|  |
| --- |
| *iRESLab, Inc.*  17/05/2018 |
|  |
| Disclaimer: All information and Intellectual Property contained in this document is the Proprietary and Confidential property of iRESlab, Inc. and intended client, and all rights herein are expressly reserved. By accepting this material the recipient agrees that the information contained herein be held in the strictest confidence and that it will not be used, copied, reproduced in whole or in part for any purpose other than it is intended. The contents shall not be revealed to any third party without iRESlab’s prior written consent. |
|  |
|  |

|  |
| --- |
| API specification document for  CoinClaim REST APIs |
| Version 0.3 |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Author** | **Section(s)** | **Revision Notes** | **Date** |
| 0.1 | iRESlab |  | Initial draft | 05 Mar 2018 |
| 0.2 | iRESlab | 2 | Updated endpoint details | 06 April 2018 |
| 0.3 | iRESlab |  | Support for Ethereum (ETH) added | 17 May 2018 |

Table of Contents

[1. Introduction 4](#_Toc511246227)

[2. REST APIs 4](#_Toc511246228)

[2.1. Access Token 4](#_Toc511246229)

[2.2. Generate Address 5](#_Toc511246230)

[2.3. Transfer Tokens 6](#_Toc511246231)

[2.4. Account Balance 7](#_Toc511246232)

[3. Response Codes 8](#_Toc511246233)

# Introduction

This document contains the information about service endpoints, method types, request-response parameters for all the REST APIs to be consumed by CoinClaim.io backend system.

Services will be exposed as rest services over HTTPS (Http over SSL or HTTP Secure) protocol.

# REST APIs

## Access Token

API authentication and authorization implemented using OAUTH 2.0. Access token provides access to all other APIs.

**URL:** [{URL}/api/oauth/token?grant\_type=client**\_**credentials](http://localhost:8086/electra/oauth/token?grant_type=client_credentials)

**Request Type** : POST

**Headers:**

* **Content-Type :** application/json
* **Authorization :** Basic <Base64Hash(client\_id:client\_secret)>
  + **client\_id =** coinclaim
  + **client\_secret =** c@incl&im

**Parameter List:** Nil

**Sample Request (JSON Payload):**

{ }

**Sample Response (JSON Payload):**

* ***Success:***

{

"access\_token": "395a94f6-7849-472d-806b-dc31eaf14640",

"token\_type": "bearer",

"expires\_in": 59,

"scope": "trust read write"

}

* ***Error:***

{

"timestamp": 1520255664497,

"status": 401,

"error": "Unauthorized",

"message": "Bad credentials",

"path": "/api/oauth/token"

}

## Generate Address

Generates unique address for supported blockchains i.e. Ethereum & Bitcoin and associate that addresses to a client/company identified by ClientCorrelationId in request.

**URL**: {URL}/api/address?access\_token=<access-token>

**Request Type** : POST

**Headers:**

* **Content-Type :** application/json

**Parameter List:**

* clientCorrelationId : uniquely identifies the company

**Sample Request (JSON Payload):**

{

"clientCorrelationId": "4cc86e25641048788d87fad7df010604",

}

**Sample Response (JSON Payload):**

* ***Success:***

{

"status": 200,

"code": 100,

"accountDetails": [{

"type": "BTC",

"address": " mnNpnxsS7QVFHeR9XdB5RaNuorrPGPmPNA"

},

{

"type": "ETH",

"address": "0x64aD6313dd83Fd4D074C78D5f291bD264c317510"

}

]

}

* ***Error*:**

{

"status": 400,

"code": 900,

"errors": [{

"code": 805,

"message": "Client already exists for Correlation Id"

}]

}

{

"error": "invalid\_token",

"error\_description": "Invalid access token: fdc31850-ac27-42ad-abe4-dc1e56685eca"

}

**Notes:**

* List of all possible response ‘status’ & corresponding ‘description’ in JSON response will be predefined. Refer to Section x.x for response codes.

## Transfer Tokens

Transfer tokens from company account to user account.

**URL**: {URL}/api/transfer?access\_token=<access-token>

**Request Type** : POST

**Headers:**

* **Content-Type :** application/json

**Parameter List**:

* clientCorrelationId
* type : BTC
* beneficiaryAddress : user bitcoin address where coins will be allocated to
* noOfTokens : total tokens to be transferred

**Sample Request (JSON Payload):**

{

"clientCorrelationId": "4cc86e25641048788d87fad7df010601",

"type": "BTC",

"noOfTokens": "0.15",

"beneficiaryAddress": "mfo2Vqgydzo6Xadm7zKEuvq2MgZqcfbDbJ"

}

**Sample Response (JSON Payload):**

* ***Success***

{

"status": 200,

"code": 100,

"message": "Success",

"accountDetails": [{

"type": "BTC",

"address": "mnNpnxsS7QVFHeR9XdB5RaNuorrPGPmPNA",

"balance": "0.55"

},

{

"type": "ETH",

"address": "0x64aD6313dd83Fd4D074C78D5f291bD264c317510",

"balance": "1.50"

}

]

}

* ***Error:***

{

"status": 400,

"code": 900,

"errors": [{

"code": 806,

"message": "Invalid Token Amount : -1"

}],

}

{

"status": 500,

"code": 900,

"message": "Internal Server Error",

"errors": [{

"code": 500,

"message": "Internal Server Error"

}]

}

## Account Balance

**URL:** {URL}/api/balance?access\_token=<access-token>

**Request Type** : GET

**Headers:**

* **Content-Type :** application/json

**Parameter List**:

* clientCorrelationId
* type

**Sample Request (JSON Payload):**

{

"clientCorrelationId": "4cc86e25641048788d87fad7df010604",

}

**Sample Response (JSON Payload):**

* ***Success:***

{

"status": 200,

"code": 100,

"accountDetails": [{

"type": "BTC",

"address": "mnNpnxsS7QVFHeR9XdB5RaNuorrPGPmPNA",

"balance": "0.55"

},

{

"type": "ETH",

"address": "0x64aD6313dd83Fd4D074C78D5f291bD264c317510",

"balance": "2.7"

}

]

}

***- Error:***

{

"status": 400,

"code": 900,

"message": "Internal Server Error",

"errors": [{

"code": 804,

"message": "Client doesn’t exists for CorrleationId"

}]

}

{

"status": 500,

"code": 900,

"message": "Internal Server Error",

"errors": [{

"code": 500,

"message": "Internal Server Error"

}]

}

# Response Codes

Following are the different response codes returned by Rest APIs based on internal processing and response received from blockchain:

|  |  |  |
| --- | --- | --- |
| **Response Code** | **Description** | **Remarks** |
| 100 | TRANSACTION\_SUCCESS |  |
| 801 | GENERAL\_ERROR | Internal server error |
| 802 | INVALID\_REQUEST | Request data sent to Rest API is invalid or incomplete |
| 803 | MISSING\_OR\_INVALID\_CLIENT\_CORRELATION\_ID | Company with correlation id already exists |
| 804 | CLIENT\_DOES\_NOT\_EXISTS | Client/Company with correlation Id doesn’t exists |
| 805 | CLIENT\_ALREADY\_EXISTS | Client/Company with correlation id already exists |
| 806 | INVALID\_TOKEN\_AMOUNT | Token amount is invalid |
| 807 | MISSING\_OR\_INVALID\_BENEFICIARY\_ADDRESS | Invalid beneficiary address |

# Keystore configuration

keytool -genseckey -alias ccsecretkey -keyalg aes -keysize 256 -keystore cckeystore.jceks -storetype jceks -storepass co!ncl@im -keypass co!ncl@im