Wallet Guru

Technical Documentation

Annex A

Documentation

API’s, Endpoints and Webhooks

Version 1.2

CONTENTS

DOCUMENT REVISIONS

Version	Date	Author	Description of Changes
1.0	2024-12-16	Myriam A. Otermin	Initial Draft created by ErgonStream Servicess LLC
1.1	2025-02-06	Myriam A. Otermin	Updates document formats
1.2	2025-02-24	Emilio Otermin	Update Section 1: About the API's

INTRODUCTION

This document provides a comprehensive overview of the technology stack implemented in **Phase 1** of the Wallet Guru project. It serves as a detailed reference for developers, system architects, and integration partners, outlining the structure, security mechanisms, and key API interactions that define the Wallet Guru ecosystem.

The document covers all critical aspects of the system's API endpoints and webhooks, ensuring a clear understanding of how external applications and services can interact with Wallet Guru. Specifically, it includes:

- **Environment Syntax** – Establishes the foundational syntax and conventions used across the system.
- **Security Schemas** – Defines the security protocols in place to protect data integrity and ensure secure communication.
- **Authentication Schemas** – Details the authentication mechanisms employed to verify and manage user access.
- **Wallet Schemas** – Describes the structure and functionality of wallet-related operations within the system.
- **Code Schemas** – Outlines the syntax and format of system-generated codes used across different modules.

Additionally, this document provides in-depth specifications for API endpoints, detailing the various interactions available:

- **Authentication Endpoints** – Covers all authentication-related API calls, including user login, token management, and access control.
- **Wallet Endpoints** – Defines the APIs responsible for managing wallets, transactions, balances, and related functions.
- **Code Endpoints** – Details the mechanisms for generating, validating, and utilizing codes within the Wallet Guru ecosystem.
- **Countries, States, and Cities API** – Provides a structured approach to accessing geographical data for seamless integration into the platform.

This documentation is designed to serve as a reliable guide for implementation, troubleshooting, and future development phases. By following the outlined schemas and API specifications, developers and stakeholders can ensure smooth integration and operational consistency within the Wallet Guru system.

ABOUT THE API

Environment Syntax

Environment	Syntax
Local environment	http://localhost:3000/api/v1/...
Dev environment	https://dev.auth.walletguru.co/ api/v1/...
QA Environment	https://qa.auth.walletguru.co/api/v1/...
Staging environment	https://stg.auth.walletguru.co/api/v1/...
Production environment	https://auth.walletguru.co/api/v1/...

Security Schemas

These apply to all three groups of endpoints: Authentication, Wallet, and Codes

JWT	
scheme	bearer
bearerFormat	JWT
type	http
in	header

Schemas - Authentication

CreateUserDto	
Type	Object

Properties	"id": { "type": "string" },
	"firstName": { "type": "string" },
	"lastName": { "type": "string" },
	"email": { "type": "string" },
	"phone": { "type": "string" },
	"passwordHash": { "type": "string" },
	"mfaEnabled": { "type": "boolean" },
	"mfaType": {
	"enum": ["SMS", "TOTP"],
	"type": "string"}
	"type": {
	"enum": ["PLATFORM", "PROVIDER", "WALLET"],
	"type": "string"}
	“roleId": { "type": "string" },
	"active": { "type": "boolean" },
	"state": {
	"enum": [0, 1, 2, 3, 4, 5],
	"type": "number"},
	"picture": { "type": "string" },

	"sendSms": { "type": "boolean" },
	"sendEmails": { "type": "boolean" },
	"serviceProviderId": { "type": "string" },
	"lastLogin": {
	"format": "date-time",
	"type": "string"},
	"termsConditions": { "type": "boolean" },
	"privacyPolicy": { "type": "boolean" },
	"socialSecurityNumber": { "type": "string" },
	"identificationType": { "type": "string" },
	"identificationNumber": { "type": "string" },
	"country": { "type": "string" },
	"stateLocation": { "type": "string" },
	"city": { "type": "string" },
	"zipCode": { "type": "string" },
	"address": { "type": "string" },
	"dateOfBirth": {
	"format": "date-time",
	"type": "string"},
	"avatar": { "type": "string" },
	"imageUrl": { "type": "string" },
	"ContactUser": { "type": "boolean" }

Required	"id", "firstName", "lastName", "email", "phone", "passwordHash", "mfaEnabled", "mfaType", "type", "roleId", "active", "state", "picture", "sendSms", "sendEmails", "serviceProviderId", "lastLogin", "termsConditions", "privacyPolicy", "imageUrl", "ContactUser"

UpdateUserDto	
Type	Object
Properties	None
Required	None

VerifyOtpDto	
Type	Object
Properties	"email": { "type": "string" }, "otp": { "type": "string", "minLength": 6, "maxLength": 6 }
Required	"email", "otp"

SignInDto	
Type	Object
Properties	"email": { "type": "string" }, "password": { "type": "string" }

Required	"email", "password"

AuthChangePasswordUserDto	
Type	Object
Properties	"token": { "type": "string" }, "currentPassword": { "type": "string" }, "newPassword": { "type": "string" }
Required	"token", "currentPassword", "newPassword"

AuthForgotPasswordUserDto	
Type	Object
Properties	"email": { "type": "string" }
Required	“email”

AuthConfirmPasswordUserDto	
Type	Object
Properties	"email": { "type": "string" }, "confirmationCode": { "type": "string" }, "newPassword": { "type": "string", "pattern": "/^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[$&+,:;=?@#\|'<>.^*()%!-])[A-Za-z\\d@$&+,:;=?@#|'<>.^*()%!-]{8,}$/"}
Required	"email", "confirmationCode", "newPassword"

UpdateStatusUserDto	
Type	Object
Properties	"type": "object", "properties": { "email": { "type": "string" }, “active": { "type": "boolean" }},
Required	"email", "active"

SendOtpDto	
Type	Object
Properties	"email": { "type": "string" }
Required	“e-mail”

ValidateAccessDto	
Type	Object
Properties	"token": { "type": "string" }, "path": { "type": "string" }, "method": { "type": "string" }

Required	"token", "path", "method"

CreateRoleDto	
Type	Object

Properties	"name": {
	"type": "string",
	"minLength": 1,
	"maxLength": 20,
	"pattern": "/^[\\p{L}\\s´\\-_]+$/u"},
	"description": {
	"type": "string",
	"minLength": 0,
	"maxLength": 50,
	"pattern": "/^[\\p{L}\\s´\\-_]+$/u"},
	"providerId": {
	"type": "object",
	"default": "EMPTY"}

Required	"name", "description"

UpdateRoleDto	
Type	Object
Properties	None
Required	None

UpdateSettingsDto	
Type	Object
Properties	"value": { "type": "string" }
Required	None

MuteNotificationsDto	
Type	Object

Properties	"duration": {
	"type": "string",
	"description": "Duration to mute notifications",
	"example": "30m"}

Required	“duration”

Schemas – Wallet

CreateWalletDto	
Type	Object
Properties	"name": { "type": "string" }, "walletType": { "type": "string" }, "walletAddress": { "type": "string" }

Required	"name", "walletType", "walletAddress"

UpdateWalletDto	
Type	Object
Properties	"id": { "type": "string" }, "name": { "type": "string" }, "walletType": { "type": "string" }, "walletAddress": { "type": "string" }
Required	“id", "name", "walletType", "walletAddress"

GetWalletDto	
Type	Object
Properties	"id": { "type": "string" }, "name": { "type": "string" }, "walletType": { "type": "string" }, "walletAddress": { "type": "string" }, "active": { "type": "boolean" }, "search": { "type": "string" }, "items": { "type": "number" }, "page": { "type": "number" }
Required	None

CreateRefundsDto	
Type	Object
Properties	"Id": { "type": "string" }, "amount": { "type": "number" }, "description": { "type": "string" }, "activityId": { "type": "string" }, "serviceProviderId": { "type": "string" }, "walletAddress": { "type": "string" }
Required	"Id", "amount", "description", "activityId", "serviceProviderId", "walletAddress"

CreateRafikiWalletAddressDto	
Type	Object
Properties	"addressName": { "type": "string", "minLength": 3, "maxLength": 20, "pattern": "/^(?=.*[a-z])(?=.*\\d)[a-z\\d]+$/"}, "assetId": { "type": "string" }
Required	"addressName", "assetId"

ReceiverInputDTO

Type	Object
Properties	"amount": { "type": "number" }, "walletAddressUrl": { "type": "string" }, "walletAddressId": { "type": "string" }
Required	"amount", "walletAddressUrl"

LinkInputDTO	
Type	Object
Properties	"walletAddressUrl": { "type": "string" }, "walletAddressId": { "type": "string" }, "sessionId": { "type": "string" }
Required	"walletAddressUrl"

UnLinkInputDTO	
Type	Object
Properties	"sessionId": { "type": "string" }
Required	None

DepositDTO	
Type	Object
Properties	"walletAddressId": { "type": "string" }, "amount": { "type": "number" }
Required	"walletAddressId", "amount"

CreatePaymentDTO	
Type	Object
Properties	"incomingAmount": { "type": "number" }, "walletAddressUrl": { "type": "string" }, "walletAddressId": { "type": "string" }
Required	"incomingAmount", "walletAddressUrl", "walletAddressId"

ActionOugoingPaymentDto	
Type	Object
Properties	"outgoingPaymentId": { "type": "string" }, "action": { "type": "string" }
Required	"outgoingPaymentId", "action"

EventWebHookDTO	
Type	Object
Properties	"id": { "type": "string" }, "type": { "type": "string" }, "data": { "type": "object" }

Required	"id", "type", "data"

ConfirmClearPaymentDto	
Type	Object
Properties	"clearPaymentId": { "type": "string" }, "observations": { "type": "string" }, "referenceNumber": { "type": "string" }
Required	"clearPaymentId", "observations", "referenceNumber"

Schemas – Codes

None

AUTHENTICATION ENDPOINTS

Comprehensive documentation for the Wallet Guru API, detailing the Authentication service and its endpoints.

user

UserController_create

Syntax

POST: /api/v1/users/register

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

CreateUserDto, p. 9

Responses

Code	Description
201	The record has been successfully created
403	Forbidden

Example

```
{
  "id": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "phone": "string",
  "passwordHash": "string",
  "mfaEnabled": true,
  "mfaType": "SMS",
  "type": "PLATFORM",
  "roleId": "string",
  "active": true,
  "state": 0,
  "picture": "string",
  "sendSms": true,
  "sendEmails": true,
  "serviceProviderId": "string",
  "lastLogin": "2025-01-17T16:51:51.856Z",
  "termsConditions": true,
```

```json
  "privacyPolicy": true,
  "socialSecurityNumber": "string",
  "identificationType": "string",
  "identificationNumber": "string",
  "country": "string",
  "stateLocation": "string",
  "city": "string",
  "zipCode": "string",
  "address": "string",
  "dateOfBirth": "2025-01-17T16:51:51.856Z",
  "avatar": "string",
  "imageUrl": "string",
  "ContactUser": true
}
```

UserController_createApp

Syntax

POST: api/v1/users/register-app

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

CreateUserDto, p. 9

Responses

Code	Description
201	The record has been successfully created
403	Forbidden

Example

```
{
  "id": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "phone": "string",
  "passwordHash": "string",
  "mfaEnabled": true,
  "mfaType": "SMS",
  "type": "PLATFORM",
  "roleId": "string",
  "active": true,
  "state": 0,
  "picture": "string",
  "sendSms": true,
  "sendEmails": true,
  "serviceProviderId": "string",
  "lastLogin": "2019-08-24T14:15:22Z",
  "termsConditions": true,
  "privacyPolicy": true,
  "socialSecurityNumber": "string",
  "identificationType": "string",
  "identificationNumber": "string",
  "country": "string",
  "stateLocation": "string",
  "city": "string",
  "zipCode": "string",
  "address": "string",
  "dateOfBirth": "2019-08-24T14:15:22Z",
  "avatar": "string",
  "imageUrl": "string",
  "ContactUser": true
}

```

UserController_updateAdmin

Syntax

PUT: /api/v1/users/update-profile/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
Id	required	id	string

Schema

UpdateUserDto, p. 10

Responses

Code	Description
200	The record has been successfully created
403	Forbidden

Example

```
{}
```

UserController_getUserInfo

Syntax

GET: /api/v1/users/current-user

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
200	Successfully returned user info
403	Invalid access token

Example

Not Applicable

UserController_verifySignUp

Syntax

POST: /api/v1/users/verify/register

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

VerifyOtpDto, p. 10.

Responses

Code	Description
200	The user has been successfully verified
201	{}
403	Forbidden

Request Example

```
{
  "email": "string",
  "otp": "string"
}
```

UserController_findOne

Syntax

GET: /api/v1/users/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	required	id	String

Schema

None

Responses

Code	Description
200	The record has been successfully retrieved
403	Forbidden

Example

Not Applicable

UserController_update

Syntax

PUT: /api/v1/users/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	required	id	String

Schema

UpdateUsersDto, p. 10

Responses

Code	Description
200	The record has been successfully updated
403	Forbidden

Example

```
{}
```

UserController_remove

Syntax

DELETE: /api/v1/users/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	required	id	String

Schema

None

Responses

Code	Description
200	The record has been successfully deleted.
403	Forbidden.

Example

None

Toggle the first field of a user

Syntax

PUT: Toggle the first field of a user

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	id	String

Schema

None

Responses

Code	Description
200	User first field toggled successfully
404	User not found

Example

None

UserController_signin

Syntax

POST: /api/v1/users/signin

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

SignInDto, p. 10

Responses

Code	Description
200	The user has been successfully signed in.
201	`{ }`
403	Forbidden.

Example

```
{
  "email": "string",
  "password": "string"
}

```

UserController_verifyOtp

Syntax

POST: /api/v1/users/verify/otp/mfa

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

VerifyOtpDto, p. 10

Responses

Code	Description
200	The user has been successfully signed in.
201	`{ }`
403	Forbidden.

Example

```
{
  "email": "string",
  "otp": "string"
}

```

UserController_changePassword

Syntax

POST: /api/v1/users/change-password

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

AuthChangePasswordUserDto, p. 11

Responses

Code	**Description**
200	The password has been successfully changed.
201	{ }
403	Forbidden.

Example

```
{
  "token": "string",
  "currentPassword": "string",
  "newPassword": "string"
}
```

UserController_forgotPassword

Syntax

POST: /api/v1/users/forgot-password

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

AuthForgotPasswordUserDto, p. 11

Responses

Code	**Description**
200	The password reset request has been successfully processed.
201	`{ }`
403	Forbidden.

Example

```
{
  "email": "string"
}
```

UserController_confirmPassword

Syntax

POST: /api/v1/users/confirm-password

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

AuthConfirmPasswordUserDto, p. 11

Responses

Code	**Description**
200	The password has been successfully confirmed.
201	`{ }`
403	Forbidden.

Example

```
{
  "email": "string",
  "confirmationCode": "string",
  "newPassword": "/^(?=a)(?=A)(?=\\d)(?=$)A{8,}$/"
}
```

UserController_getUsers

Syntax

GET: /api/v1/users

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
type	Optional	type	String
search	Optional	search	String
firstName	Optional	firstName	String
lastName	Optional	lastName	String
serviceProviderId	Optional	serviceProviderId	String
email	Optional	email	String
id	Optional	id	String
items	Optional	items	Number
page	Optional	page	Number
orderBy	Optional	orderBy	String
ascending	Optional	ascending	Boolean
state	Optional	state	Number
wallet	Optional	wallet	String

Schema

None

Responses

Code	Description
200	Successfully returned users
403	Forbidden.

Example

None

UserController_changeStatusUser

Syntax

PATCH: /api/v1/users/update-status/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

UpdateStatusUserDto, p. 11

Responses

Code	Description
200	The user has been successfully updated.
403	Forbidden.

Example

```json
{
  "email": "string",
  "active": true
}
```

UserController_sendOtpEmail

Syntax

POST: /api/v1/users/send-otp

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

SendOtpDto, p. 11

Responses

Code	Description
200	The OTP email has been sent successfully.
201	`{ }`
403	Forbidden.

Example

```
{
  "email": "string"
}
```

UserController_revokeTokenLogout

Syntax

POST: /api/v1/users/logout

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
200	Logout successfully.
201	{ }
403	Forbidden.

Example

None

UserController_validateAccess

Syntax

POST: UserController_validateAccess

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

ValidateAccessDto, p. 11

Responses

Code	**Description**
200	Validate access successfully.
201	`{ }`
403	Forbidden.

Example

```
{
  "token": "string",
  "path": "string",
  "method": "string"
}
```

Update user profile image

Syntax

PUT: /api/v1/users/upload-image/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	ID of the provider	String

Schema

None

Responses

Code	Description
200	Provider updated successfully.
500	Error updating provider.

Example

None

Toggle the first field of a user

Syntax

PATCH: /api/v1/users/{id}/toggle-contact

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
Id	Required	ID of the user	String

Schema

None

Responses

Code	Description
200	User first field toggled successfully.
404	User not found.

Example

None

UserController_refreshToken

Syntax

POST: /api/v1/users/refresh-token

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	Token has been refreshed succefully.
201	`{ }`
403	Forbidden.

Example

```
{
  "token": "",
  "email": ""
}
```

UserController_kyc

Syntax

POST: /api/v1/users/kyc

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
200	verify kyc.
201	`{ }`
403	Forbidden.

Request Example

None

Reset password of user

Syntax

PATCH: /api/v1/users/reset-password/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	ID of the User	String

Schema

None

Responses

Code	Description
200	User password updated successfully.
500	Error updating user password.

Example

None

role

RoleController_create

Syntax

POST: /api/v1/roles

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

CreateRoleDto, p. 12

Responses

Code	Description
201	The role has been successfully created.
403	Forbidden.

Example

```
{
  "name": "/^p$/u",
  "description": "/^p$/u",
  "providerId": "EMPTY"
}
```

RoleController_findAllPaginated

Syntax

GET: /api/v1/roles

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
providerId	Optional	providerId	String
items	Optional	items	String
page	Optional	page	String
orderBy	Optional	orderBy	String
ascending	Optional	ascending	Boolean

Schema

None

Responses

Code	Description
200	Roles have been successfully retrieved.
403	Forbidden.

Example

None

RoleController_findAllActive

Syntax

GET: /api/v1/roles/active

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
provierID	Required	Provider ID	String

Schema

None

Responses

Code	Description
200	Active roles have been successfully retrieved.
403	Forbidden.

Example

None

RoleController_getRoleInfo

Syntax

GET: /api/v1/roles/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	id	String

Schema

None

Responses

Code	Description
200	Role have been successfully retrieved.
403	Forbidden.

Example

None

RoleController_update

Syntax

PUT: /api/v1/roles/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	required	Id of the role	string

Schema

UpdateRoleDto, p. 12

Responses

Code	Description
200	The role has been successfully updated.
403	Forbidden.

Example

```
{}
```

Toggle the active status of a role

Syntax

PATCH: /api/v1/roles/{id}/toggle

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
Id	Required	ID of the role	Sting

Schema

None

Responses

Code	Description
200	Role status toggled successfully.
404	Role not found.

Example

None

Create or update a new access level for a module

Syntax

POST: /api/v1/roles/access-level/{roleId}/{moduleId}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
roleID	Required	ID of the role	String
moduleID	Required	ID of the module	String

Schema

Any

Responses

Code	Description
201	Access level created successfully.
404	Role or Module not found.

Example

```
{
  "accessLevel": 11
}
```

Create or update a new access level for a module for service providers

Syntax

POST: /api/v1/roles/module-access-level/{roleId}/{moduleId}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
roleID	Required	ID of the role	String
moduleID	Required	ID of the module	String

Schema

See example below

Responses

Code	Description
201	Access level created successfully.
404	Role or Module not found.

Example

```
{
  "accessLevel": 11,
  "serviceProvider": "provider1"
}
```

Create or update a new access level

Syntax

POST: /api/v1/roles/general-access-level/{roleId}/{moduleId}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
roleID	Required	ID of the role	String
moduleID	Required	ID of the module	String

Schema

See example below

Responses

Code	Description
201	Access level created successfully.
404	Role or Module not found.

Request Example

```
{
  "accessLevel": 11,
  "serviceProvider": "provider1"
}
```

List access levels for a role by modules

Syntax

GET: /api/v1/roles/simple-access-level/{roleId}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
roleId	Required	ID of the role	String

Schema

None

Responses

Code	Description
200	List of levels of access obtained successfully.
403	Role not found

Example

None

List access levels for a role with service providers

Syntax

GET: /api/v1/roles/access-level/{roleId}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
roleId	Required	ID of the role	String

Schema

None

Responses

Code	Description
200	List of levels of access obtained successfully.
404	Role not found

Example

None

Retrieve roles from the Provider ID (optional)

Syntax

GET: /api/v1/roles/list/providers/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
providerId	Optional	providerId	String
items	Optional	items	String
page	Optional	page	String
orderBy	Optional	orderBy	String
ascending	Optional	ascending	Boolean
id	Required	id	String

Schema

None

Responses

Code	Description
200	Provider found.
404	Provider not found.

Example

None

provider

Create a new provider

Syntax

POST: /api/v1/providers

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

The provider creation data; see example below

Responses

<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>201</td><td><pre>{
  "statusCode": 201,
  "customCode": "WGS0077",
  "data": {
    "Name": "Provider Name",
    "Description": "Provider description",
    "Email": "provider@example.com",
    "Phone": "+123456789",
    "EINNumber": "12-3456789",
    "Country": "Argentina",
    "City": "Balvanera",
    "ZipCode": "59569",
    "CompanyAddress": "Company Address test",
    "WalletAddress": "test wallet address",
    "ContactInformation": "Contact Info test",
    "CreateDate": 1725592379481,
    "UpdateDate": 1725592379481,
    "Id": "4fef9ad1-1dba-4e65-976a-1bea78043e66"
  }
}</pre></td></tr>
<tr><td>500</td><td>Error creating provider.</td></tr>
</table>

Examples

```
{
  "name": "Provider Name",
  "description": "Provider Description",
  "email": "provider@example.com",
  "phone": "+123456789",
  "einNumber": "12-3456789",
  "country": "Argentina",
  "city": "Balvanera",
  "zipCode": "59569",
  "companyAddress": "Company Address test",
  "walletAddress": "",
  "contactInformation": "Contact Info test"
}
```

Retrieve a list of providers with pagination and optional search.

Syntax

GET: /api/v1/providers

Description

Authorization Schema: JWT
Response body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
Page	Optional	Page number	Number
Items	Optional	Number of providers per page	Number
Search	Optional	Filter by provider name	String

Schema

None

Responses

<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>200</td><td>

```
{
  "statusCode": 200,
  "customCode": "WGE0135",
  "data": {
    "providers": [
      {
        "id": "123",
        "name": "Provider Name",
        "description": "Provider Description",
        "email": "provider@example.com",
        "phone": "+123456789",
        "einNumber": "12-3456789",
        "country": "Argentina",
        "city": "Balvanera",
        "zipCode": "59569",
        "companyAddress": "Company Address",
        "walletAddress": "",
        "logo": "https://example.com/logo.png",
        "contactInformation": "Contact Info"
      }
    ],
    "currentPage": 1,
    "total": 1,
    "totalPages": 1
  }
}
```

</td></tr>
<tr><td>403</td><td>Access forbidden.</td></tr>
</table>

Example

None

Retrieve a single provider by ID

Syntax

GET: /api/v1/providers/{id}

Description

Authorization Schema: JWT
Response body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
Id	Required	ID of the provider	String

Schema

None

Responses

Code	Description

200	{
	  "statusCode": 200,
	  "customCode": "WGE0074",
	  "data": {
	    "id": "123",
	    "name": "Provider Name",
	    "description": "Provider Description",
	    "email": "provider@example.com",
	    "phone": "+123456789",
	    "einNumber": "12-3456789",
	    "country": "Argentina",
	    "city": "Balvanera",
	    "zipCode": "59569",
	    "companyAddress": "Company Address",
	    "walletAddress": "",
	    "logo": "https://example.com/logo.png",
	    "contactInformation": "Contact Info"
	  }
	}

404	Provider not found.

Example

None

Update an existing provider

Syntax

PUT: /api/v1/providers/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	ID of the provider	String

Schema

Data to update the provider; see example below.

Responses

Code	Description
200	Provider updated successfully.
403	Error updating provider.

Example

```
{
  "name": "Updated Provider Name",
  "description": "Updated Description",
  "email": "updated@example.com",
  "phone": "+123456789",
  "einNumber": "12-3456789",
  "country": "Argentina",
  "city": "Balvanera",
  "zipCode": "59569",
  "companyAddress": "Updated Address",
  "walletAddress": "",
  "contactInformation": "Updated Contact Info"
}
```

Activate or deactivate an existing provider

Syntax

PATCH: /api/v1/providers/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	ID of the provider	String

Schema

Status change data for the provider; see example below

Responses

Code	Description
200	Provider status updated successfully.
403	Error updating provider status.

Example

```
{
  "active": false
}
```

Upload service provider image

Syntax

PUT: /api/v1/providers/upload-image/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	ID of the provider	String

Schema

None

Responses

Code	Description
200	Provider updated successfully.
500	Error updating provider.

Example

None

Retrieve list of users from the service provider with pagination & optional search

Syntax

GET: /api/v1/providers/list/users/{id}

Description

Authorization Schema: JWT
Response body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
Page	Optional	Page number	Number
Items	Optional	Number of providers per page	Number
Search	Optional	Filter by provider name	String
Id	Optional	Optional provider ID	String

Schema

None

Responses

<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>200</td><td>

```
{
  "statusCode": 200,
  "customCode": "WGE0139",
  "data": {
    "providers": [
      {
        "id": "123",
        "name": "Provider Name",
        "description": "Provider Description",
        "email": "provider@example.com",
        "phone": "+123456789",
        "einNumber": "12-3456789",
        "country": "Argentina",
        "city": "Balvanera",
        "zipCode": "59569",
        "companyAddress": "Company Address",
        "walletAddress": "",
        "logo": "https://example.com/logo.png",
        "contactInformation": "Contact Info"
      }
    ],
    "currentPage": 1,
    "total": 1,
    "totalPages": 1
  }
}
```

</td></tr>
<tr><td>403</td><td>Access forbidden.</td></tr>
</table>

Example

None

Upload service provider image

Syntax

PUT: /api/v1/providers/edit/users/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	ID of the provider	String

Schema

UpdateUserDto, p. 10

Responses

Code	Description
200	Provider updated successfully.
403	Error updating provider.

Example

```
{}
```

Create or update a fee configuration for service providers

Syntax

PATCH: /api/v1/providers/create/fee-configurations

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	`{}`
201	Fee Configuration created successfully.
404	feeConfigurationId for service provider not found.

Example

```
{
  "comision": 1,
  "percent": 1,
  "base": 2,
  "serviceProviderId": "8bf931ea-3710-420b-ae68-921f94bcd937",
  "feeConfigurationId": "8bf931ea-3710-420b-ae68-921f94bcd937"
}
```

Get Provider Fee Configurations

Syntax

GET: /api/v1/providers/fee-configurations/{providerId}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
providerId	Required	providerId	String

Schema

None

Responses

Code	Description
200	List of fee configurations obtained successfully.

Example

None

payment

Create a new payment parameters for service providers

Syntax

POST: /api/v1/payments/create/payment-parameters

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	**Description**
201	Payment Parameter created succefully.

Example

```
{
  "name": "Provider1_Paramter",
  "cost": 10,
  "frequency": 30,
  "timeIntervalId": "5894d088-0cc6-4aea-9df5-ba348c9d364d",
  "serviceProviderId": "8bf931ea-3710-420b-ae68-921f94bcd937"
}
```

Update a payment parameter for service providers

Syntax

PUT: /api/v1/payments/payment-parameters/{paymentParameterId}

Description

Authorization Schema: JWT
Request body Schema: application/json
Response Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
paymentParameterId	Required	paymentParameterId	String

Schema

See example below

Responses

Code	Description
200	`{}`
201	Payment Parameter updated succefully.
404	PaymentParameterId for service provider not found.

Example

```
{
  "name": "Provider1_Paramter",
  "cost": 10,
  "frequency": 30,
  "timeIntervalId": "5894d088-0cc6-4aea-9df5-ba348c9d364d",
  "serviceProviderId": "8bf931ea-3710-420b-ae68-921f94bcd937"
}
```

List payment parameters for a service providers

Syntax

GET: /api/v1/payments/list/payment-parameters

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
page	Optional	Page number	Number
items	Optional	Number of payments parameters per page	Number
search	Optional	Filter by payment parameter name	String
serviceProviderId	Optional	Filter by service provider id	String

Schema

[[SCHEMA name AND LINK!!!]]

Responses

Code	Description
200	List of payment parameters obtained successfully.

Example

None

List time intervals

Syntax

GET: /api/v1/payments/list/time-intervals

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
200	List of time intervals obtained successfully.

Example

None

Toggle the active status of a payment parameter

Syntax

PATCH: /api/v1/payments/payment-parameters/{paymentParameterId}/toggle

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
paymentParameterID	Required	ID of the payment parameter	String

Schema

Optional, any (see example)

Responses

Code	Description
200	Payment parameter status toggled successfully.
404	Payment parameter not found.

Example

```
{
   "serviceProviderId": "8bf931ea-3710-420b-ae68-921f94bcd937"
}
```

modules

ModuleController_findAll

Syntax

GET: /api/v1/modules

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
belongs	Optional	Belongs	String
isProvider	Required	isProvider	Boolean

Schema

None

Responses

Code	Description
200	Successfully returned modules

Example

None

settings

SettingController_findAll

Syntax

GET: /api/v1/settings

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
belongs	optional	belongs	String

Schema

None

Responses

Code	Description
200	Successfully returned settings

Example

None

SettingController_update

Syntax

PUT: /api/v1/settings/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	id	String

Schema

UpdateSettingsDto, p. 12

Responses

Code	Description
200	The record has been successfully updated.
403	Forbidden.

Example

```
{
  "value": "string"
}
```

notifications

Check if notifications are active for a user

Syntax

GET: /api/v1/notifications/status/{userId}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
userId	Required	ID of the user	String

Schema

None

Responses

Code	Description
200	Notification status retrieved successfully.
404	User not found

Example

None

Activate or deactivate all notifications for a user

Syntax

POST: /api/v1/notifications/toggle/{userId}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
userId	Required	ID of the user whose notifications will be updated	String

Schema

Required; see example below

Responses

Code	Description
201	Notifications status updated successfully.
404	User not found.

Example

```
{
   "notificationsActive": true
}
```

Mute notifications for a user

Syntax

POST: /api/v1/notifications/mute/{userId}

Description

Mute notifications for a certain duration
Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
userId	Required	ID of the user whose notifications will be muted	String

Schema

MuteNotificationsDto, p. 12

Responses

Code	Description
201	Notifications muted successfully.
400	Invalid mute duration.
404	User not found.

Example

30 minutes

```
{
  "duration": "30m"
}
```

health-check

HealthController_healthCheck

Syntax

POST: /api/v1/health-check

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
201	Verify health check.
403	Forbidden.

Example

None.

HealthController_getInfoUptime

Syntax

GET: /api/v1/health-check/uptime

Description

Authorization Schema: JWT
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	`{}`
201	Retrieve Uptime Information
403	Forbidden

Example

None

WALLET ENDPOINTS

Comprehensive documentation for the Wallet Guru API, detailing the wallet service and its endpoints

wallet

Checks if a wallet address exists

Syntax

GET: Checks if a wallet address exists

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
address	Optional	Address	String

Schema

None

Responses

Code	Description
200	Wallet address successfully verified.
206	Incomplete parameters
401	Unauthorized access.
500	Server error.

Example

None

Return info wallet

Syntax

GET: /api/v1/wallets/info

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
address	Optional	Address	String

Schema

None

Responses

Code	Description
200	Wallet address successfully verified.
206	Incomplete parameters
401	Unauthorized access.
500	Server error.

Example

None

Create a new wallet

Syntax

POST: /api/v1/wallets

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

CreateWalletDto, p. 12

Responses

Code	Description
201	Wallet successfully created.
403	Forbidden access.

Example

```
{
  "name": "string",
  "walletType": "string",
  "walletAddress": "string"
}
```

Retrieve all wallets

Syntax

GET: /api/v1/wallets

Description

Authorization Schema: JWT
Response body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	Optional	id	String
name	Optional	name	String
walletType	Optional	walletType	String
walletAddress	Optional	walletAddress	String
active	Optional	active	Boolean
search	Optional	search	String
items	Optional	items	Number
page	Optional	page	Number

Schema

For Response: GetWalletDto, p. 13

Responses

<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>200</td><td>

```
[
  {
    "id": "string",
    "name": "string",
    "walletType": "string",
    "walletAddress": "string",
    "active": true,
    "search": "string",
    "items": 0,
    "page": 0
  }
]
```

</td></tr>
<tr><td>500</td><td>Server error.</td></tr>
</table>

Example

None

Update an existing wallet

Syntax

PUT: /api/v1/wallets/{id}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	Wallet ID	String

Schema

UpdateWalletDto, p. 13.

Responses

Code	Description
200	Wallet successfully updated.
403	Forbidden access
404	Wallet not found.

Example

```
{
  "id": "string",
  "name": "string",
  "walletType": "string",
  "walletAddress": "string"
}
```

Retrieve a wallet by ID

Syntax

GET: /api/v1/wallets/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	Id	String
walletId	Required	Wallet ID	String

Schema

None

Responses

Code	Description
200	Wallet successfully retrieved.
404	Wallet not found.
500	Server error.

Example

None

Toggle the activation state of a wallet

Syntax

PATCH: /api/v1/wallets/{id}/toggle

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	Wallet ID	String

Schema

None

Responses

Code	Description
200	Wallet state successfully toggled.
404	Wallet not found.
500	Server error.

Example

None

Retrieve a wallet using a token

Syntax

GET: /api/v1/wallets/wallet/token

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
200	Wallet successfully retrieved.
401	Unauthorized access.
500	Server error.

Example

None

Create a new refund

Syntax

POST: /api/v1/wallets/refunds

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

CreateRefundsDto, p. 13

Responses

Code	Description
201	Refund created.
403	Forbidden access

Example

```
{
  "Id": "string",
  "amount": 0,
  "description": "string",
  "activityId": "string",
  "serviceProviderId": "string",
  "walletAddress": "string"
}
```

Get refunds

Syntax

GET: /api/v1/wallets/get/refunds

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
walletAddress	Optional	walletAddress	String
serviceProviderId	Optional	serviceProviderId	String
page	Optional	page	String
items	Optional	items	String
startDate	Optional	startDate	String
endDate	Optional	endDate	String

Schema

None

Responses

Code	Description
200	Refunds successfully retrieved.
404	Refund not found.
500	Server error.

Example

None

Retrieve provider revenues

Syntax

GET: /api/v1/wallets/provider/revenues/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Optional	Provider ID	String
createDate	Optional	Start date for filtering, ISO format	String
endDate	Optional	End date for filtering, ISO format	String

Schema

None

Responses

Code	Description
200	Provider revenues successfully retrieved.
401	Unauthorized access.
500	Server error.

Example

None

wallet-rafiki

Create a new wallet address

Syntax

POST: /api/v1/wallets-rafiki/address

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Data required to create a new wallet address
CreateRafikiWalletAddressDto, p. 13

Responses

Code	Description
201	Wallet Address Created Successfully
400	Invalid Address Name or Address Name Already in Use
500	Internal Server Error

Example

```
{
  "addressName": "/^(?=a)(?=\\d)a$/",
  "assetId": "string"
}
```

Create a new wallet address (Service Provider)

Syntax

POST: /api/v1/wallets-rafiki/service-provider-address

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Data required to create a new wallet address
CreateRafikiWalletAddressDto, p. 13

Responses

Code	Description
201	Wallet address creating successfully
400	Invalid address name or address name already in use
500	Internal server error

Example

```
{
  "addressName": "/^(?=a)(?=\\d)a$/",
  "assetId": "string"
}
```

Retrieve all Rafiki assets

Syntax

GET: /api/v1/wallets-rafiki/assets

Description

Authorization Schema: JWT

Parameters

None

Schema

Mone

Responses

Code	Description
200	Assets successfully retrieved
401	Unauthorized access
500	server error

Example

None

List all user transactions

Syntax

GET: /api/v1/wallets-rafiki/list-transactions

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
type	Optional	type	String
userType	Optional	userType	String
startDate	Optional	startDate	String
endDate	Optional	endDate	String
state	Optional	state	String
activityId	Optional	activityId	String
walletAddress	Optional	walletAddress	String
page	Optional	page	String
items	Optional	items	String
isRevenue	Optional	isRevenue	Boolean
orderBy	Optional	orderBy (Order by fields (e.g., providerId, date))	Array of strings
report	Optional	report (Order by fields (e.g., user, period, revenue, dispute, reserved))	Array of strings
providerIDs	Optional	providerIDs	Array of strings

Schema

None

Responses

Code	Description
200	Transactions successfully retrieved.
206	Incomplete parameters.
401	Unauthorized access.
500	Server error.

Example

None

Detail transaction IDs

Syntax

GET: /api/v1/wallets-rafiki/transaction

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
transactionIds	Optional	Transaction IDs	String

Schema

None

Responses

Code	Description
200	Transactions successfully retrieved.
401	Unauthorized access.
500	Server error.

Example

None.

Create a new transaction

Syntax

POST: /api/v1/wallets-rafiki/transaction

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

ReceiverInputDTO, p. 13

Responses

Code	Description
201	Transaction successfully created.
400	Bad request.
401	Unauthorized access.
500	Server error.

Example

```
{
  "amount": 0,
  "walletAddressUrl": "string",
  "walletAddressId": "string"
}
```

Detail transaction

Syntax

GET: /api/v1/wallets-rafiki/transaction/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	id	Strings

Schema

None

Responses

Code	Description
200	Transaction successfully retrieved.
401	Unauthorized access.
500	Server error

Example

None

Download all provider transactions

Syntax

GET: /api/v1/wallets-rafiki/download-transactions-provider

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
type	Optional	type	String
startDate	Optional	startDate	String
endDate	Optional	endDate	String
State	Optional	State	String
page	Optional	page	String
items	Optional	items	String
isRevenue	Optional	isRevenue	Boolean
orderBy	Optional	orderBy (order by fields (e.g., providerId, date))	Array of strings
report	Optional	Report (order by fields (e.g., user, period, revenue, dispute, reserved))	Array of strings
providerIds	Optional	providerIds	Array of strings

Schema

None

Responses

Code	Description
200	Transactions successfully downloaded.
206	Incomplete parameters.
401	Unauthorized access.
500	Server error.

Example

None

Download all transactions by activityId

Syntax

GET: /api/v1/wallets-rafiki/download-transactions-activity

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
type	Optional	type	String
startDate	Optional	startDate	String
endDate	Optional	endDate	String
state	Optional	State	String
activityId	Optional	activityId	String
page	Optional	page	String
items	Optional	items	String
isRevenue	Optional	isRevenue	Boolean
orderBy	Optional	orderBy (order by fields (e.g., providerId, date))	Array of strings
report	Optional	Report (order by fields (e.g., user, period, revenue, dispute, reserved))	Array of strings

Schema

None

Responses

Code	Description
200	Transactions successfully downloaded.
206	Incomplete parameters.
401	Unauthorized access.
500	Server error.

Example

None

Download all user transactions

Syntax

GET: /api/v1/wallets-rafiki/download-transactions-user

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
type	Optional	type	String
startDate	Optional	startDate	String
endDate	Optional	endDate	String
State	Optional	State	String
page	Optional	page	String
items	Optional	items	String
walletAddress	Optional	walletAddress	String
activityId	Optional	activityId	String
isRevenue	Optional	isRevenue	Boolean
orderBy	Optional	orderBy (order by fields (e.g., providerId, date))	Array of strings
report	Optional	Report (order by fields (e.g., user, period, revenue, dispute, reserved))	Array of strings
providerIds	Optional	providerIds	Array of strings

Schema

None

Responses

Code	Description
200	Transactions successfully downloaded
206	Incomplete parameters.
401	Unauthorized access.
500	Server error.

Example

None

Download all user provider transactions

Syntax

GET: /api/v1/wallets-rafiki/download-transactions-user-provider

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
search	Optional	search	String
type	Optional	type	String
startDate	Optional	startDate	String
endDate	Optional	endDate	String
state	Optional	State	String
page	Optional	page	String
items	Optional	items	String
isRevenue	Optional	isRevenue	Boolean
orderBy	Optional	orderBy (order by fields (e.g., providerId, date))	Array of strings
report	Optional	Report (order by fields (e.g., user, period, revenue, dispute, reserved))	Array of strings
providerIds	Optional	providerIds	Array of strings

Schema

None

Responses

Code	Description
200	Transactions successfully downloaded.
206	Incomplete parameters.
401	Unauthorized access.
500	Server error.

Example

None

Create a transaction to link a service provider

Syntax

POST: /api/v1/wallets-rafiki/service-provider-link

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

LinkInputDTO, p. 14

Responses

Code	Description
201	Service provider successfully linked.
400	Bad request.
401	Unauthorized access.
500	Server error.

Example

```
{
  "walletAddressUrl": "string",
  "walletAddressId": "string",
  "sessionId": "string"
}
```

Retrieve linked service providers

Syntax

GET: /api/v1/wallets-rafiki/linked-providers

Description

Authorization Schema: JWT

Parameters

None

Schema

None

Responses

Code	Description
200	Service providers successfully retrieved.
400	Bad request.
401	Unauthorized access.
500	Server error

Example

None

Unlink a service provider by session id

Syntax

POST: /api/v1/wallets-rafiki/service-provider-unlink

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

UnLinkInputDTO, p. 14

Responses

Code	Description
201	Service provider successfully unlinked
400	Bad request.
401	Unauthorized access.
500	Server error.

Example

```
{
   "sessionId": "string"
}
```

Retrieve all exchange rates

Syntax

GET: /api/v1/wallets-rafiki/exchange-rates

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
base	Required	base	String
search	Optional	Search	String

Schema

None

Responses

Code	Description
200	Exchange rates successfully retrieved.
401	Unauthorized access.
500	Server error.

Example

None

Create a deposit

Syntax

POST: /api/v1/wallets-rafiki/create-deposit

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

DepositDTO, p. 14

Responses

Code	Description
200	Deposit created successfully
201	`{ }`
400	Bad request.
500	Server error.

Example

```
{
  "walletAddressId": "string",
  "amount": 0
}
```

Open payment - Create transaction

Syntax

POST: /api/v1/wallets-rafiki/create-transaction-open-payment

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Auth Open Payment; See example below

Responses

Code	Description
201	`{}`

Example

```
{
  "senderWalletAddress": "",
  "receiverWalletAddress": "",
  "amount": "",
  "assetCode": "",
  "assetScale": "",
  "metadata": {},
  "authHost": "",
  "paymentHost": "",
  "interactionHost": ""
}
```

Get wallet address by Rafiki ID

Syntax

GET: /api/v1/wallets-rafiki/{id}/asset

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	id	String

Schema

None

Responses

Code	Description
200	Wallet address retrieved successfully.
401	Unauthorized access.
500	Server error.

Example

None

Create an Incoming Payment

Syntax

POST: /api/v1/wallets-rafiki/create/incoming-payment

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

CreatePaymentDTO, p. 14

Responses

Code	Description
200	Incoming Payment created successfully.
201	`{ }`
400	Bad Request
401	Unauthorized access.
500	Server error.

Example

```
{
  "incomingAmount": 0,
  "walletAddressUrl": "string",
  "walletAddressId": "string"
}
```

Execute an outgoing payment action

Syntax

POST: /api/v1/wallets-rafiki/action/outgoing-payment

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

ActionOugoingPaymentDto, p. 14

Responses

Code	Description
200	Outgoing payment action successfully executed.
201	`{ }`
400	Bad Request
401	Unauthorized access.
500	Server error.

Example

```
{
  "outgoingPaymentId": "string",
  "action": "string"
}
```

List all incoming payments

Syntax

GET: /api/v1/wallets-rafiki/list-incoming-payments

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
status	Optional	status	Boolean
startDate	Optional	startDate	String
endDate	Optional	endDate	String
walletAddress	Optional	walletAddress	String
serviceProviderId	Optional	serviceProviderId	String
page	Optional	page	String
items	Optional	items	String
search	Optional	search	String

Schema

None

Responses

Code	Description
200	Incoming payments retrieved successfully.
401	Unauthorized access.
500	Server error.

Example

None

Cancel an incoming payment by ID

Syntax

PATCH: /api/v1/wallets-rafiki/{id}/cancel-incoming

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	Id	String

Schema

None

Responses

Code	Description
200	Incoming payment cancelled successfully
401	Unauthorized access.
404	Incoming payment not found.
500	Server error.

Example

None

Open payment - auth payment

Syntax

POST: /api/v1/wallets-rafiki/auth-payment

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Auth open payment; See example below

Responses

Code	Description
201	`{ }`

Example

```
{
  "clientWalletAddress": "0x123456789abcdef"
}
```

Open payment - incoming payment

Syntax

POST: /api/v1/wallets-rafiki/incoming-payment

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Incoming open payment; see example below

Responses

Code	Description
201	`{ }`

Example

```
{
  "receiverWalletAddress": "0x123456789abcdef"
}
```

Open payment - outgoing payment

Syntax

POST: /api/v1/wallets-rafiki/outgoing-payment-auth

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Outgoing payment authentication; see example below

Responses

Code	Description
201	`{ }`

Example

```
{
  "senderWalletAddress": "0x987654321fedcba",
  "clientWalletAddress": "0xa1b2c3d4e5f67890",
  "debitAmount": 100,
  "receiveAmount": 95
}
```

Open payment - continue

Syntax

POST: /api/v1/wallets-rafiki/continue/{continueId}

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
continueId	Required	continueId	String

Schema

Continue open payment interaction; see example below.

Responses

Code	Description
201	`{ }`

Example

```
{
  "interact_ref": "ref12345"
}
```

Open payment - create outgoing payment

Syntax

POST: /api/v1/wallets-rafiki/outgoing-payment

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

Create outgoing open payment; see example below

Responses

Code	Description
201	`{ }`

Example

```
{
  "senderWalletAddress": "0x123456789abcdef",
  "incomingPaymentUrl": "https://payment-url.com",
  "debitAmount": 150
}
```

Open payment - get outgoing payment

Syntax

GET: /api/v1/wallets-rafiki/outgoing-payment/{outgoingPaymentId}

Description

Authorization Schema: JWT
Response body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
OutgoingPaymentId	Required	outgoingPaymentId	String

Schema

None

Responses

Code	Description
200	`{ }`

Example

None

webhook

Recieve event from rafiki hook

Syntax

POST: /api/v1/webhook

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

EventWebHookDTO, p. 14

Responses

Code	Description
201	Event received successfully
403	Bad request

Example

```
{
  "id": "string",
  "type": "string",
  "data": {}
}
```

clear-payments

Confirm a clear payment

Syntax

POST: /api/v1/clear-payments/confirm

Description

Authorization Schema: JWT
Request body Schema: application/json

Parameters

None

Schema

ConfirmClearPaymentDto, p. 15

Responses

Code	Description
200	Clear Payment Has Been Confirmed
201	`{ }`
400	Error Confirming Clear Payment
500	Internal Server Error

Example

```
{
  "clearPaymentId": "string",
  "observations": "string",
  "referenceNumber": "string"
}
```

Retrieve provider revenues by Id

Syntax

GET: /api/v1/clear-payments/{id}

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
id	Required	Provider revenue ID	String

Schema

None

Responses

Code	Description
200	Provider revenues successfully retrieved.
401	Unauthorized access.
500	Server error.

Example

None

List all clear payments

Syntax

GET: /api/v1/clear-payments/list/payments

Description

Authorization Schema: JWT

Parameters

Parameter	Optional/Required	Description	Data Type
month	Optional	month	Number
year	Required	year	Number
providerId	Optional	providerId	String
status	Optional	status	Boolean
page	Optional	page	String
Items	Optional	Items	String

Schema

None

Responses

Code	Description
200	Clear payments successfully retrieved.
206	Incomplete parameters.
401	Unauthorized access.
500	Server error.

Example

None

CODES ENDPOINTS

Comprehensive documentation for the Wallet Guru API, detailing the codes service and its endpoints.

Codes

Retrieve a list of codes

Syntax

GET: /api/v1/codes/{lang}

Description

Retrieve a list of codes

Parameters

Parameter	Optional/Required	Description	Data Type
lang	Required	lang	String
search	Optional	Optional search query	String

Schema

None

Responses

Code	Description
200	Codes retrieved successfully.
403	Access forbidden.

Example

None

Retrieve a single code by ID and Lang

Syntax

GET: /api/v1/codes/{lang}/{codeId}

Description

Retrieve a single code by ID and Lang

Parameters

Parameter	Optional/Required	Description	Data Type
Lang	Optional	Lang of the code	String
codeId	Required	Optional search query	String

Schema

None

Responses

Code	Description
200	Code found.
404	Code not found.

Example

None

COUNTRIES & CITIES API

API calls for the "Countries Now" Countries and Cities API, also documented online at: https://documenter.getpostman.com/view/1134062/T1LJjU52

Population

Get single city and its population data

Syntax

POST: https://countriesnow.space/api/v0.1/countries/population/cities

Description

Request body Schema: application/json

Parameters

None

Schema

See example below.

Responses

Code	Description

200	{
	  "error": false,
	  "msg": "lagos with population",
	  "data": {
	    "city": "Lagos",
	    "country": "Nigeria",
	    "populationCounts": [
	      {
	        "year": "1991",
	        "value": "5195247",
	        "sex": "Both Sexes",
	        "reliabilty": "Final figure, complete"
	      }
	    ]
	  }
	}

Example

```
{
      "city": "lagos"
}
```

Filter cities and population data

Syntax

POST: https://countriesnow.space/api/v0.1/countries/population/cities/filter

Description

Filter cities and population data by Country, Order,
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Returns Filtered result]

Example

```
{
        "limit": 10,
        "order": "asc",
        "orderBy": "name",
        "country": "nigeria"
}
```

Get cities and their population data

Syntax

GET: https://countriesnow.space/api/v0.1/countries/population/cities

Description

Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [Returns all cities with population]

Example

None

Filter countries population data

Syntax

POST: https://countriesnow.space/api/v0.1/countries/population/filter

Description

Allowed queries are: { year: Number, limit: Number, gt: Number, lt: Number, order: String, orderBy: String }
Request body Schema: application/json

Parameters

None

Schema

See example below.

Responses

Code	Description
200	OK –[Returns filtered result]

Example

```
{       "year": 2000,
        "limit": 10,
        "lt": 651348588,
        "gt": 6513485,
        "orderBy": "name",
        "order": "dsc"
}
```

Get single country and population data

Syntax

POST: https://countriesnow.space/api/v0.1/countries/population

Description

Request body Schema: application/json

Parameters

None

Schema

See example below.

Responses

Code	Description
200	OK []Returns country name and population]

Example

```
{
        "country": "nigeria"
}
```

Get all countries and respective population

Syntax

GET: https://countriesnow.space/api/v0.1/countries/population

Description

Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	**Description**
200	OK – return asll countries and populations 1961-2018

Example

None

Location

Endpoints related to countries and cities location (long, lat)

Get countries by position range

Syntax

POST: https://countriesnow.space/api/v0.1/countries/positions/range

Description

Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Returns countries between long of min and max values]

Example

```
{
        "type": "long",
        "min": 1,
        "max": 40
}
```

Get single country and its positions

Syntax

POST: https://countriesnow.space/api/v0.1/countries/positions

Description

Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [country position retrieved]

Example

```
{
    "country": "nigeria"
}
```

Get countries and their positions

Syntax

GET: https://countriesnow.space/api/v0.1/countries/positions

Description

Get all countries with their longitude and latitude
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [Countries and Positions retrieved]

Example

None

Flag

Get single country and flag URL

Syntax

POST: https://countriesnow.space/api/v0.1/countries/flag/images

Description

Get single country with flag image
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Country and flag retrieved]

Example

```
{
    "iso2": "NG"
}
```

Get single country and unicode flag

Syntax

POST: https://countriesnow.space/api/v0.1/countries/flag/unicode

Description

Get a single country with its flag in unicode format
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Country and Unicode flag retrieved]

Example

```
{
    "iso2": "NG"
}
```

Get countries and unicode flags

Syntax

GET: https://countriesnow.space/api/v0.1/countries/flag/unicode

Description

Get all countries with their flag in unicode format
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	**Description**
200	OK [Countries and unicode flags retrieved]

Example

None

Countries and flag URL

Syntax

GET: https://countriesnow.space/api/v0.1/countries/flag/images

Description

Get all countries with their flag images url
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	**Description**
200	OK [flags images retrieved]

Example

None

Currencies

Get countries and their currencies

Syntax

GET: https://countriesnow.space/api/v0.1/countries/currency

Description

Get all countries and their currencies
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [countries and currencies retrieved]

Example

None

Get single country and its currency

Syntax

POST: https://countriesnow.space/api/v0.1/countries/currency

Description

Get currencies of a single country
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Country and currency retrieved]

Example

```
{
    "country": "nigeria"
}
```

Capital

Get countries and their capital

Syntax

GET: https://countriesnow.space/api/v0.1/countries/capital

Description

Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [Countried and capitals retrieved]

Example

None

Get a single country and capital

Syntax

POST: https://countriesnow.space/api/v0.1/countries/capital

Description

Get a specific country and its capital
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Country and capital retrieved]

Example

```
{
    "country": "nigeria"
}
```

Cities

Get cities of a specified country

Syntax

POST: https://countriesnow.space/api/v0.1/countries/cities

Description

Get all cities in a specified country
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Cities in Country retrieved]

Example

```
{
    "country": "nigeria"
}
```

Get all countries and their cities

Syntax

GET: https://countriesnow.space/api/v0.1/countries

Description

Get all countries and thier respective cities
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	**Description**
200	OK [Countries and cities retrieved]

Example

None

Get cities in a state

Syntax

POST: https://countriesnow.space/api/v0.1/countries/state/cities

Description

Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Cities in state X of country Y retrieved]

Example

```
{
    "country": "Nigeria",
    "state": "Lagos"
}
```

Codes

Get countries and ISO2&3 Codes

Syntax

GET: https://countriesnow.space/api/v0.1/countries/iso

Description

Get all countries with their ISO2&3 Codes
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [countries and ISO Codes retrieved]

Example

None

Get Single Country and ISO2&3 Codes

Syntax

POST: https://countriesnow.space/api/v0.1/countries/iso

Description

Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [Country's ISO code retrieved]

Example

```
{
    "country": "Nigeria"
}
```

Get cities and dial codes

Syntax

GET: https://countriesnow.space/api/v0.1/countries/codes

Description

Get all countries, country code and dial codes
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [Countries and codes retrieved]

Example

None

Get single country dial codes

Syntax

POST: https://countriesnow.space/api/v0.1/countries/codes

Description

Get single country dial codes
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	**Description**
200	OK [Country dial code retrieved]

Example

```
{
    "country": "Nigeria"
}
```

Filter

Get only specified countries information

Syntax

GET:
https://countriesnow.space/api/v0.1/countries/info?returns=currency,flag,unicodeFlag,dialCode

Description

Get only specified countries information
Response body Schema: application/json

Parameters

Parameter	Optional/Required	Description	Data Type
returns	?	currency,flag,unicodeFlag,dialCode	Object

Schema

None

Responses

Code	Description
200	OK [Countries details: 'currency,flag,unicodeFlag,dialcode' have been retrieved]

Example

None

States

Get all countries and their states

Syntax

GET: https://countriesnow.space/api/v0.1/countries/states

Description

Get every country and their respective states
Response body Schema: application/json

Parameters

None

Schema

None

Responses

Code	Description
200	OK [Countries and states retrieved]

Example

None

Get single country and its states

Syntax

POST: https://countriesnow.space/api/v0.1/countries/states

Description

Get a country and its states
Request body Schema: application/json

Parameters

None

Schema

See example below

Responses

Code	Description
200	OK [States in Country retrieved]

Example

```
{
    "country": "Nigeria"
}
```