EasyPay Online Bill Payment

Receiver Rest API Integration Guidelines for Receiver API Developers

Version 1.0.0

Revision History

Date	Version	Description	Author
2025-03-13	1.0	New Release	Bouwer Delport

Table of Contents

1. Introduction

The purpose of this document is to give an overview of the Bill Payment Rest API Receiver interface specification and its functionalities. The document is intended to provide the requirements and guidelines for the software engineer designing and implementing the related server-side API for the Receiver.

1.1 Scope

This document defines additional information needed to describe the accompanying Receiver Rest Interface specification documents.

1.2 Definitions, Acronyms and Abbreviations

Term	Definition
API	Application Program Interfcace
Authorization	A message sent to the receiver giving them the oppertunity to deside if a payment is accepted on their behalf or not.
BPS	Bill Payment Server
Bill Issuer	An organisation that issues bills to customers for payment at EasyPay Bill Payment enabled POS systems. Also known as a Receiver.
Collector	Also known as a Merchants. The collector is the instigator of an electronic transaction in which an account is paid and the entity that collects bill payments on behalf of Easypay and the Receiver. (Merchants, Retailers, Point of Sale Devices, etc.)
Customer	Clients who need to pay bills.
Epsilon	The software name of the Latest Bill Payment Server. Used to distinguish this version from other, older, versions of the application.
High Order	The receiver application that accepts authorisation requests and notifications from a BPS.
High Order Messaging	Communication between an EasyPay BPS and the receiver application
Low Order	The point of service device where the system interfaces with the customer and the payment originates.
Low Order Messaging	Communication between a Collector POS device and an EasyPay BPS.
Notification	A message sent to the receiver to tell them that a payment has been completed
OpenAPI	Specification standard https://swagger.io/specification/
POS	Collector Point Of Service device where the system interfaces with the customer and the payment originates.
Receiver	An organisation that issues bills to consumers for payment at EasyPay Bill Payment enabled Collector POS systems. Also known as a Bill Issuer.
Receiver Id	The receiver is identified by a 4 digit receiver Id that can be found in the 4 digits of the EasyPay Number immediately following the leading ‘9’. Eg 2021 is the receiver id in the EasyPay Number 9**2021**00011337826620

SAF	Store and Forward. A mechanism used by online systems to ensure the delivery of messages while allowing other processes to continue. In Epsilon this implies only that the Server may respond to the POS prior to receiving a response from the high order. It does not imply an automatic or incidental delay to the delivery of the notification. Notifications are placed in the queue and will be processed as soon as possible with high order communications being the limiting factor.
SOF File	Standard Output File. A file generated by the EasyPay Settlement System that contains a complete set of transactions received for a particular receiver for the previous day.

1.3 Related Documentation

Document Name	Description
EasypayReceiver.yaml	OpenAPI yaml Receiver Rest API interface specification
EasypayReceiver.html	This provides a more readable version from a web browser of the OpenAPI yaml interface specification
EasyPay BillPayment Receiver API.postman_collection.json	A Postman collection that can be used by the Receiver developers for basic API testing of the implementation of their server-side implementation according to the provided specification.
EasyPay Bill Payment Output File Specification	The EasyPay Output file (SOF) specification that contains the transactions that have been settled for a particular receiver the previous day.

2. Specification

2.1 Introduction

The specification is based on the OpenAPI specification (https://swagger.io/specification/). The associated yaml file can be imported into Swagger Hub or Swagger Editor and from there server and client stubs can be generated for most programming languages. The receiver must implement the server side of the API. Easypay will connect with a client-side implementation to the receiver's server implementation of this API.

Note that as per the OpenAPI specification all method and parameter names are case sensitive and should be implemented as per the provided API specification.

2.2 Security Mechanism

The Easypay client implementation is based on HTTPS with an API Key. The receiver must provide Easypay with a secret API Key that will be included in the HTTP header with an Authorization key name. Easypay can provide the originating IP address that can be used for whitelisting on the Receiver side. (If IP's change for some reason receivers will be notified)

2.3 HTTP Transport

HTTP response code 200 is used for most responses and the ResponseCode field numeric value in the response body indicates the status:
Allow payment=0
Invalid account=1
Invalid amount=2
Expired payment=3
Unknown APIkey=4
Already paid=5

Other HTTP response codes do not require a body in the response. The following HTTP response codes will fail with the associated messages to the Collector but note that some Collectors implement their own messages.
HTTP 400: High Order Data Validation Failed
HTTP 401 and 403: High Order Cannot Authenticate Request
HTTP 500, 502 and 503: High Order Institution Not Available
HTTP 504: A connection time-out occurred. Please try again later.

Any other HTTP code will fail with: This payment cannot be accepted

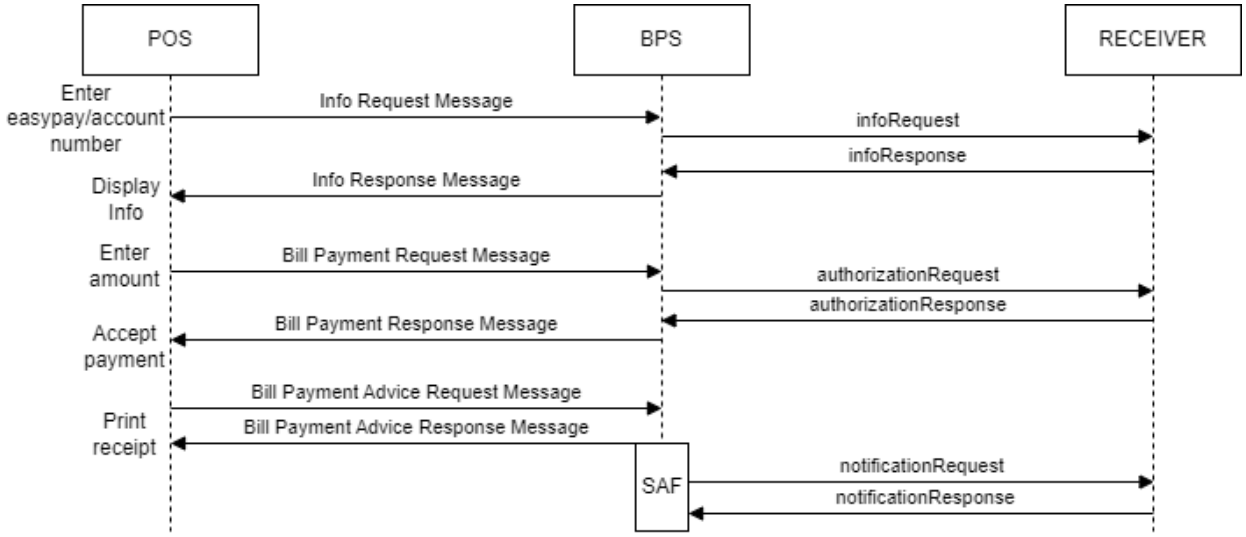
2.4 POS to Receiver Transactional Messaging flow

While terminology might change slightly, the low order (POS to BPS) and high order (BPS to Receiver) messaging are essentially mirrors of one another. Receivers must implement Information, Authorisation and Notification requests.

Collectors have the following options:

- Information, Authorisations and Notifications (Preferred implementation)
- Authorisations and Notifications (Some Collectors do not implement Information requests)

The diagrams below illustrate the full protocol including information request, authorisations and notifications.



A note on SAF: Placing notifications in a SAF queue implies not only that the Server may respond to the POS prior to receiving a response from the high order, but also that the BPS understands that the message must be delivered and that it should attempt to deliver the notification until it receives a response from the receiver. It does not imply an automatic or incidental delay to the delivery of the notification. Notifications are placed in the queue and will be processed as soon as possible with high order communications being the limiting factor.

The Bill Payment Server will not allow duplicate Bill Payment Requests from the same POS with the same reference number.

2.5 The EasyPay Receiver Service API

The EasyPay Receiver Service API specification defines 4 methods:

- ping - a basic call that just indicates if the service is active
- infoRequest - a call requesting the receiver for more information about the payment
- authorisationRequest - a call requesting the receiver to authorise the acceptance of a payment
- notification - a call notifying the receiver that a payment has been accepted on their behalf.

The detail of these API calls and their associated responses are specified in the associated EasypayReceiver.yaml (EasypayReceiver.html) files. This specification just provides an overview with additional information for clarification.

All amounts in fields are in cents without decimal, except in text message related fields.

2.5.1 Ping

The receiver must implement this method. EasyPay does not require the service to perform any task on receipt of this other than to return a body as specified with the value OK. Ie:

```
{
  "Ping": "OK"
}
```

The Epsilon Server uses this method to ascertain if the service is operational.

2.5.2 Info Request

The info request is used by EasyPay and its collector POS devices that implemented this message to request more information about the payment from the receiver. It is used especially to determine if the easyPayNumber/account number is valid for payment, to obtain the due payment amount, the minimum and maximum amounts that will be accepted for payment, and any other relevant information. It in no way implies that the payment has been accepted or that it will be accepted.

Collectors that connect to Easypay to collect Bill Payments are instructed to take the correct, minimum and maximum amounts that are returned in the info response into account. If the minimum and/or maximum values are present, then the POS should prevent amounts outside of these limits to be entered and accepted. If a correct amount is provided with minimum and/or maximum values as zero, or with the same value as the correct amount, then the POS should only allow the correct amount and not allow editing of the amount. If the correct, minimum and maximum values all have zero values, or are not provided in the response, then the POS should allow any amount.

Not all collectors implement these rules for various reasons, and some do not implement the info request but only do an authorization request with an advice request after. Receivers must validate the amount received in the Authorization request.

2.5.3 Authorisation Request

The authorisation request is used by EasyPay to ask for authorisation to accept the specified payment at the specific amount. It in no way implies that the payment has been accepted or that it will be accepted.

EasyPay will send all the information it has about the authorisation request to the Receiver and based on that information the Receiver can either authorize the payment or instruct EasyPay to decline the payment.

The response will indicate to EasyPay the state of the request (authorized or declined) and, provide EasyPay with any additional information that is available that EasyPay can provide to the POS that may help with processing future requests (for example: If an authorisation request was declined because the payment amount was not sufficient the response should indicate the correct amount so that the Customer could rectify this on the next request. This logic however depends on how the collector implemented it on the POS device.)

2.5.4 Notification

A notification is used by EasyPay to notify the receiver that the payment has already been accepted. The Easypay Bill Payment system will only send a notification once an advice message has been received from the collector for the associated authorized transaction.

EasyPay will send all the information it has about the completed payment to the Receiver in the notification message. The receiver can then log that payment has having been completed and match it to the SOF file that will be received from the settlement system the following day.

The notification request must return a notification response, but unlike the authorisation response the notification response does not allow the receiver to 'decline' the notification. The receiver is obliged to respond to each notification with a notification response. The content of this notification response is the echo data (from the notification request).

The Easypay Bill Payment server will try multiple times to send notifications if it does not get a response. Notification exceptions where no notification was received or where the receiver did not respond, or throw an exception will be handled as exceptional cases during reconciliation.

3. Reconciliation

Once a day EasyPay will send a SOF file to the receiver detailing all transactions that were collected and for which notifications were send and responses received over the previous day.

In theory this list of transactions should exactly match the notifications that were set during the previous day. However there are times when this will not be the case.

- Since The SOF file is produced at a fixed time it may not contain transactions that were processed after its creation. In this case the SOF file from the following day will contain these transactions.
- There might be exceptional cases where the authorization was performed in the previous day but where the advice message from the collector is only received the next day or not at all. For example, in the case where there are network or server related problems at the collector side.
- In the case where EasyPay is allowed to capture payments for the receiver at offline collectors that are not connected to the Easypay BPS in an online manner, there may be notifications that are never sent to the Receiver online, and that are only represented in the SOF file. (Note: we will not allow offline collectors to collect transactions where the receiver indicated that they always require online authorisation).
- When 2 or more payments are received for the same account number on the same day at the same merchant, the settlement system currently combines these payments into one single payment in the SOF file. However notifications for these payments will be sent as the transactions are advised. In this way 2 notifications might be represented as a single transaction in the SOF file.

Fields that should be used to match the online notifications to the SOF file transactions are as follows.

- Merchant ID
- Terminal ID
- Date
- Time
- EasyPay Number

- Amount
- Trace

In the case of a combined payment the terminal and time fields will match the first of the combined notifications and the amount will equal the combined value of the notifications.