



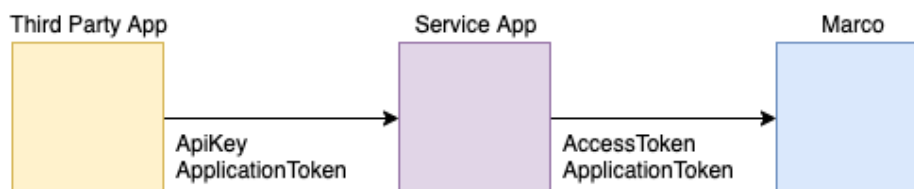
Third Party API Integration

How it Works

A Third Party Application can consume the Service Application by providing an **ApiKey** registered and accepted on the Service Application.

This **ApiKey** is associated with an active **MarcoApplication** representing the Third Party Application.

When a valid **ApiKey** is received by the Service Application, the Service Application can execute API calls to Marco authenticated as the **ServiceAccount** assigned to the **ApiKey**.



Marco Requirements

Resource	Description	Permissions
Marco Application	Represents the Third Party Application for executing API calls to the Service Application providing the its <i>applicationToken</i> and assigning its <i>marppId</i> to the Service Application ApiKey.	
Marco Cloud Wallet	For signing Service Application ledger transactions.	
Marco Service Account	Assigned to the Service Application ApiKey for executing authenticated API requests to Marco.	<ul style="list-style-type: none">• Role MARPP_USER_BASIC on the Marco Cloud Wallet to use.• Allowed access by the Marco Application



Service Application Requirements

Resource	Description
Api Key	For executing API requests from the Third Party Application to the Service Application.

API Request Base Structure

Api Server

Protocol	Host
https://	api-blockstamp.finboot.io

Request Headers

Name	Value
Authorization	ApiKey <YOUR_BASE_64_ENCODED_API_KEY>
ApplicationToken	<YOUR_APPLICATION_TOKEN>

API Methods

BS-STAMPS-1 Create Stamp

Method

Property	Value
Path	/stamps
Method	POST

**Body**

Property	Value
Media Type	multipart/form-data

Form Parts

Part Name	Type	Description
stamp	application/json	JSON representing <i>Stamp</i> object with required fields <i>description</i> , <i>fileType</i> and <i>workspace</i> (view Data Models sections).
cloudWalletId	string	Marco Cloud Wallet to sign <i>ledgerTransaction</i> .
cloudWalletPassword	string	Password to unlock Cloud Wallet identified by <i>cloudWalletId</i> .
file	multipartFile	Binary file to be stamped.

Response

Property	Type	Description
Status Code	HTTP Status Code	201 CREATED
Body	string	JSON representing <i>Stamp</i> object created (view Data Models sections).

BS-STAMPS-2 List Stamps on Workspace**Method**

Property	Value
Path	/stamps
Method	GET



Query Parameters

Property	Type	Description
workspaceID	string	ID of <i>Workspace</i> containing list of <i>Stamps</i> to be returned.

Request Headers

Name	Value
Content-Type	application/json

Response

Property	Type	Description
Status Code	HTTP Status Code	200 Ok
Body	string	JSON representing list of <i>Stamps</i> objects obtained (view Data Models sections).

BS-STAMPS-3 Get Stamp

Method

Property	Value
Path	/stamps/{stampID}
Method	GET

Path Variables

Property	Type	Description
stampID	string	ID of <i>Stamp</i> to get

Request Headers

Name	Value
Content-Type	application/json



Response

Property	Type	Description
Status Code	HTTP Status Code	200 Ok
Body	string	JSON representing <i>Stamp</i> object obtained (view Data Models sections).

Data Models

Ledger

Property	Type
ledgerID	string

SmartContractInstance

Property	Type
smartContractInstanceID	string

LedgerTransaction

Property	Type
hash	string
ledger	Ledger
walletID	string
smartContractInstance	SmartContractInstance
state	string
dateCreated	long



LedgerWorkspace

Property	Type
ledgerWorkspaceID	string
name	string
ledger	Ledger
stampSmartContractInstance	SmartContractInstance
confirmationsSmartContractInstance	SmartContractInstance

Workspace

Property	Type
workspaceID	string
name	string

Stamp

Property	Type
stampID	string
description	string
fileType	string
stampHash	string
state	string
ledgerTransaction	LedgerTransaction
workspace	Workspace
ledgerWorkspace	LedgerWorkspace
dateCreated	long
dateUpdated	long