     request.setBillState("billState");
     request.setBillCity("billCity");
     request.setBillState("billState");
     request.setBillZipcode("123456789");
           request.setShipStreet1("shipStreet1");
     request.setShipStree2("shipStree2");
     request.setShipCity("shipCity");
     request.setShipState("shipState");
     request.setShipZipcode("123456789");
           List<ProductDataBean> productList = new
ArrayList<>();
                     ProductDataBean bean = new
ProductDataBean();
                           bean.setProductId("Prod1");
     bean.setProductDescription("productDescription");
     bean.setQuantity("1");
                                      bean.setPrice("10");
           bean.setTxnAmount("10");
     bean.setShippingFee("0");
     bean.setDiscountPrice("0");
     productList.add(bean);
     request.setProductData(productList);
     request.setApiURL("https://www.test.fdmerchantservices.com/Fi
rstPayL2Services/gettoken");
                * /
                           FDConnectSaleResponse resp =
FDConnectUtils.saleTxn(request);
                                 System.out.println("resp
SessionTokenId:
"+resp.getSessionTokenId());
```

	System.out.println("resp
ErrorCode :	• • • • • • • • • • • • • • • • • • • •
"+resp.getErrorCode());	
100p.gccd110100dc(///	
	System.out.println("resp
ErrorMessage :	
"+resp.getErrorMessage());	
١	
}	

Language	Dot Net
}	

```
namespace sale testing FD
     class
FDConnectSale
        static void
Main(string[] args)
            FristPayRequest FRObj = new FristPayRequest();
                                      FristPayRequestSample
FRSample = new
FristPayRequestSample();
            FristPayResponse frobject = new FristPayResponse();
                                      FristPayRequestSample
request = new
FristPayRequestSample("47000000332252",
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezq2VQ8u5o=",
"9T4hd3Nx0b0sMgYuyWLCTg==",
"https://WWW.test.fdmerchantservices.com/FirstPayL2Services/getTo
k en", "1.5", "INR", "123456", "SALE", "Prashanth", "Kumar", "M",
"9769970899", "prashanth@gmail.com",
"/FirstPayL2Services/merchantSuccessPage.jsp");
                                 /*FRSample.amount = "1.5";
            FRSample.currencyCode = "INR";
            FRSample.clientTransactionId = "5104475445";
            FRSample.transactionType = "SALE";
            FRSample.fpURL =
"https://WWW.test.fdmerchantservices.com/FirstPayL2Services/getTo
k en";
            FRSample.merchantid = "47000000332260";
            FRSample.Key =
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=";
            FRSample.IV = "9T4hd3Nx0b0sMqYuyWLCTq==";
            FRSample.customerId = "ABC1234";
List<String> obj = new List<String>();
obj.Add("KRS1220ESFFTR8490");
            FRSample.hostedIds = obj;
            FRSample.firstName = "ABC";
            FRSample.lastName = "XYZ";
            FRSample.middleName = "DEF";
            FRSample.suffix = "ADX";
            FRSample.mobileNo = "1234567890";
            FRSample.emailId = "abc@gmail.com";
            FRSample.street1 = "Ghodbandar Rode";
                FRSample.stree2 = "Sai Nagar";
```

FRSample.city = "Mumbai";

```
FRSample.state = "Maharashtra";
            FRSample.country = "India";
            FRSample.zipcode = "400000";
            FRSample.street1 = "Ghodbandar Rode";
            FRSample.stree2 = "Sai nagar";
            FRSample.city = "Mumbai";
            FRSample.state = "Maharashtra";
            FRSample.country = "India";
            FRSample.zipcode = "400000";
                FRSample.paymentMethodType = "CREDITCARD";
            FRSample.integrationType =
"MERCHANT PAYMENT MODE INTEGRATION"; */
               string sResponsePayRequest = FRObj.sale(request);
frobject =
JsonConvert.DeserializeObject<FristPayResponse>(sResponsePayReques
t);
   }
```

```
Language
                          PHP
<?php
require once('firstPayFormRequest.php');
require once 'firstPayResponse.php';
if(isset($ POST['submit']) && !empty($ POST['clientid'])) {
    \frac{10000000}{1000000}, \frac{10000000}{1000000}, \frac{10000000}{1000000};
    $response = new \FristPayResponse();
    $bsObj = new \firstPayForm("47000000332260",
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=",
"9T4hd3Nx0b0sMqYuyWLCTq==",
"https://WWW.test.fdmerchantservices.com/FirstPayL2Services/gettok
en","1.5", "INR","9805", "SALE","ABC", "XYZ", "DEF", "1234567890",
"abc@gmail.com",
"http://localhost:81/FirstData/sampleTestResult.php");
    $response = json decode($bsObj->sale($bsObj));
         if(isset($response->response-
>sessionTokenId)){
        $sessiontokenID = $response->response->sessionTokenId;
```

### 6 MERCHANT RESPONSE HANDLING

After transaction is completed, FDConnect application post the final response on merchant **resultURL.** 

### The following fields available as part of response

- fpTxnId
- 2. encData Encrypted response message
- merchantId

After getting the above fields write the following code to decrypt the data and display the response to customer.

```
String fpTxnId = request.getParameter("fpTxnId");
String merchantId = request.getParameter("merchantId");
String encrypatData = request.getParameter("encData");
```

Request Type	Decrypt Response API Call(Server2Server call)
Description	This API used do the decrypt the transaction data.
Event/Trigger of Request Type	This is Server2Server call. Merchant can initiate this API to decrypt the transaction details
List of Parameters in Request & Response	Refer 6.1 & 6.2

### 6.1 Request Parameters for Decrypt message

**URL:** https://www.test.fdmerchantservices.com/FirstPayL2Services/decryptMerchantResponse

Parameter Name	Description	Optional/ Mandatory	Sample value
----------------	-------------	------------------------	--------------

merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	Mandatory	M123456789
apiURL	This is the url in which transaction request has to be posted and receives response	Mandatory	
fdcTxnId	This transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	Mandatory	
encrypatData	Transaction status in Encrypted form	Mandatory	

# 6.2 Response Parameters for Decrypt message

Parameter Name	Description	Sample valu e
merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	
resultUrl	This is the url in which transaction response is done	
fdcTxnId	This transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	
transactionType	Transaction Type like Sale	
baseCurrency	Base Currency	
transactionCurrency	Transaction Currency	
baseAmount	Base Amount	
transactionAmount	Transaction Amount	
balanceAmount	Amount available for withdraw	
paymentMethod	Payment Method	
convFee	Convenience fee applied on base amount	
taxFee		
transactionStatus	Transaction Status	
transactionStatusCode	Status code	

transactionStatusDescri ption	Transaction description	
transactionDateTime	Transaction Date and Time	
standingInstruction	True in case of Standing instruction	
StandingInstructionBea n	Standing instruction details	
MerchDccResposeBean	In case of DCC transaction this would present	
MerchEmiDetail	In Case of EMI transaction this would present	
udf1	User defined field	
udf2	User defined field	
udf3	User defined field	
udf4	User defined field	
udf5	User defined field	
udf6	User defined field	
udf7	User defined field	
udf8	User defined field	
udf9	User defined field	
udf10	User defined field	
errorCode	Error code	
errorMessage	Error description	

### Sample code:

Language	JAVA

```
public class DecryptTest {      public
static void main(String[] args) {
```

FDConnectDecryptRequest fdConnectDecryptRequest = new FDConnectDecryptRequest("47000000332252","dMaRD5LgrSQ2sfIM8eFjZAV 2lus/Kkr7FhKDuv7sNMqkQheLJaKjoKayt+rDBAeuf4Uhy901MTRIFCZcRCr7jAasA NBcbmLENRToIaz54HSgCMT1Y6F5HA52WSpm+qWFMiJhvvQg0dV9c4TjZHGd5H1k9Jq 6Ti4WZv8hti0Cl+k6uEP4VATCFEFEBnROoOHDW1CipglBkwQyDb5Zz+n04il3QFcuD kHsDvphj40wEZetGInQTD3JG+NUUIi/JjEenkARW5F6F6+xYkuLRBYW6XVxs/MLdsG zVjTlEuHh5Q2xGE4TQ6bVGCq3Z37ojNceUKX7pCJv47aAJC0xpUjuK0TZwC8h0x5/V MZXPZR3E4NP03sKOos8cJxam4GFi9Ize7ci+efcI20EnbS8f4OncFZ3u6z59pBKaGn ytTfaM+GzdcVxS1IapXpQwQO5pDhLF8N3pqRR6U3mU39ArsWyv9bh1c2TO7aAMRqnz xhvAqlIveVHGo3ZoAvtavVcVX/1vuIlJyetziblaQzCJU7isXW2iDDYlyXUZA2Qp2U pAilHTY2pqo/WQLR8q8wbjNViDt6cAh7BYHtHGiikZiQU3rHJS5UuHvNl1d3w8pd92 8oBjzvoAbqMCCDqJ9wCYzoUDLmSyZCV2v64vIrFGFrDuATVPi+SKG42BGNBM61kSMw QtUZhHJ1jDfAAamVMmVdnmVh2vUZQoSQ6hBhYWTxYvj3BR8jKFSfnuwIf+Cy65n+KZ FO6xSckOEsznWalcXD0z51Z8U3c5AHUpHEwhX62qFvDpxHEhXphX2jhiFdHD3kX4oL WNtVtL2VXGqtwS5U7MqraDx8RIHoUlfwdUA+HXw==","2019102520378696","htt ps://www.test.fdmerchantservices.com/FirstPayL2Services/decryptMer chantResponse");

FDConnectDecryptResponse resp =
FDConnectUtils.decryptMsg(fdConnectDecryptRequest);

```
System.out.println(resp.getErrorCode());

System.out.println(resp.getErrorMessage());

System.out.println(new
Gson().toJson(resp));

}
```

Language Dot Net

```
rr.fpTxnId =
        Result rr = new Result();
strfpTxnId;
                   rr.encData = strencData;
rr.merchantId = strmerchantId;
                                      dynamic data =
JsonConvert.SerializeObject(rr);
                           string sResponsePayRequest =
this.getresponse("https://WWW.test.fdmerchantservices.com/FirstPa
L2Services/decryptMerchantResponse", strmerchantId, data);
        dynamic data1 =
JsonConvert.DeserializeObject(sResponsePayRequest);
        if
              (data1["transactionStatus"]
"SUCCESS")
        {
                     txtPaymentId.Text = strfpTxnId;
txtTrackId.Text = strmerchantId;
txtResult.Text = data1["transactionStatus"];
txtamount.Text = data1["baseAmount"];
pnlpaymentsuccessinfo.Visible = true;
```

Language PHP

```
<?php
$fpTxnId=$_REQUEST['fpTxnId'];
$encData=$_REQUEST['encData'];
$merchantId=$_REQUEST['merchantId'];

include("firstPayResult.php");

$bsObj = new \firstPayResult($merchantId, $encData, $fpTxnId);
$getJson = (json_encode($bsObj, JSON_UNESCAPED_SLASHES));
var_dump($getJson); exit;
$response = $bsObj-
>curlCall($merchantId,$getJson,'https://WWW.test.fdmerchantservicees.com/FirstPayL2Services/decryptMerchantResponse');

$jsonObject = json_decode($response,true); ?>
```

### 7 TRANSACTION INQUIRY API

**Request Type** 

Transaction Inquiry API Call(Server2Server call)

Description	This API used for transaction inquiry. In response it will return SALE and REFUND transactions based on FDConnect transaction Id	
Event/Trigger Request Type	This is Server2Server call. Merchant can initiate whenever he required	
List of Parameters in Request & Response	Refer 7.1 & 7.2	

# 7.1 Request Parameters for Transaction Inquiry

### URL:

https://www.test.fdmerchantservices.com/FirstPayL2Services/getTxnInquiryDetail

Parameter Name	Description	Optional/ Mandatory	Sample value
merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	Mandatory	M123456 789
key	First Pay would assign a unique key to the merchant. This will part of sharing the integration kit with the merchant	Mandatory	FD0001
iv	IV value is associated with the key shared with the merchant	Mandatory	Srffhjtcv1 345nk
merchantTxnId	Merchant needs to pass this value for every transaction request and this can be referred for future interaction. It must be unique for every transaction.	Mandatory	TR1234
apiURL	This is the url in which transaction request has to be posted and receives response	Mandatory	https:// www.test .fdmerch antservi ces.com/ FirstPay L2Servic es/getTx nInquiry Detail
fdcTxId	This transaction id is returned by	Optional	

fdcTxId	This transaction id is returned by	Optional	
	FDConnect to merchant after		
	successful/failure of the transaction.		
	The same need to pass to get the		
	transaction status		

# 7.2 Response Parameters for Transaction Inquiry

Parameter Name	Description	Sample value
fdcRefundTxnId	This transaction id is returned by FDConnect to merchant after successful/failure refund transaction. The same need to pass to get the transaction status	M123456789
fdcTxnld	This transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	
merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	
merchantTxnId	Merchant needs to pass this value for every transaction request and this can be referred for future interaction. It must be unique for every transaction.	
baseAmount	Transaction amount	
txnStatus	Status of the transaction	
txnDescription	Transaction description	
txnType	Transaction type like sale/refund	
channel	Transaction channel	
currency	Transaction Currency Code	
txnDate	Transaction date	
balanceAmount	Available amount for refund	_

### Sample code:

Language	JAVA

```
public class InquiryTest {
                                public
static void main(String[] args) {
           //Using Constructor
           FDConnectInquiryRequest firstPayInquiryRequest = new
FDConnectInquiryRequest("47000000332252","CNy+HimxmI4PmrJWrpLarBf
bo6jIY/CHcezg2VQ8u5o=","9T4hd3Nx0b0sMgYuyWLCTg==","https://www.tes
t.fdmerchantservices.com/FirstPayL2Services/getTxnInquiryDetail","
123456", "2019092518359898");
           //Using Setter methods
           FDConnectInquiryRequest firstPayInquiryRequest1 = new
FDConnectInquiryRequest();
     firstPayInquiryRequest1.setMerchantId("47000000332252");
     firstPayInquiryRequest1.setMerchantTxnId("123456");
     firstPayInquiryRequest1.setFdcTxnId("2019092518359898");
     firstPayInquiryRequest1.setApiURL("https://www.test.fdmercha
n tservices.com/FirstPayL2Services/getTxnInquiryDetail");
     firstPayInquiryRequest1.setKey("CNy+HimxmI4PmrJWrpLarBfbo6jIY
/CHcezg2VQ8u5o=");
     firstPayInquiryRequest1.setIv("9T4hd3Nx0b0sMgYuyWLCTg==");
           FDConnectInquiryResponse resp =
FDConnectUtils.inquiryTxn(firstPayInquiryRequest);
           System.out.println(resp.getErrorCode());
           System.out.println(resp.getErrorMessage());
           System.out.println(new Gson().toJson(resp));}
```

```
namespace for_testing_FD
{     class
Program
     {
         static void Main(string[] args)
          {
                FirstPayInquiryRequest frpayeqyrqt = new
FirstPayInquiryRequest();
                FirstPayInquiryRequestSample FRINQsample = new
FirstPayInquiryRequestSample();
                FirstPayInquiryRequestSample frpayeqyresponse = new
FirstPayInquiryResponse();
```

```
/*FRINQsample.merchantTxnId = "123456";
            FRINQsample.fpTransactionId = "2019092618971403";
            FRINQsample.fpURL =
"https://www.test.fdmerchantservices.com/FirstPayL2Services/getTxn
InquiryDetail";
            FRINQsample.merchantId = "47000000332252";
            FRINQsample.key =
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=";
            FRINQsample.iv = "9T4hd3Nx0b0sMgYuyWLCTg==";
            FRINQsample.logpath = "C:\\PaymentGateway Log\\";*/
            FirstPayInquiryRequestSample FRINQsampleCons = new
FirstPayInquiryRequestSample("123456", "2019092618971403",
"https://www.test.fdmerchantservices.com/FirstPayL2Services/qetTxn
InquiryDetail", "47000000332252",
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezq2VQ8u5o=",
"9T4hd3Nx0b0sMgYuyWLCTg==", "C:\\PaymentGateway Log\\");
             string sResponsePayRequest2 =
frpayeqyrqt.getresponsePayInquiryandRefundRequest(FRINQsampleCons)
            frpayeqyresponse =
JsonConvert.DeserializeObject<FirstPayInquiryResponse>(sResponsePa
yRequest2);
   }
```

```
c?php
namespace PaymentGateway;
require_once 'FirstPayRefundRequest.php'; require_once
'FirstPayInquiryRequest.php'; require_once
'FirstPayRefundResponse.php'; require_once
'FirstPayInquiryResponse.php';
require_once('firstPayFormRequest.php');
$firstPayRefundRequest = new FirstPayRefundRequest();
$firstPayInquiryRequest = new FirstPayInquiryRequest();
$bsObj = new \firstPayForm();
$firstPayInquiryRequest->merchantTxnId = "123456";
```

```
$firstPayInquiryRequest->fpTransactionId =
"2019093014080475";//"2019093022517507";

$firstPayInquiryRequest->fpURL
='https://www.test.fdmerchantservices.com/FirstPayL2Services/getTx
nInquiryDetail';
$firstPayInquiryRequest->key =
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=";
$firstPayInquiryRequest->iv = "9T4hd3Nx0b0sMgYuyWLCTg==";
$firstPayInquiryRequest->merchantId = "470000000332252";
$responsel = new \RootObject();
$responsel = json_decode($bsObj-
>refund_inquiry($firstPayInquiryRequest));
var_dump($responsel);
```

### 8 TRANSACTION REFUND API

Request Type	Refund API Call(Server2Server call)	
Description	This API used do the refund transaction.	
Event/Trigger of Request Type	This is Server2Server call. Merchant can initiate whenever he required	
List of Parameters in Request & Response	Refer 8.1 & 8.2	

### 8.1 Request Parameters for Refund Transaction

URL: https://www.test.fdmerchantservices.com/FirstPayL2Services/refundTxnDetail

Parameter Name	Description	Optional/ Mandatory	Sample value
	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	Mandatory	M123456789
	First Pay would assign a unique key to the merchant. This will part of sharing the integration kit with the merchant	Mandatory	
	IV value is associated with the key shared with the merchant	Mandatory	

merchantTxnId	Merchant needs to pass this value for every transaction request and this can be referred for future interaction.	Optional	
	It must be unique for every transaction.		
apiURL	This is the url in which transaction request has to be posted and receives response	Mandatory	https://www.t est.fdmerchan tservices.com /FirstPayL2Se rvices/refund TxnDetail
fdcTxnId	This transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	Mandatory	
refundAmount	Amount to refund	Mandatory	

# 8.2 Response Parameters for Refund Transaction

Parameter Name	Description	Sample value
merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	M123456789
fdcTxId	This transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	
fdcRefundTxnId	This refund transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	
txnStatus	Transaction Status description	
txnStatusCode	Transaction Status code	399
amount	Refund amount	

txDate	Transaction date	
currencyCode	Transaction currency code	
Balance	Balance amount after refund	
cardBrand	Card brand	
bankCode	Bank Code	
errorCode	Error code	
errorMessage	Error description	

### Sample code:

```
JAVA
Language
public class RefundTest {
     public static void main(String[] args) {
           FDConnectRefundRequest firstPayRefundRequest = new
FDConnectRefundRequest("47000000332252", "CNy+HimxmI4PmrJWrpLarBfb
o6jIY/CHcezg2VQ8u5o=","9T4hd3Nx0b0sMgYuyWLCTg==","https://www.test
.fdmerchantservices.com/FirstPayL2Services/refundTxnDetail","12345
6","2019092518359898","1");
           FDConnectRefundResponse resp =
FDConnectUtils.refundTxn(firstPayRefundRequest);
           System.out.println(resp.getAmount());
           System.out.println(resp.getFdcRefundTxnId());
           System.out.println(resp.getFdcTxnId());
           System.out.println(resp.getPaymentType());
           System.out.println(resp.getTxnStatus());
           System.out.println(new Gson().toJson(resp));
     }
```

Language	Dot Net

```
namespace for_testing_FD
{     class
Program
     {
        static void Main(string[] args)
          {
```

```
FirstPayRefundRequest frpayrefreq = new
FirstPayRefundRequest();
            FirstPayRefundResponse frpayrefres = new
FirstPayRefundResponse();
                /*FRRefndsample.merchantTxnId = "123456";
            FRRefndsample.fpTransactionId =
"2019092618971403";//2019092618971403
            FRRefndsample.refundAmount = "1.00";
            FRRefndsample.fpURL =
"https://www.test.fdmerchantservices.com/FirstPayL2Services/refund
TxnDetail";
            FRRefndsample.merchantId = "47000000332252";
            FRRefndsample.key =
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=";
            FRRefndsample.iv = "9T4hd3Nx0b0sMqYuyWLCTg==";
            FRRefndsample.logpath = "C:\\PaymentGateway Log\\";*/
               FirstPayRefundRequestSample FRRefndsampleCons =
new FirstPayRefundRequestSample("123456", "2019092618971403",
"1.00",
"https://www.test.fdmerchantservices.com/FirstPayL2Services/refund
TxnDetail", "47000000332252",
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=",
"9T4hd3Nx0b0sMgYuyWLCTg==", "C:\\PaymentGateway Log\\");
             string sResponsePayRequest3 =
frpayrefreq.getresponsePayInquiryandRefundRequest (FRRefndsampleCon
s);
            frpayrefres =
JsonConvert.DeserializeObject<FirstPayRefundResponse>(sResponsePay
Request3);
    }
```

```
<?php
namespace PaymentGateway;
require once 'FirstPayRefundRequest.php'; require once
'FirstPayInquiryRequest.php'; require once
'FirstPayRefundResponse.php'; require_once
'FirstPayInquiryResponse.php';
require once('firstPayFormRequest.php');
$firstPayRefundRequest = new FirstPayRefundRequest();
$firstPayInquiryRequest = new FirstPayInquiryRequest();
$bsObj = new \firstPayForm();
$firstPayRefundRequest->merchantTxnId = "123456";
$firstPayRefundRequest->fpTransactionId = "2019093014080475";
$firstPayRefundRequest->refundAmount = "1";
$firstPayRefundRequest->merchantId = "470000000332252";
$firstPayRefundRequest->fpURL
='https://www.test.fdmerchantservices.com/FirstPayL2Services/refun
dTxnDetail';
$firstPayRefundRequest->key =
"CNy+HimxmI4PmrJWrpLarBfbo6jIY/CHcezg2VQ8u5o=";
$firstPayRefundRequest->iv = "9T4hd3Nx0b0sMqYuyWLCTq==";
$response = new FirstPayRefundResponse();
$response = json decode($bsObj-
>refund inquiry($firstPayRefundRequest));
var dump($response);
```

### 9 STANDING INSTRUCTION TRANSACTION API

Request Type	Standing Instruction subsequent API Call(Server2Server call)
Description	This API used do the SI subsequent transactions.
Event/Trigger of Request Type	This is Server2Server call. Merchant can initiate whenever he required
List of Parameters in Request & Response	Refer 9.1 & 9.2

## 9.1 Request Parameters for Standing Instruction Transaction

URL: https://www.test.fdmerchantservices.com/FirstPayL2Services/refundTxnDetail

Parameter Name	Description	Optional/ Mandatory	Sample value
merchantId	This is the unique id assigned to the merchant at the time of on boarding	Mandatory	M123456789
	in FD Connect system		

key	First Pay would assign a unique key to the merchant. This will part of sharing the integration kit with the merchant	Mandatory	
iv	IV value is associated with the key shared with the merchant	Mandatory	
merchantTxnId	Merchant needs to pass this value for every transaction request and this can be referred for future interaction. It must be unique for every transaction.	Mandatory	
apiURL	This is the url in which transaction request has to be posted and receives response	Mandatory	https://www.t est.fdmerchan tservices.com /FirstPayL2Se rvices/refund TxnDetail
fdcTxnld	This transaction id is returned by FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status	Mandatory	
txnAmount	Transaction Amount	Mandatory	
hostedId			
currencyCode			
customerId			

# 9.2 Response Parameters for Standing Instruction Transaction

Parameter Name	Description	Sample valu e
merchantId	This is the unique id assigned to the merchant at the time of on boarding in FD Connect system	
resultUrl	This is the url in which transaction response is done	

fdcTxnId FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction status  transactionType Transaction Type like Sale  baseCurrency Base Currency transactionCurrency Transaction Currency baseAmount Base Amount transactionAmount Transaction Amount balanceAmount Amount available for withdraw paymentMethod Payment Method convFee Convenience fee applied on base amount transactionStatus Transaction Status transactionStatusCode Status code transactionStatusDescription transactionDateTime Transaction Date and Time udf1 User defined field udf2 User defined field udf4 User defined field udf5 User defined field udf6 User defined field udf7 User defined field udf7 User defined field udf8 User defined field udf9 User defined field udf9 User defined field udf10 User defined field errorCode Error code errorMessage Error description			
baseCurrency transactionCurrency baseAmount Base Amount transactionAmount balanceAmount paymentMethod convFee Convenience fee applied on base amount transactionStatus Transaction Status transactionStatusCode transactionStatusDescription transactionDateTime udf1 User defined field udf3 User defined field udf5 User defined field udf6 User defined field udf7 User defined field udf7 User defined field udf8 User defined field udf9 User defined field udf10 User defined field errorCode Error code	fdcTxnld	FDConnect to merchant after successful/failure of the transaction. The same need to pass to get the transaction	
transactionCurrency baseAmount Base Amount transactionAmount balanceAmount paymentMethod ConvFee Convenience fee applied on base amount taxFee transactionStatus Transaction Status transactionStatusCode transactionStatusDescri ption transactionDateTime udf1 User defined field udf2 User defined field udf4 User defined field udf5 User defined field udf6 User defined field udf7 User defined field udf7 User defined field udf8 User defined field udf9 User defined field udf10 Error code	transactionType	Transaction Type like Sale	
baseAmount transactionAmount Transaction Amount balanceAmount paymentMethod convFee Convenience fee applied on base amount taxFee transactionStatus Transaction Status transactionStatusCode transactionStatusDescri ption transactionDateTime udf1 User defined field udf3 User defined field udf4 User defined field udf5 User defined field udf6 User defined field udf7 User defined field udf7 User defined field udf8 User defined field udf9 User defined field udf10 User defined field errorCode Error code	baseCurrency	Base Currency	
transactionAmount Amount available for withdraw paymentMethod Payment Method convFee Convenience fee applied on base amount taxFee transactionStatus Transaction Status transactionStatusCode Status code transactionStatusDescri ption transactionDateTime Transaction Date and Time udf1 User defined field udf2 User defined field udf3 User defined field udf4 User defined field udf5 User defined field udf6 User defined field udf7 User defined field udf8 User defined field udf9 User defined field udf9 User defined field udf1 User defined field udf9 User defined field udf9 User defined field udf10 User defined field errorCode Error code	transactionCurrency	Transaction Currency	
balanceAmount Amount available for withdraw paymentMethod Payment Method convFee Convenience fee applied on base amount taxFee transactionStatus Transaction Status transactionStatusCode Status code transactionStatusDescri ption transactionDateTime Transaction Date and Time udf1 User defined field udf2 User defined field udf3 User defined field udf4 User defined field udf5 User defined field udf6 User defined field udf7 User defined field udf8 User defined field udf9 User defined field udf9 User defined field udf1 User defined field udf9 User defined field udf9 User defined field udf10 User defined field errorCode Error code	baseAmount	Base Amount	
paymentMethod Payment Method  convFee Convenience fee applied on base amount  taxFee  transactionStatus Transaction Status  transactionStatusCode Status code  transactionStatusDescri ption  transactionDateTime Transaction Date and Time  udf1 User defined field  udf2 User defined field  udf3 User defined field  udf4 User defined field  udf5 User defined field  udf6 User defined field  udf7 User defined field  udf8 User defined field  udf9 User defined field  udf1 User defined field  udf9 User defined field  udf10 User defined field  errorCode Error code	transactionAmount	Transaction Amount	
convFee Convenience fee applied on base amount taxFee transactionStatus Transaction Status transactionStatusCode Status code transactionStatusDescri ption transactionDateTime Transaction Date and Time  udf1 User defined field  udf2 User defined field  udf3 User defined field  udf4 User defined field  udf5 User defined field  udf6 User defined field  udf7 User defined field  udf8 User defined field  udf9 User defined field  udf10 User defined field  errorCode Error code	balanceAmount	Amount available for withdraw	
transactionStatus  transactionStatusCode  transactionStatusDescri ption  transactionDateTime  udf1  User defined field  udf2  User defined field  udf4  User defined field  udf5  User defined field  udf6  User defined field  udf7  User defined field  udf7  User defined field  udf9  User defined field  udf10  User defined field  udf10  User defined field  udf10  User defined field  Error code	paymentMethod	Payment Method	
transactionStatus  transactionStatusCode  transactionStatusDescri ption  transactionDateTime  udf1  User defined field  udf2  User defined field  udf3  User defined field  udf4  User defined field  udf5  User defined field  udf6  User defined field  udf7  User defined field  udf7  User defined field  udf9  User defined field  udf9  User defined field  udf9  User defined field  udf9  User defined field  udf1  User defined field  udf8  User defined field  udf9  User defined field  udf9  User defined field  udf10  User defined field  errorCode  Error code	convFee	Convenience fee applied on base amount	
transactionStatusDescri ption  transactionDateTime  udf1  User defined field  udf2  User defined field  udf4  User defined field  udf5  User defined field  udf6  User defined field  udf7  User defined field  udf8  User defined field  udf9  User defined field  udf9  User defined field  udf10  Error code	taxFee		
transactionStatusDescri ption  transactionDateTime Transaction Date and Time  udf1 User defined field  udf2 User defined field  udf3 User defined field  udf4 User defined field  udf5 User defined field  udf6 User defined field  udf7 User defined field  udf7 User defined field  udf8 User defined field  udf9 User defined field  errorCode Error code	transactionStatus	Transaction Status	
transactionDateTime Transaction Date and Time  udf1 User defined field  udf2 User defined field  udf3 User defined field  udf4 User defined field  udf5 User defined field  udf6 User defined field  udf7 User defined field  udf8 User defined field  udf9 User defined field  errorCode Error code	transactionStatusCode	Status code	
udf1 User defined field   udf2 User defined field   udf3 User defined field   udf4 User defined field   udf5 User defined field   udf6 User defined field   udf7 User defined field   udf8 User defined field   udf9 User defined field   udf10 User defined field   errorCode Error code		Transaction description	
udf2User defined fieldudf3User defined fieldudf4User defined fieldudf5User defined fieldudf6User defined fieldudf7User defined fieldudf8User defined fieldudf9User defined fieldudf10User defined fielderrorCodeError code	transactionDateTime	Transaction Date and Time	
udf3 User defined field   udf4 User defined field   udf5 User defined field   udf6 User defined field   udf7 User defined field   udf8 User defined field   udf9 User defined field   udf10 User defined field   errorCode Error code	udf1	User defined field	
udf4 User defined field  udf5 User defined field  udf6 User defined field  udf7 User defined field  udf8 User defined field  udf9 User defined field  udf10 User defined field  errorCode Error code	udf2	User defined field	
udf5     User defined field       udf6     User defined field       udf7     User defined field       udf8     User defined field       udf9     User defined field       udf10     User defined field       errorCode     Error code	udf3	User defined field	
udf6 User defined field   udf7 User defined field   udf8 User defined field   udf9 User defined field   udf10 User defined field   errorCode Error code	udf4	User defined field	
udf7     User defined field       udf8     User defined field       udf9     User defined field       udf10     User defined field       errorCode     Error code	udf5	User defined field	
udf8 User defined field udf9 User defined field udf10 User defined field errorCode Error code	udf6	User defined field	
udf9 User defined field udf10 User defined field errorCode Error code	udf7	User defined field	
udf10 User defined field errorCode Error code	udf8	User defined field	
errorCode Error code	udf9	User defined field	
	udf10	User defined field	
errorMessage Error description	errorCode	Error code	
	errorMessage	Error description	

### Sample code:

Language	Δ	JAVA
-anguage	_	