# Gateway Requirements

# Index

[Gateway Requirements](#_Toc15631)

[Index](#_Toc25168)

[ChangeLog](#_Toc4965)

[1. Introduction](#_Toc26827)

[1.1. Purpose](#_Toc3353)

[1.2. Target Audience](#_Toc22363)

[2. Overview](#_Toc9475)

[2.1. Communication Protocal](#_Toc245)

[2.2. Definitions](#_Toc32720)

[2.2.1. PeraHub](#_Toc26327)

[2.2.2. BananaPay(BNNP)](#_Toc29679)

[2.3. Syntax](#_Toc32592)

[2.3.1. Overview](#_Toc6559)

[2.3.2. XML Structure](#_Toc22543)

[2.3.3. MsgHeader](#_Toc27798)

[2.3.4. MsgBody](#_Toc6058)

[2.3.5. Signature](#_Toc5097)

[2.4. Encoding](#_Toc16492)

[3. API Functions](#_Toc18963)

[3.1. Sign-Up Request](#_Toc26302)

[3.2. Payment Request](#_Toc28236)

[3.3. Refund Request](#_Toc24481)

[3.4. Transaction Query](#_Toc20887)

[4. Exceptions](#_Toc20488)

[5. Statement Check](#_Toc9511)

# ChangeLog

# Introduction

## Purpose

This document is a description of the API Functions that BananaPay hope PeraHub can provide.

## Target Audience

PeraHub developers And BananaPay Developers.

# Overview

## Communication Protocal

All api functions uses standard protocols and data structure.

The data should be submitted to EndPost using POST method through HTTPS in XML FORMAT.

Response to the request are returned synchronously on the same connection from which the request originated

# Definitions

## PeraHub

## BananaPay(BNNP)

## Message

# Message Overview

## Message List

### Message Number

Message Number must follow the rule as “XXXX.YYY.ZZZ.NN”,

* XXXX represents the system id, e.g. ‘bnnp’ or ‘pera’
* YYY represents Message name, e.g. ‘101’, this will diff message type.
* ZZZ is reserved, now should be filled as ‘001’
* NN represents the version no, now should be filled as ‘01’

### Message List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SeqNo | Message Number | Message  Name | Message  Direction | Need Signature | Need statement check |
| 1 | Bnnp.101.001.01 | Authentication and sign-up application request | BNNP -> pera hub | Y | N |
| 2 | Bnnp.102.001.01 | Authentication and sign-up application response | Pera hub -> bnnp | Y | N |
| 3 | Bnnp.103.001.01 | Termination request | BNNP -> pera hub | Y | N |
| 4 | Bnnp.104.001.01 | Termination response | Pera hub -> bnnp | Y | N |
| 5 | Epcc.201.001.01 | Sign-up payment request | Bnnp->pera hub | Y | Y |
| ~~6~~ | ~~Epcc.231.001.01~~ |  |  |  |  |
| ~~7~~ | ~~Epcc.233.001.01~~ |  |  |  |  |
| ~~8~~ | ~~Epcc.241.001.01~~ |  |  |  |  |
| 9 | Epcc.205.00.01 | Refund request | Bnnp->pera hub | Y | Y |
| 10 | Epcc.211.001.01 | Payment-to-bank request | Bnnp->pera hub | Y | Y |
| 11 | Epcc.206.001.01 | Payment receipt message | Pera hub -> bnnp | Y | Y |
| 12 | Epcc.242.001.01 |  |  |  |  |
| 13 | Epcc.243.001.01 |  |  |  |  |
| 14 | Epcc.207.001.01 | Sign-up payment result message | Prea hub -> bnnp | Y | Y |
| 15 | epcc.244.001.01 |  |  |  |  |
| 16 | Epcc.208.001.01 | Refund result message | Pera hub -> bnnp | Y | Y |
| 17 | Epcc.213.001.01 | Payment-to-bank result message | Pera hub -> bnnp | Y | Y |
| 18 | Epcc.251.001.01 |  |  |  |  |
| 19 |  |  |  |  |  |
| 20 |  |  |  |  |  |
| 21 |  |  |  |  |  |
| 22 |  |  |  |  |  |
| 23 |  |  |  |  |  |
| 24 |  |  |  |  |  |
| 25 | Epcc.301.001.01 | Transaction query request | Bnnp->pera hub | Y | N |
| 26 | Epcc.302.001.01 | Transaction query response | Pera hub -> bnnp | Y | N |
| 27 |  |  |  |  |  |
| 28 |  |  |  |  |  |
| 29 |  |  |  |  |  |
| 30 |  |  |  |  |  |
| 31 |  |  |  |  |  |
|  |  |  |  |  |  |

## Data Type

Data type will be described like: MAX[MIN]<Size><Type>

* Min: Optional, if this word shows, that this means this field’s length must be EXACTLY equal to Size.
* Size: Must, this part defines the maximum length of this field value.
* Type: MUST, this part defines the type of this field’s value. And the enumerations are shown in Figure

|  |  |  |
| --- | --- | --- |
| SeqNo | Type | Meaning |
| 1 | Code |  |
| 2 | Text | Number, characters |
| 3 | NumericText | only numbers |
| 4 | AmountText | Amount, the format is “Currency Code” +” Integer” +”.” +” Integer”  e.g. PHP12345.67  e.g. CNY12345.67 |
| 5 | ISODateTime | yyyy-mm-ddThh:mm:ss  e.g. 2017-10-26T14:24:29 |

## Data Requirements

In Column “Requirements” ,

## Encoding

## Reserved Name

# Message Syntax

## Overview

This chapter describe the details of XML structure.

## XML Structure

Data will be transferred from endpoint to endpoint. XML Structure show in Figure 1.

|  |  |  |
| --- | --- | --- |
| root (XML document root element) | | Signature |
| MsgHeader | MsgBody |

Figure

The example of this XML structure is shown in Figure 2.

|  |
| --- |
| <? xml version=”1.0” encoding=”UTF-8”>  <root>  <MsgHeader>  ...  </MsgHeader>  <MsgBody>  ...  </MsgBody>  </root>\r\n  {Signature} |

Figure

## MsgHeader

MsgHeader format is “<MsgHeader>MsgHeaderContent</MsgHeader>”, the fields are listed in Figure .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Meaning | Type | Requirements | Memo |
| SndDt | Send Date | IOSDataTime | [1..1] | E.g.  2017-10-26T13:03:21 |
| MsgTp | Message No | Max15Text | [1..1] | See in |

## MsgBody

## Signature

# API Functions

## Sign-Up Request

## Payment Request

## Refund Request

## Transaction Query

# Exceptions

# Statement Check