PayGo Zimbabwe REST API Admin and Non-Functional Protocol Specification

Version v1.2.0, 17.05.2022

Contents

1. Introduction . . . 1
1.1. Audience . . . 1
1.2. Pre-Requisites . . . 1
2. Integration . . . 1
3. Authentication . . . 1
4. Endpoints . . . 2
4.1. URL’s . . . 2
4.2. Public . . . 2
4.3. Financial Institutions . . . 3
4.3.1. GET Financial Institutions . . . 3
4.4. Merchants . . . 4
4.4.1. GET Merchants . . . 5
4.4.2. Create Merchant . . . 6
4.4.3. Update Merchant . . . 7
4.4.4. Delete Merchant . . . 7
4.5. Authorised Credentials . . . 8
4.5.1. Get Authorised Credential . . . 9
4.5.2. Create Authorised Credential . . . 9
4.5.3. Update Authorised Credentials . . . 10
4.5.4. Delete Authorised Credential . . . 10
4.6. Merchant Credentials . . . 11
4.6.1. Get Merchant Credentials . . . 11
4.6.2. Create Merchant Credential . . . 12
4.6.3. Delete Merchant Credential . . . 12
5. Admin Actions . . . 12
5.1. Expire Payment Request . . . 12
5.2. Query Payment Status . . . 13

Table 1. Changelog

Version	Date	Author	Comments
Draft	16/08/2021	bgrobler	Initial
1.0.0	04/03/2022	bgrobler	Correct sandbox url, Correct merchant-credentials url, Fix requestPayment Example, Add public endpoint (Helpers)
1.0.1	27/04/2022	bgrobler	Correct expire url
1.1.0	23/01/2023	jpurse	Add get Merchant By MSP reference
1.2.0	08/04/2024	jpurse	update ZWL to ZiG

1. Introduction

PayGo Zimbabwe provides payment lookup and routing services.

1.1. Audience

The target audience for this document are Business Analysts and Software Engineers.

1.2. Pre-Requisites

A basic understanding of financial payments / electronics funds transfer terminology and concepts is expected.

An `API Key` provided by PayGo.

2. Integration

This integration follows REST CRUD Design Principles.

Each PayGo Transaction is a strict contract between PayGo and the Member, any non-compliant messages will be dropped. See the PayGo Manual for further details concerning compliance and SLA.

3. Authentication

refer to 'PayGo Zimbabwe REST API Protocol Specification' document.

4. Endpoints

4.1. URL’s

Table 2. Environments

Environment	URL
Production	https://api.paygo.co.zw
Sandbox	https://api.paygo.co.zw/sandbox

PayGo will provide a production key upon successful completion of the on-boarding process.

Table 3. Common HTTP Request Headers

Header	Type	Header Description
X-Api-Key	String	The API Key provided by PayGo

API Keys are unique to each environment

4.2. Public

These endpoints do not require authentication

Helper endpoints for the QR codes as they are not "easy" to make, one for a base 64 encoded image and the other for the actual image. these can be used as a backend call or directly on the browser side as an image source.

```
GET /admin/public/qr
```

returns the image file, suitable for html src files or downloading etc.

```
GET /admin/public/qr64
```

returns the image as a base64 encoded string.

Table 4. REQUIRED URL Parameters

Name	Description	Example
shortCode	PayGoID	40010
narration		bananas
merchantName		My Shop
address		23 Cookie Lane

Name	Description	Example
amount	Number (minor denomination ie. cents)	123

Examples

https://api.paygo.co.zw/sandbox/admin/public/qr?shortCode=40010&narration=bananas&merchantName=My%20Shop&address=23%20Cookie%20Lane&amount=123

https://api.paygo.co.zw/sandbox/admin/public/qr64?shortCode=40010&narration=bananas&merchantName=My%20Shop&address=23%20Cookie%20Lane&amount=123

usage in html ie.

```
<img
src="https://api.paygo.co.zw/sandbox/admin/public/qr?shortCode=40010&narration=bananas
&merchantName=My%20Shop&address=23%20Cookie%20Lane&amount=123">
```

4.3. Financial Institutions

Available Institutions to the PayGo Ecosystem.

Read only access to Financial Institutions is available for all connections.

Returned Financial Institutions are based on the ones allowed access to. if the one you are wanting is not listed. please contact your PayGo integration contact.

4.3.1. GET Financial Institutions

```
GET /admin/financial-institution
```

Table 5. FinancialInstitutionDTO

Field Name	Field Type	Field Description
id	String (UUID)	The entity id
created	OffsetDateTime	Time at which the entity was created
updated	OffsetDateTime	Time at which the entity was updated
active	boolean	Indicates whether this FI is available for use.
name	String	The human friendly name of the institutions

Field Name	Field Type	Field Description
bin	String	The BIN assigned to this institution.

Example Response 200 OK


```
[{
  "id": "ecf47c2d-12e7-49e3-8d0b-8ec0250cc6f1",
  "created": "2021-07-19T13:58:46.687Z",
  "updated": "2021-07-19T13:58:46.687Z",
  "active": true,
  "name": "Acme Bank",
  "bin": "123456"
},
{
  "id": "a5cb5bf3-41d8-42b0-940f-44fe70919fd3",
  "created": "2021-07-19T13:58:46.687Z",
  "updated": "2021-07-19T13:58:46.687Z",
  "active": true,
  "name": "Big Bank",
  "bin": "654321"
}]
```


4.4. Merchants

Merchants are created to be linked to relevant Authorised credentials.

Table 6. MerchantDTO

Field Name	Field Type	Field Description
id	String (UUID)	The entity id
created	OffsetDateTime	Time at which the entity was created
updated	OffsetDateTime	Time at which the entity was updated
active	Boolean	Indicates whether this credential is available for use.
name	String	Indicates the merchant Name
addressLine1	String	address Line 1
addressLine2	String	address Line 2
region	String	Region location of the merchant
city	String	City
countryCode	String AN.3	Country Code eg ZWE
description	String	Description

Field Name	Field Type	Field Description
emailAddress	String	Email Address
mspReference	String	an MSP held reference that can be used to retrieve the merchant
phoneNumber	String	Phone Number
url	String	Merchant website URL (Optional)

4.4.1. GET Merchants

```
GET /admin/merchant
```

returns a paged list, maximum 100 results. optional param "page" defaults to 0

```
GET /admin/merchant?page=2
```

Get specific Merchant

```
GET /admin/merchant/{id}
```

Get specific Merchant by MSP Reference

```
GET /admin/merchant/ref/{reference}
```

Example Response 200 OK


```
[
    {
        "id": "795bf316-cfc3-48fd-8c20-4efed2093722",
        "name": "The Rainy Hammer",
        "active": true,
        "addressLine1": "shop 12",
        "addressLine2": "rainy lane",
        "city": "Harare",
        "countryCode": "ZWE",
        "created": "2021-07-31T21:46:55.225+02:00",
        "description": "merchant desc",
        "emailAddress": "info@rainy.org",
        "mspReference": "custom_ref_2293f",
        "phoneNumber": "+263773123123",
        "region": "Mashonaland",
        "updated": "2021-07-31T21:46:59.615+02:00",
```

```
        "url": "http://rainy.test.org"
    }
]
```

4.4.2. Create Merchant

```
POST /admin/merchant
```

Example Request


```
{
    "id": null,
    "name": "The Rainy Hammer",
    "active": true,
    "addressLine1": "shop 12",
    "addressLine2": "rainy lane",
    "city": "Harare",
    "countryCode": "ZWE",
    "created": "2021-07-31T21:46:55.225+02:00",
    "description": "merchant desc",
    "emailAddress": "info@rainy.org",
    "mspReference": "custom_ref_2293f",
    "phoneNumber": "+263773123123",
    "region": "Mashonaland",
    "updated": "2021-07-31T21:46:59.615+02:00",
    "url": "http://rainy.test.org"
}
```


Example Response 200 OK


```
{
    "id": "795bf316-cfc3-48fd-8c20-4efed2093722",
    "name": "The Rainy Hammer",
    "active": true,
    "addressLine1": "shop 12",
    "addressLine2": "rainy lane",
    "city": "Harare",
    "countryCode": "ZWE",
    "created": "2021-07-31T21:46:55.225+02:00",
    "description": "merchant desc",
    "emailAddress": "info@rainy.org",
    "mspReference": "custom_ref_2293f",
    "phoneNumber": "+263773123123",
    "region": "Mashonaland",
    "updated": "2021-07-31T21:46:59.615+02:00",
    "url": "http://rainy.test.org"
}
```

4.4.3. Update Merchant

```
PUT /admin/merchant
```

Example Request


```
{
    "id": "795bf316-cfc3-48fd-8c20-4efed2093722",
    "name": "The Rainy Hammer",
    "active": true,
    "addressLine1": "shop 12",
    "addressLine2": "rainy lane",
    "city": "Harare",
    "countryCode": "ZWE",
    "created": "2021-07-31T21:46:55.225+02:00",
    "description": "merchant desc 2",
    "emailAddress": "info@rainy.org",
    "mspReference": "custom_ref_2293f",
    "phoneNumber": "+263773123123",
    "region": "Mashonaland",
    "updated": "2021-07-31T21:46:59.615+02:00",
    "url": "http://rainy.test.org"
}
```


Example Response 200 OK


```
{
    "id": "795bf316-cfc3-48fd-8c20-4efed2093722",
    "name": "The Rainy Hammer",
    "active": true,
    "addressLine1": "shop 12",
    "addressLine2": "rainy lane",
    "city": "Harare",
    "countryCode": "ZWE",
    "created": "2021-07-31T21:46:55.225+02:00",
    "description": "merchant desc 2",
    "emailAddress": "info@rainy.org",
    "mspReference": "custom_ref_2293f",
    "phoneNumber": "+263773123123",
    "region": "Mashonaland",
    "updated": "2021-07-31T21:46:59.615+02:00",
    "url": "http://rainy.test.org"
}
```


4.4.4. Delete Merchant

```
DELETE /admin/merchant/{id}
```

```
200 OK
```

4.5. Authorised Credentials

Each entity that wants to receive funds must register their credentials, these are added to an authorized list of credentials. Without this, any `NominatedCredential` that does not match the authorised credentials will be dropped, and a security violation raised, *the transaction will be aborted*).

Access to the authorized credentials will be based on the specified API Key which identifies this call.

Table 7. AuthorisedCredentialDTO

Field Name	Field Type	Field Description
id	String (UUID)	The entity id
created	OffsetDateTime	Time at which the entity was created
updated	OffsetDateTime	Time at which the entity was updated
active	Boolean	Indicates whether this credential is available for use.
type	String	Indicates the type of credential, e.g. ACCEPTOR, BANK_ACCOUNT, etc
credential	String	The primary identifier for a credential, e.g. Account Id
credentialReference	String	A reference provided by the calling system, e.g. a UUID
terminalId	String ANS.8	The terminal identifier at the acquiring bank (identifying the merchant account)
cardAcceptorId	String ANS.15	The card acceptor identified at the acquiring bank (identifying the merchant account)
currencyCode	String N.3 (alpha)	The currency code that the merchant can be settled in. (this may be null for multi-currency conversions)
fiId	String (UUID)	The financial institution correlated to this credential.

4.5.1. Get Authorised Credential

```
GET /admin/authorised-credential
```

returns a paged list, maximum 100 results. optional param "page" defaults to 0

```
GET /admin/authorised-credential?page=2
```

Get specific Authorised Credential

```
GET /admin/authorised-credential/{id}
```

4.5.2. Create Authorised Credential

```
POST /admin/authorised-credential
```

Example Request

```
{
  "id": null,
  "created": null,
  "updated": null,
  "active": true,
  "type": "ACCEPTOR",
  "credential": null,
  "credentialReference": "83d33750-553f-490c-94ec-647d39ee6b11",
  "terminalId": "12345678",
  "cardAcceptorId": "123456789012345",
  "currencyCode": "ZiG",
  "fiId": "87fb2f40-5b0f-4b46-b45a-ac12dcc94edd"
}
```

Example Response 200 OK

```
{
  "id": "ecf47c2d-12e7-49e3-8d0b-8ec0250cc6f1",
  "created": "2021-07-19T13:58:46.687Z",
  "updated": "2021-07-19T13:58:46.687Z",
  "active": true,
  "type": "ACCEPTOR",
  "credential": null,
  "credentialReference": "83d33750-553f-490c-94ec-647d39ee6b11",
  "terminalId": "12345678",
  "cardAcceptorId": "123456789012345",
  "currencyCode": "ZiG",
```

```
  "fiId": "87fb2f40-5b0f-4b46-b45a-ac12dcc94edd"
}
```

4.5.3. Update Authorised Credentials

```
PUT /admin/authorised-credential
```

Example Request


```
{
  "id": "ecf47c2d-12e7-49e3-8d0b-8ec0250cc6f1",
  "created": "2021-07-19T13:58:46.687Z",
  "updated": "2021-07-19T13:58:46.687Z",
  "active": false,
  "type": "ACCEPTOR",
  "credential": null,
  "credentialReference": "83d33750-553f-490c-94ec-647d39ee6b11",
  "terminalId": "12345678",
  "cardAcceptorId": "123456789012345",
  "currencyCode": "ZiG",
  "fiId": "87fb2f40-5b0f-4b46-b45a-ac12dcc94edd"
}
```


Example Response 200 OK


```
{
  "id": "ecf47c2d-12e7-49e3-8d0b-8ec0250cc6f1",
  "created": "2021-07-19T13:58:46.687Z",
  "updated": "2021-07-19T13:59:32.143Z",
  "active": false,
  "type": "ACCEPTOR",
  "credential": null,
  "credentialReference": "83d33750-553f-490c-94ec-647d39ee6b11",
  "terminalId": "12345678",
  "cardAcceptorId": "123456789012345",
  "currencyCode": "ZiG",
  "fiId": "87fb2f40-5b0f-4b46-b45a-ac12dcc94edd"
}
```


4.5.4. Delete Authorised Credential

When an authorised credential is deleted it will also delete any "merchant credential" links between merchants and this credential.

```
DELETE /admin/authorised-credential/{id}
```

```
200 OK
```

4.6. Merchant Credentials

Each Merchant must be associated with the authorised credentials that can be used.

Table 8. MerchantCredentialsDTO

Field Name	Field Type	Field Description
id	String (UUID)	The entity id
created	OffsetDateTime	Time at which the entity was created
updated	OffsetDateTime	Time at which the entity was updated
credentialId	String (UUID)	the ID of the authorised credential
merchantId	String (UUID)	the merchant ID

4.6.1. Get Merchant Credentials

Returns a list merchant credentials associated with a given merchant id

```
GET /admin/merchant-credentials/merchant/{merchant id}
```

Returns a list merchant credentials associated with a given credential id

```
GET /admin/merchant-credentials/credential/{credential id}
```

Example Request

```
GET /admin/merchant-credentials/merchant/795bf316-cfc3-48fd-8c20-4efed2093722
```

Example Response 200 OK


```
[
    {
        "id": "3479e2ac-f7d4-47a5-a9b7-ee203f400694",
        "created": "2021-08-13T12:01:16.534+02:00",
        "updated": "2021-08-13T12:01:15.219+02:00",
        "credentialId": "260949fc-b943-440b-848d-e74f85e17c2c",
        "merchantId": "795bf316-cfc3-48fd-8c20-4efed2093722"
    }
]
```

4.6.2. Create Merchant Credential

```
POST /admin/merchant-credentials
```

Example Request


```
{
    "id": null,
    "created": null,
    "updated": null,
    "credentialId": "260949fc-b943-440b-848d-e74f85e17c2c",
    "merchantId": "795bf316-cfc3-48fd-8c20-4efed2093722"
}
```


Example Response 200 OK


```
{
    "id": "60ff59ec-dc80-41c5-a72b-1ebb93bd379f",
    "created": "2021-08-13T10:06:18.343262+02:00",
    "updated": "2021-08-13T10:06:18.343265+02:00",
    "credentialId": "260949fc-b943-440b-848d-e74f85e17c2c",
    "merchantId": "795bf316-cfc3-48fd-8c20-4efed2093722"
}
```


4.6.3. Delete Merchant Credential

```
DELETE /admin/merchant-credentials/{id}
```

```
200 OK
```

5. Admin Actions

5.1. Expire Payment Request

Due to the Async nature of PayGo Payments, if an MSP has received an AUTH for a transaction there is no way to cleanly tell if and when a payment might be processed. As such, any endpoint device should request a premature transaction expiry, only on response from PayGo can the endpoint device cancel the transaction

```
GET /admin/action/expire/{deviceReference}
```

Table 9. ExpireRspDto

Field Name	Field Type	Field Description
created	OffsetDateTime	Time at which the entity was created
responseCode	String	Refer to below table
responseDescription	String	Textual description of the response

Table 10. Response Codes

Response Code	Description
000	Payment Expired
005	Do not honour (Transaction Already Processed - Request received too late). perform query on transaction state if no advice received.
025	Not Found
091	Transaction possibly in flight, unknown result. try again after a few seconds.

Response


```
{
  "created": "2021-06-16T10:24:43.594+02:00",
  "responseCode": "000",
  "responseDescription": "Payment Expired"
}
```


5.2. Query Payment Status

```
GET /admin/query/{deviceReference}
```

Table 11. MerchantPaymentDto

Field Name	Field Type	Field Description
merchantPaymentId	String	Device Reference as specified in the first auth response
amount	Double	The amount in decimal (2 decimal place precision)
complete	boolean	true/false
success	boolean	null/true/false. null= payment state not final

Field Name	Field Type	Field Description
created	OffsetDateTime	Time at which the entity was created
updated	OffsetDateTime	Time at which the entity was updated
currencyCode	String	currency code as specified
expiresAt	OffsetDateTime	Time at which this payment expires based on first auth response expires seconds
initiator	String	initiator as specified in first auth response
narration	String	narration as specified in first auth response
reference	String	reference as specified in first auth response
token	String	token as specified in first auth response
token	String	token as specified in first auth response
advices	List<BridgeTransactionDto>	refer to the bridge api, these are the same advices as would be received through the async push