CRC FICO API

REPRESENTATIONAL STATE TRANSFER (REST)

VERSION 1.0 | 12TH SEPTEMBER 2018

REVISION HISTORY

S/N	DATE	VERSION	AUTHOR	COMMENT
1	17/09/18	1.0	Ademola Adewumi	Initial release

Table of Content

1.0 INTRODUCTION

The CRC Credit Bureau FICO API was developed by CRC Credit Bureau Limited for financial institutions get information about FICO score and the reasons for the score.

1.1 OBJECTIVE

This API documentation specifies the set of instructions for successful consummation and exchange of information between member institutions and CRC FICO Score API.

1.2 INTENDED AUDIENCE

This User Guide is intended for all the Member Institutions and other user's developers that will be consuming CRC DIVS application API.

1.3 REQUIREMENTS

An institution must have an API token in order to access CRC DIVS API. This token is generated registrations and activation of accounts on the CRC FICO API Site.

1.4 SERVICE SPECIFICATION

The communication mode is through HTTP POST.
Request and response language is JSON
Service endpoint is http://172.16.7.21:8000/api/v1/post/single-batch/return-fico
Every API call requires an API token in the JSON headers.

2.0 SERVICE CONSUMMATION

2.1 SEARCH REQUEST

To make a search operation on a subject, the API expects an HTTP POST with JSON payload request to the URI /post as described below.

Sample single request:


```
[
        {
                "bvn": "22180291852",
                "phone": "",
                "acc_num": "",
                "dob": "",
                "gender": "",
                "names": ""
        },
]
```


Sample batch request:


```
[
        {
                "bvn": "22180291852",
                "phone": "",
                "acc_num": "",
                "dob": "",
                "gender": "",
                "names": ""
        },
        {
                "bvn": "22277268482",
                "phone": "",
                "acc_num": "",
                "dob": "",
                "gender": "",
                "names": ""
        },
        {
                "bvn": "22193224106",
                "phone": "",
                "acc_num": "",
                "dob": "",
                "gender": "",
                "names": ""
        },
        {
                "bvn": "22240425676",
                "phone": "d",
                "acc_num": "",
```

```
        "dob": "",
        "gender": "",
        "names": ""
    },
    {
        "bvn": "22346082618",
        "phone": "d",
        "acc_num": "",
        "dob": "",
        "gender": "",
        "names": ""
    },
    {
        "bvn": "",
        "phone": "234009057648887",
        "acc_num": "",
        "dob": "",
        "gender": "",
        "names": ""
    },
    {
        "bvn": "1112016033363000",
        "phone": "f",
        "acc_num": "",
        "dob": "",
        "gender": "",
        "names": ""
    },
    {
        "bvn": "111201603336",
        "phone": "",
        "acc_num": "",
        "dob": "",
        "gender": "",
        "names": ""
    }
]
```


ENDPOINT URL

http://172.16.7.21:8000/api/v1/post/single-batch/return-fico

PAYLOAD

Headers:

1. Accept: application/json
2. Authorization: Bearer .$token
   //Concatenate the word "Bearer " to the access token generated.

Body:

1. email: you-email@your-company.com
   //The email must be used during registration on CRC FICO API website
2. json_array: payload of the request
   // An array of the accounts to be checked.

SECURITY

Generating Access Tokens

After registering on our site, go ahead and create a client. Copy the client id and the callback url which was specified during registration. It will be used to generate access token as shown below

Make a redirect request to 'http://172.16.7.21:8000/oauth/authorize?'

Appending the credentials of the user at the end.

```
Route::get('/redirect', function () {
    $query = http_build_query([
        'client_id' => 'client-id',
        'redirect_uri' => 'http://example.com/callback',
        'response_type' => 'code',
        'scope' => '',
    ]);

    return redirect('http://your-app.com/oauth/authorize?'.$query);
});
```

If the user approves the authorization request, they will be redirected back to the consuming application. The consumer should then issue a POST request to your application to request an access token. The request should include the authorization code that was issued by your application when the user approved the authorization request. In this example, we'll use the Guzzle HTTP library to make the POST request:

```
Route::get('/callback', function (Request $request) {
    $http = new GuzzleHttp\Client;
    $response = $http->post('http://172.16.7.21:8000/oauth/token', [
        'form_params' => [
            'grant_type' => 'authorization_code',
            'client_id' => 'client-id',
            'client_secret' => 'client-secret',
            'redirect_uri' => 'http://example.com/callback',
            'code' => $request->code,
        ],
    ]);

    return json_decode((string) $response->getBody(), true);
});
```

Passing The Access Token

When calling routes that are protected by Passport, your application's API consumers should specify their access token as a Bearer token in the Authorization header of their request. For example, when using the Guzzle HTTP library:

```
$response = $client->request('POST', "http://172.16.7.21:8000/api/user', [
    'headers' => [
        'Accept' => 'application/json',
        'Authorization' => 'Bearer '.$accessToken,
    ],
]);
```

Kindly see below the procedure of using a client simulator like **postman** to test the endpoint.

Please note that the email address must have been registered and activated on the App.

After logging in, Click on create Access token and copy the token generated to a secured location. **Do not disclose the access token to a third party.**

Endpoint url: http://172.16.7.21:8000/api/v1/post/single-batch/return-fico
inside headers

Accept: application/json
Authorization: Bearer '.$token

inside body
email: oseuntotosho@gmail.com

json_array: