RBP FINIVIS
RURAL BANKING PROGRAMME

PAYOUT API

Version 1.0

Author: Monika Virmani

Content

Description	Base URL
Sandbox	https://sandbox.rbpfinivis.com
Production	After completing the audit process production URL will be shared

Description	Bank Name
Sandbox Bank PipeName	ICICI

Partner Authorization

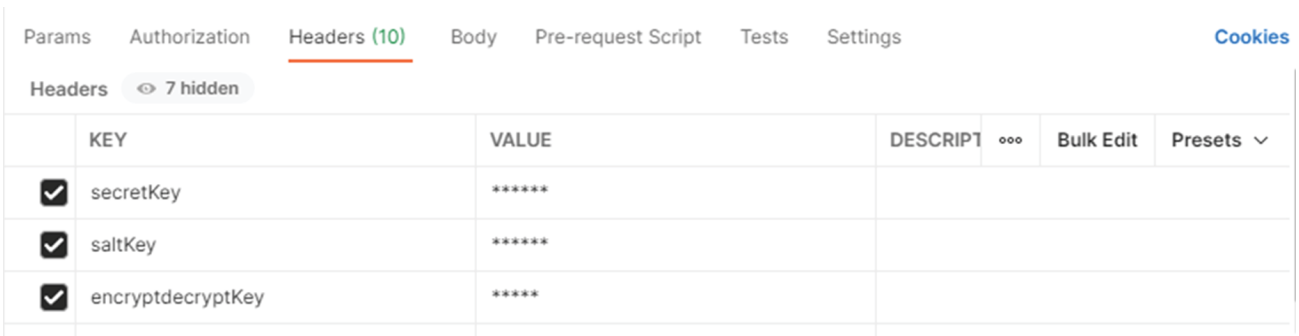
URL - /api/Signature/authorize

Method Type – HTTP POST

Description – Authorization API is used to check the request coming to the RBP switch is from the authorized partner or not.

Headers Parameter	Type	Data Type	Description	Requisite
secretKey	Authorization	Alphanumeric	Secret Key partner will get at the time of partner's onboarding. This key is used for authorization purpose.	Mandatory
saltKey	Authorization	Alphanumeric	Salt Key partner will get at the time of partner's onboarding. This key is used for authorization purpose.	Mandatory
encryptdecryptKey	Authorization	Alphanumeric	Encrypt Decrypt Key partner will get at the time of partner's onboarding. This key is used for authorization purpose as well as to encrypt the payload.	Mandatory

Postman Sample Request



Sample Response


```
{
   "isSuccess": true,
   "message": "Success",
   "data": {
      "token": "eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNobWFjLXNoYTI1NiIsInR5cCI6IkpXVCJ9.eyJQYXJ0bmVySWQiOiI2M2M2MGMyMS1hZjBhLTQ3MzItYjdlNC0wNjA0MjkwY2EwMzEiLCJzZWNyZXRLZXkiOiIxMjM0Iiwic2FsdEtleSI6IjQ1NiIsImVuY3J5cHRkZWNyeXB0S2V5IjoiYWJjZCIsIlJPTEUiOiJQQVJUTkVSIiwiSVBBZGRyZXNzIjoiMTIyLjE3My4xNzIuMjM5IiwiaXNNQVRNQXZhaWxhYmxlIjoiVHJ1ZSIsImlzQWVQU0F2YWlsYWJsZSI6IlRydWUiLCJhZXBzUGlwZTEiOiJUcnVlIiwiYWVwc1BpcGUyIjoiRmFsc2UiLCJhZXBzUGlwZTMiOiJGYWxzZSIsImFlcHNQaXBlNCI6IkZhbHNlIiwiYWVwc1BpcGU1IjoiRmFsc2UiLCJtYXRtUGlwZTEiOiJUcnVlIiwibWF0bVBpcGUyIjoiRmFsc2UiLCJtYXRtUGlwZTMiOiJGYWxzZSIsIm1hdG1QaXBlNCI6IkZhbHNlIiwibWF0bVBpcGU1IjoiRmFsc2UiLCJtaW5BbW91bnQiOiIxMC4wMCIsIm1heEFtb3VudCI6IjEwMDAuMDAiLCJuYmYiOjE2MTcxMjczNzEsImV4cCI6MTYxNzEyODU3MSwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo1MDAwLyIsImF1ZCI6Imh0dHA6Ly9sb2NhbGhvc3Q6NTAwMC8ifQ.MmAAK3JpkNOAq2fYSs9uaCNEpMILAnpJt7y1U2fPxO8"
   },
   "statusCode": "000"
}
```


Acquire State

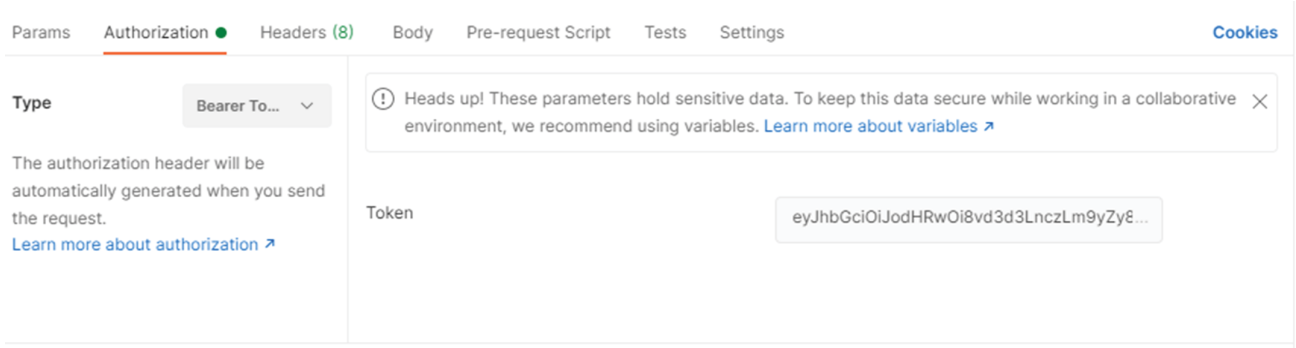
URL - /api/Common/acquireState

Method Type – HTTP GET

Description – Acquire state is used to check the state list which helps in merchant onboarding.

Headers Parameter	Type	Data Type	Description	Requisite
Bearer Token	Authorization	Alphanumeric	token value which you received from the partner authorization API	Mandatory

Postman Sample Request



Sample Response


```
{
   "isSuccess": true,
   "message": "Success",
   "data": [
      {
         "key": "bf6ead33-4394-4b8d-a942-026954c0dd9a",
         "value": "Chandigarh"
      },
      {
         "key": "4dbb03d8-bb06-4074-9b02-063d0f1880f6",
         "value": "Sikkim"
      },
      {
         "key": "113fe51f-ee39-40cc-b9d8-09fad6d745b3",
         "value": "Uttarakhand"
      },

      {
         "key": "47ea36a9-ab1c-4c06-85ea-19a85b617318",
         "value": "Nagaland"
      }
   ],
   "statusCode": "000"
}
```


Acquire District

URL - /api/Common/acquireDistrictViaState

Method Type – HTTP POST

Description – Acquire district is used to check the district list under particular state which helps in merchant onboarding.

Authorization Type – Bearer Token (token value which we received from the partner authorization API call need to pass here.)

Headers Parameter	Type	Data Type	Description	Requisite
Bearer Token	Authorization	Alphanumeric	token value which you received from the partner authorization API	Mandatory

Request Parameter	Type	Data Type	Description	Requisite
stateId	Body Param	Alphanumeric	State Key Id need to be encrypted via the encrypt method given below.	Mandatory

Content Type – application/json

Body Type - JSON raw format

Request - All the JSON raw format payload must be encrypted by the keys provided at the time of partner onboarding and it should be pass in a data param.

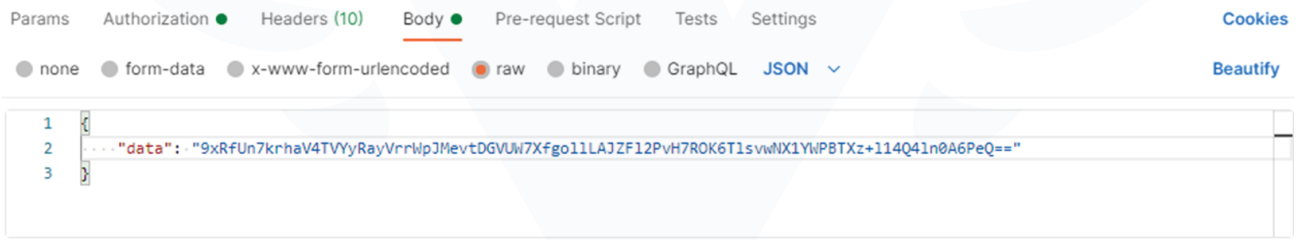
Sample Plain Text Payload –

{
"stateId":"2a057be3-eaad-4c0b-9df7-342aa7331847"
}

Sample Encrypted Payload –

9xRfUn7krhaV4TVYyRayVrrWpJMevtDGVUW7XfgollLAJZ-
Fl2PvH7ROK6TlsvwNX1YWPBTXz+l14Q4ln0A6PeQ==

Postman Sample Request



Sample Response

{
"isSuccess": true,
"message": "Success",
"data": [
{
"key": "2d3b553b-9aeb-49aa-93a1-e6e56cfede90",
"value": "Ahmednagar"
},
{
"key": "f9593fb2-fd54-4400-8176-ed9994c4e61c",
"value": "Akola"
},
{
"key": "8aa3aa3f-2559-424e-9002-579ba46d8639",
"value": "Amravati"
}
],
"statusCode": "000"
}

Merchant Registration

URL - /api/Onboarding/merchantRegistration

Method Type – HTTP POST

Description – Merchant onboarding is used to register every merchant so that they can access our AePS service.

Authorization Type – Bearer Token (token value which we received from the partner authorization API call need to pass here.)

Headers Parameter	Type	Data Type	Description	Requisite
Bearer Token	Authorization	Alphanumeric	token value which you received from the partner authorization API	Mandatory

Request Parameter	Type	Data Type	Description	Requisite
name	Body Param	Alphabetical	Merchant complete name	Mandatory
emailId	Body Param	Alphanumeric	Merchant email address	Mandatory
mobileNo	Body Param	Numeric	Merchant mobile number	Mandatory
shopName	Body Param	Alphabetical	Merchant shop name	Mandatory
address1	Body Param	Alphanumeric	Merchant address1	Mandatory
address2	Body Param	Alphanumeric	Merchant address 2	Optional
pincode	Body Param	Numeric	Merchant pincode	Mandatory
aadhaarNo	Body Param	Numeric	Merchant aadhaar card number	Mandatory
panNo	Body Param	Alphanumeric	Merchant Pancard number	Mandatory
stateId	Body Param	Alphanumeric	Merchant State Id (value get from acquireState API)	Mandatory
districtId	Body Param	Alphanumeric	Merchant district id (value get from acquireDistrictViaState API	Mandatory

Content Type – application/json

Body Type - JSON raw format

Request - All the JSON raw format payload must be encrypted by the keys provided at the time of partner onboarding and it should be pass in a data param.

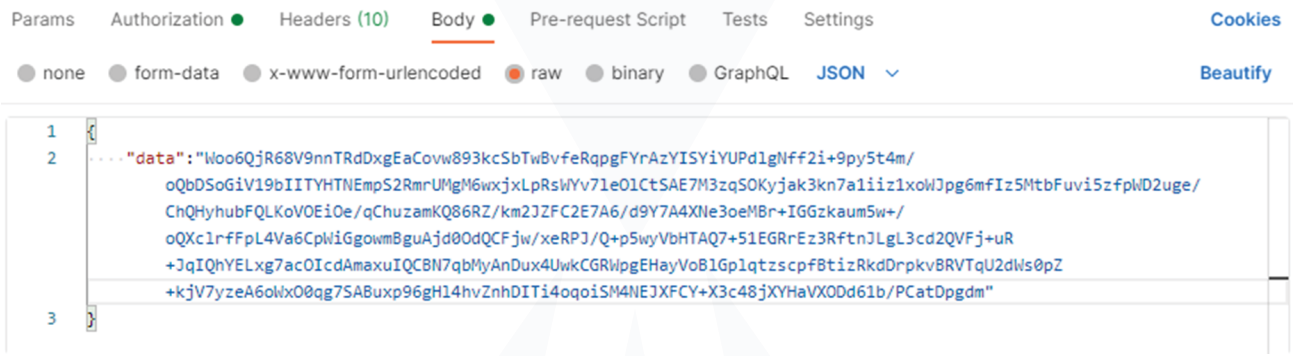
Sample Plain Text Payload –


```
{
 "name": "D*********",
 "emailId": "D*********@gmail.com",
 "mobileNo": "99*********9",
 "shopName": "RBP",
 "address1": "Office no 1",
 "address2": "Panchkula",
 "pincode": "135117",
 "aadhaarNo":"33********42vfvdf",
 "panNo": "BGVPG0554D",
 "stateId":"58b0a545-4bb9-480f-aeed-4b849a2d99cc",
 "districtId":"67b6ce46-e643-4da3-8a6d-2c364c2c0d86",
 }
```


Sample Encrypted Payload –

Woo6QjR68V9nnTRdDxgEaCovw893kcSbTwBvfeRqpgFYrAzYISYiYUPdl-gNff2i+9py5t4m/oQbDSoGiV19bIITYHTNEmpS2RmrUMgM6wxjxLpRsWYv7leOlCtSAE7M3zqSOKyjak3kn7a1iiz1xoWJpg6mfIz5MtbFuvi5zfpWD2uge/ChQHyhubFQLKoVOEiOe/qChuzamKQ86RZ/km2JZFC2E7A6/d9Y7A4XNe3oeMBr+IGGzkaum5w+/oQXclrfFpL4Va6CpWiGgowmBguAjd0OdQCFjw/xeRPJ/Q+p5wyVbHTAQ7+51EGRrEz3RftnJLgL3cd2QVFj+uR+JqIQhYELxg7acOIcdAmaxuIQCBN7qbMyAnDux4UwkCGRWpgEHayVoBlGplqtzscpfBtizRkdDrpkvBRVTqU2dWs0pZ+kjV7yzeA6oWxO0qg7SABuxp96gHl4hvZnhDITi4oqoiSM4NEJXFCY+X3c48jXYHaVXODd61b/PCatDpgdm

Postman Sample Request



Sample Response


```
{
   "isSuccess": true,
   "message": "Success",
   "data": {
      "merchant_Id": "MERAECCEF257D47",
      "name": "D*********",
      "emailId": "Dee******t@gmail.c**",
      "mobileNo": "999*****91",
      "merchantStatus": "PK",
      "statusDescription": "Pending For KYC"
   },
   "statusCode": "000"
}
```

Merchant Status

URL - /api/Onboarding/merchantStatus

Method Type – HTTP POST

Description – Merchant status is used to get the updated status of the merchant. Merchant must be active to initiate any financial or non-financial transaction.

Authorization Type – Bearer Token (token value which we received from the partner authorization API call need to pass here.)

Headers Parameter	Type	Data Type	Description	Requisite
Bearer Token	Authorization	Alphanumeric	token value which you received from the partner authorization API	Mandatory

Request Parameter	Type	Data Type	Description	Requisite
merchant_id	Body Param	Alphanumeric	Merchant id recived from merchant registeration API	Mandatory
emailId	Body Param	Alphanumeric	Merchant email address	Mandatory
mobileNo	Body Param	Numeric	Merchant mobile number	Mandatory

Content Type – application/json

Body Type - JSON raw format

Request - All the JSON raw format payload must be encrypted by the keys provided at the time of partner onboarding and it should be pass in a data param.

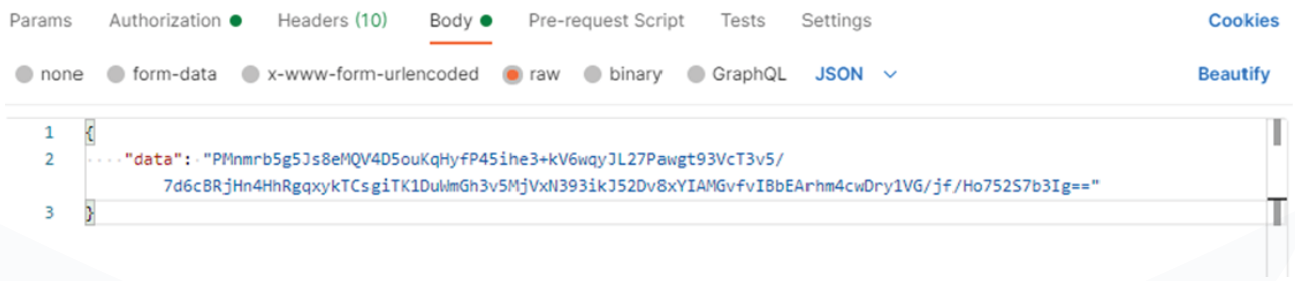
Sample Plain Text Payload –

{
"merchant_id":"MERAECCEF257D47",
"emailId":"D*****t@gmail.c**",
"mobileNo":"999*****91"
}

Sample Encrypted Payload –

PMnmrb5g5Js8eMQV4D5ouKqHyfP45ihe3+kV6wqyJL27Paw-gt93VcT3v5/7d6cBRjHn4HhRgqxykTCsgiTK1DuWmGh3v5MjVxN393ikJ52Dv8xYIAMGvfvIBbEArhm4cwDry1VG/jf/Ho752S7b3Ig==

Postman Sample Request



Sample Response


```
{
   "isSuccess": true,
   "message": "Success",
   "data":
{
      "merchant_Id": "MERAECCEF257D47",
      "statusDescription": "Pending For KYC",
      "statusCode": "PK",
      "name": "D*********",
      "emailId": "D********@gmail.com",
      "mobileNo": "99*********91",
      "pinCode": "135117",
      "address": "Office no 1,Panchkula",
      "districtName": "Kurukshetra",
      "stateName": "Haryana",
      "aadhaarNo": "33********42",
      "pan": "BGVPG0554J"
   },
   "statusCode": "000"
}
```

Payout Composite

URL - api/payout/payout

Method Type – HTTP POST

Description – Payout Composite API is used to send money to the beneficiary account on real time.

Authorization Type – Bearer Token (token value which we received from the partner authorization API call need to pass here.)

Headers Parameter	Type	Data Type	Description	Requisite
Bearer Token	Authorization	Alphanumeric	token value which you received from the partner authorization API	Mandatory

Request Parameter	Type	Data Type	Description	Requisite
merchant_Id	Body Param	Alphanumeric	Merchant id received from merchant registration API	Mandatory
emailId	Body Param	Alphanumeric	Merchant email address	Mandatory
mobileNo	Body Param	Numeric	Merchant mobile number	Mandatory
accountHolderName	Body Param	Numeric	Beneficiary Account Holder Name	Mandatory
accountNo	Body Param	Numeric	Beneficiary Account Number	Mandatory
ifscCode	Body Param	Alphanumeric	Beneficiary ifsc Code	Mandatory
txnAmount	Body Param	Numeric	Beneficiary txn Amount	Mandatory
partnerRefId	Body Param	Alphanumeric	Unique reference id for every request	Mandatory

Content Type – application/json

Body Type - JSON raw format

Request - All the JSON raw format payload must be encrypted by the keys provided at the time of partner onboarding and it should be pass in a data param.

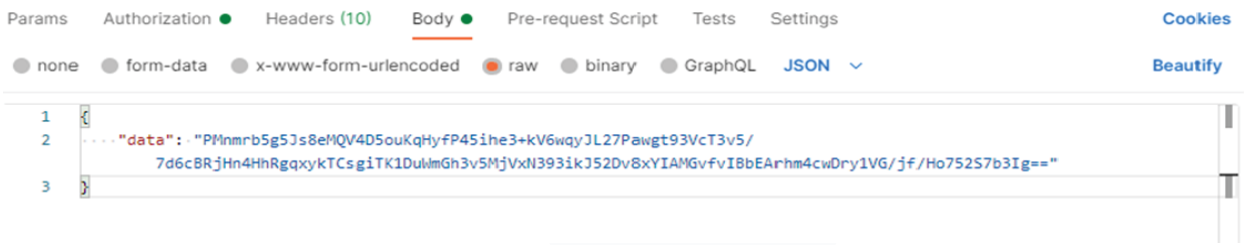
Sample Plain Text Payload –

{
"txnAmount":5,
"merchant_Id": "MER1982d29330bc",
"accountNo":"10061076247",
"ifscCode":"IDFB0021374",
"emailId":"abc@gmail.com",
"accountHolderName":"RBP Finivis Pvt Ltd",
"mobileNo":"8010401404",
"partnerRefId":"5032AE41-AE4E-4F1C-9092-D5AD2336A445"
}

Sample Encrypted Payload –

dvkBARmMYhiEeSnfDMZrQZzTHwh9MW3UAvOWTkpplTLhw5fLqQzd/lF-naa9J5gQ9AZqBg8KxBZCCEE1+Up9D2BIe7U/v6li+idDumz43Vt3VhTbgc16XhLjVWy7LAsaljJ3gG8r7ALDA2u6yKDbGij7C/7SQeTIXEAJ/awiTHhRAVleGnR8zYKPMFJV3oGxIsQskuUrv90PoW9NIK2OP219NqRh17Nw2VlD3bNI+76pvbAbkWxuaso5HZs6h+fCfMi2NFFXb71LL6FbR5bdBk7+KPdwiy6v/bRLdtkbwKLabgRIzI3fwGv+x7T46ARGKFZvkEJ+ifszV9RZa+HLGeKN2Soa7Tc4dmu6zq5ZSXZ8Un0pydJfaq9N0ITv3KSxK

Postman Sample Request



Sample Response

{
"isSuccess": true,
"message": "Success",
"data": {
"stanNo": "c550e946-5a49-4a02-b2ec-4377e0d61b53",
"txnAmount": 5.0,
"txnFee": 1.0,
"txnDate": "2022-10-29T18:48:56.5612323+05:30",
"merchant_Id": "MER1982d29330bc",
"beneAccountNo": "10061076247",
"beneIfsccode": "IDFB0021374",
"beneEmail": "abc@gmail.com",
"beneName": "RBP Finivis Pvt Ltd",
"beneMobileNo": "8010401404",
"partnerRefId": "5032AE41-AE4E-4F1C-9092-D5AD2336A445",
"txnStatus": "Success",
"operatorRefId": "pout_KZbLBUBIpRlJgM"
},
"statusCode": "000"
}

Payout Composite Status Check

URL - /api/TxnReport/PayoutTxnStatus

Method Type – HTTP POST

Description – Payout Composite API status check api is used to check the final status of the transaction.

Authorization Type – Bearer Token (token value which we received from the partner authorization API call need to pass here.

Headers Parameter	Type	Data Type	Description	Requisite
Bearer Token	Authorization	Alphanumeric	token value which you received from the partner authorization API	Mandatory

Request Parameter	Type	Data Type	Description	Requisite
PartnerRefIdStan	Body Param	Alphanumeric	Partner can pass his reference id that was generated at the time of initiating the transactions OR RBP FINIVIS stan id or number which he received in the response)	Mandatory

Content Type – application/json

Body Type - JSON raw format

Request - All the JSON raw format payload must be encrypted by the keys provided at the time of partner onboarding and it should be pass in a data param.

Sample Plain Text Payload –

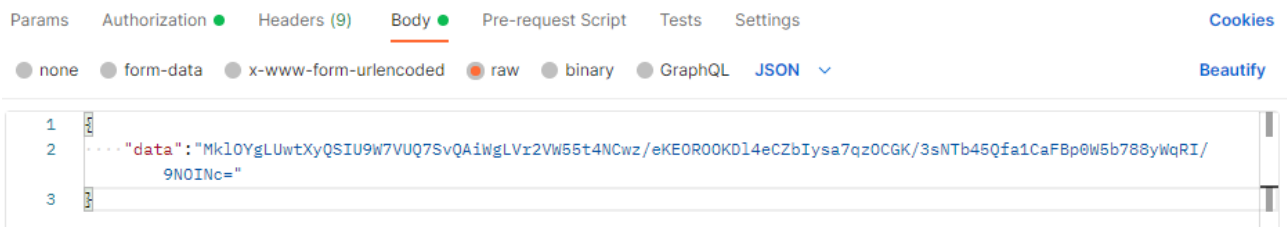
{
"partnerRefIdStan":"D8153C70-213B-4F26-AE04-026EC1E79B60"

}

Sample Encrypted Payload –

oxvMxfSxsS0D9nkiZLZ7gnugoEUDI8iFmZguyrD6mBF7bvdio-By8j6acJESxfsWAuZB9IA7ImH1W6WbYBW6ihTaEFJm/pTXic6CnCCJGsoDC8swHy50KCoKZ1v

Postman Sample Request



Sample Response


```
{
   "isSuccess":true,
   "message": "Success",
   "data": {
      "stanNo": "c550e946-5a49-4a02-b2ec-4377e0d61b53",
      "txnAmount": 5.0,
      "txnFee": 1.0,
      "txnDate": "2022-10-29T18:48:56.5612323+05:30",
      "merchant_Id": "MER1982d29330bc",
      "beneAccountNo": "10061076247",
      "beneIfsccode": "IDFB0021374",
      "beneEmail": "abc@gmail.com",
      "beneName": "RBP Finivis Pvt Ltd",
      "beneMobileNo": "8010401404",
      "partnerRefId": "5032AE41-AE4E-4F1C-9092-D5AD2336A445",
      "txnStatus": "Success",
      "operatorRefId": "pout_KZbLBUBlpRlJgM"
   },
   "statusCode": "000"
}
```

Status Code

Status Code	Description
000	Success
001	Failed
002	Timeout / Pending
003	Email id is required
004	Invalid email address
005	Mobile number is required
006	Invalid mobile number
007	Amount is required
008	Amount cannot be less than 0
009	StateId is required
010	Address is missing
011	Aadhaar number is missing
012	Invalid aadhaar number
013	PAN is missing
014	Invalid PAN
015	Pincode is missing
016	Invalid pincode
017	Shop name is missing
018	Merchant id is missing
019	Invalid merchant id
020	Customer id is required
021	Bank IIN is missing
022	Invalid Bank IIN
023	Partner reference id is missing
024	Latitude is missing
025	Invalid latitude
026	Longitude is missing
027	Invalid longitude
028	Fingerprint data is missing
029	Customer mobile number is missing
030	Customer name is missing
031	Transaction code is missing
032	Pipeid is missing
033	Merchant name is missing
034	Partner id is missing
035	Transaction reference id or STAN is missing
036	Request is missing
037	Parameters is missing
038	Sdk version is missing
039	Invalid transaction reference id or STAN
042	Invalid merchant details
043	Invalid partner details

Status Code	Description
044	Merchant not found
045	Your location is inappropriate to initiate any transaction
046	Invalid credentials
047	Session or Token expire
048	Something went wrong, while decrypting
049	Invalid transaction code
050	Invalid customer id
051	Invalid sdk version
052	Partner reference id length is too large max size is (25)
053	Terminal id is pending for approval
054	Header is missing
055	District is required
056	Invalid district id
057	Role not match
058	Encrypt Decrypt key is invalid
059	Invalid payload
060	Invalid pipeid
061	Merchant not active to initiate any transactions
062	Partner is not active to initiate any transactions
063	Invalid vendor log id
064	Vendor log is required
065	Lazy load limit is missing
066	Lazy load offset is missing
067	For non-financial transaction amount must be 0
068	It seems your IP address does not match our database
069	Invalid file format
070	Password is missing
071	Invalid password
072	Password must contain special charchter with minimum length of 6
073	Oops! something unexcepted happens, request you to wait
074	Already registered
075	Ohh! service is down at this time
076	Your account is blocked due to many unsuccessful try
077	Gateway timeout
078	Token is required
079	Merchant device id is missing
080	Invalid pid block
081	Pipename is missing
082	Pipename is invalid
083	Invalid amount
084	Invalid customer details
085	No Record Found
086	Status already updated for this transaction id
087	Invalid stateid
400	Seems a bad request

Status Code	Description
401	You are unauthorized
404	URL invlaid or URL not found
405	Method not allowed
408	Request timeout
415	Unsupported media tag
500	Internal server error
501RBP	Location permission not granted
502RBP	Parameter are missing
503RBP	Invalid Transaction Type
504RBP	Inspect your internet connection may be it is down
505RBP	Server flaw
506RBP	Oops! something unexcepted happens, request you to wait
507RBP	Message from server
508RBP	Back key press

Encrypt and Decrypt Sample Code

For C#

```
#region Using

using System;
using System.Security.Cryptography;
using System.Text;

#endregion

namespace YourApp.Security.Cryptography
{
   /// <summary>
   /// Rijndael Encryptor / Decryptor Helper
   ///
   /// <remarks>
   /// Created by: Jafet Sanchez
   /// Last Update: [date],[author],[description]
   ///
   public class RijndaelCrypt
   {
      #region Private/Protected Member Variables

      /// <summary>
      /// Decryptor
      ///
      private readonly ICryptoTransform _decryptor;

      /// <summary>
      /// Encryptor
      ///
      private readonly ICryptoTransform _encryptor;

      /// <summary>
      /// 16-byte Private Key
      ///
```

```
private static readonly byte[] IV = Encoding.UTF8.GetBytes("ThisIsUrPassword");

/// <summary>
/// Public Key
///
private readonly byte[] _password;

/// <summary>
/// Rijndael cipher algorithm
///
private readonly RijndaelManaged _cipher;

#endregion

#region Private/Protected Properties

private ICryptoTransform Decryptor { get { return _decryptor; } }
private ICryptoTransform Encryptor { get { return _encryptor; } }

#endregion

#region Private/Protected Methods
#endregion

#region Constructor

/// <summary>
/// Constructor
///
/// <param name="password">Public key
public RijndaelCrypt(string password)
{
   //Encode digest
   var md5 = new MD5CryptoServiceProvider();
   _password = md5.ComputeHash(Encoding.ASCII.GetBytes(password));

   //Initialize objects
   _cipher = new RijndaelManaged();
   _decryptor = _cipher.CreateDecryptor(_password, IV);
   _encryptor = _cipher.CreateEncryptor(_password, IV);

}

#endregion

#region Public Properties
#endregion

#region Public Methods

/// <summary>
/// Decryptor
///
/// <param name="text">Base64 string to be decrypted
/// <returns>
public string Decrypt(string text)
```

```
        {
            try
            {
                byte[] input = Convert.FromBase64String(text);

                var newClearData = Decryptor.TransformFinalBlock(input, 0, input.Length);
                return Encoding.ASCII.GetString(newClearData);
            }
            catch (ArgumentException ae)
            {
                Console.WriteLine("inputCount uses an invalid value or inputBuffer has an invalid offset length. "
+ ae);
                return null;
            }
            catch (ObjectDisposedException oe)
            {
                Console.WriteLine("The object has already been disposed." + oe);
                return null;
            }


        }

        /// <summary>
        /// Encryptor
        ///
        /// <param name="text">String to be encrypted
        /// <returns>
        public string Encrypt(string text)
        {
            try
            {
                var buffer = Encoding.ASCII.GetBytes(text);
                return Convert.ToBase64String(Encryptor.TransformFinalBlock(buffer, 0, buffer.Length));
            }
            catch (ArgumentException ae)
            {
                Console.WriteLine("inputCount uses an invalid value or inputBuffer has an invalid offset length. "
+ ae);
                return null;
            }
            catch (ObjectDisposedException oe)
            {
                Console.WriteLine("The object has already been disposed." + oe);
                return null;
            }

        }

        #endregion
    }
}
```

For Java

```
package com.yourpackage.security;
```

```
import android.util.Log;
import android.util.Base64;

import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class RijndaelCrypt {

   public static final String TAG = "YourAppName";

   private static String TRANSFORMATION = "AES/CBC/PKCS7Padding";
   private static String ALGORITHM = "AES";
   private static String DIGEST = "MD5";

   private static Cipher _cipher;
   private static SecretKey _password;
   private static IvParameterSpec _IVParamSpec;

   //16-byte private key
   private static byte[] IV = "ThisIsUrPassword".getBytes();


   /**
    Constructor
    @password Public key

   */
   public RijndaelCrypt(String password) {

      try {

         //Encode digest
         MessageDigest digest;
         digest = MessageDigest.getInstance(DIGEST);
         _password = new SecretKeySpec(digest.digest(password.getBytes()), ALGORITHM);

         //Initialize objects
         _cipher = Cipher.getInstance(TRANSFORMATION);
         _IVParamSpec = new IvParameterSpec(IV);

      } catch (NoSuchAlgorithmException e) {
         Log.e(TAG, "No such algorithm " + ALGORITHM, e);
      } catch (NoSuchPaddingException e) {
         Log.e(TAG, "No such padding PKCS7", e);
      }
   }
```

```
/**
Encryptor.

@text String to be encrypted
@return Base64 encrypted text

*/
public String encrypt(byte[] text) {

   byte[] encryptedData;

   try {

      _cipher.init(Cipher.ENCRYPT_MODE, _password, _IVParamSpec);
      encryptedData = _cipher.doFinal(text);

   } catch (InvalidKeyException e) {
      Log.e(TAG, "Invalid key  (invalid encoding, wrong length, uninitialized, etc).", e);
      return null;
   } catch (InvalidAlgorithmParameterException e) {
      Log.e(TAG, "Invalid or inappropriate algorithm parameters for " + ALGORITHM, e);
      return null;
   } catch (IllegalBlockSizeException e) {
      Log.e(TAG, "The length of data provided to a block cipher is incorrect", e);
      return null;
   } catch (BadPaddingException e) {
      Log.e(TAG, "The input data but the data is not padded properly.", e);
      return null;
   }

   return Base64.encodeToString(encryptedData,Base64.DEFAULT);

}

/**
Decryptor.

@text Base64 string to be decrypted
@return decrypted text

*/
public String decrypt(String text) {

   try {
      _cipher.init(Cipher.DECRYPT_MODE, _password, _IVParamSpec);

      byte[] decodedValue = Base64.decode(text.getBytes(), Base64.DEFAULT);
      byte[] decryptedVal = _cipher.doFinal(decodedValue);
      return new String(decryptedVal);


   } catch (InvalidKeyException e) {
      Log.e(TAG, "Invalid key  (invalid encoding, wrong length, uninitialized, etc).", e);
      return null;
   } catch (InvalidAlgorithmParameterException e) {
```

```
            Log.e(TAG, "Invalid or inappropriate algorithm parameters for " + ALGORITHM, e);
            return null;
        } catch (IllegalBlockSizeException e) {
            Log.e(TAG, "The length of data provided to a block cipher is incorrect", e);
            return null;
        } catch (BadPaddingException e) {
            Log.e(TAG, "The input data but the data is not padded properly.", e);
            return null;
        }

    }

}
```

Merchant Status Code

Merchant Status Code	Description
PK	Pending For KYC
A	Active
R	Rejected
D	Deactive

Escalation Matrix

Level 1	devsupport@rbpfinivis.com
Level 2	tech@rbpfinivis.com

Useful links

- *https://www.rbpfinivis.com/generic/converter*
- *https://www.rbpfinivis.com/generic/phpcode*
- *https://www.rbpfinivis.com/generic*

RBP FINIVIS
RURAL BANKING PROGRAMME

Thank You!

PAYOUT API

Version 1.0