API Documentation

API TYPE:

pbVerify KYC

CALL FUNCTION:
pbverify-kyc

Release date: 22 October 2018

REVISION

Version 1.2
13 March 2018

phone: 011 516 9400 | eMAIL: seanb@pbsa.co.za | www.pbverify.co.za

Document owner: **PBSA (Pty) Ltd**

Technical contact: **Philip Csaplar**

Authorised by: **Leon van der Merwe**

Contents

Signed off by:

X_____

This document is signed with an advanced electronic signature which protects the integrity of the document and the identity of the signer.

Please ensure that the signature is VALID (content is unchanged) by clicking on the signature before using this document.

This document is tamper-evident and may not be changed or altered in any way.

1. Introduction

JSON
(JavaScript Object Notation) is a text-based, human-readable data interchange format used for representing simple data structures and objects in Web browser-based code. **JSON** is also sometimes used in desktop and server-side programming environments.

REST
(Representational State Transfer) is an architectural style, and an approach to communications that is often used in the development of Web services. The use of REST is often preferred over the more heavyweight SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use over the Internet. The SOAP approach requires writing or using a provided server program (to serve data) and a client program (to request data).

This document serves as technical guide to allow 3rd parties to access the pbVerify API Service by means of RESTful web service through a secure HTTPS tunnel.

Note that this is not a free service and all transactions must be considered billable. To activate the service please contact us for more information on 011 516 9400 and ask to speak with a pbVerify consultant.

2. Service Overview

pbVerify KYC API

Function: pbverfiy-kyc

This API call gives businesses that are required to FICA their customers, the ability to digitally compare inputted ID and Address data against the SACRRA (South African Credit Risk Reporting Association) database, using a rule set of match criteria to match against the member's database. The API will also display the names of the SACRRA member where that address appears. If the address does not appear according to the match criteria KYC will fail using an exception rule.

The information required to perform a KYC check.

1. Consumer Details
    a. ID Number – South African ID Number
    b. Firstname – Consumers first name
    c. Second name – Consumers second name
    d. Surname – Consumers Surname
    e. Enquiry reason – FICA|Tracing
    f. Your Reference – Customer purpose
2. Match Criteria – Use to match your inputted info
    a. Street – Street number and Name
    b. Suburb
    c. City
    d. Province
    e. Postal Code
    f. Date Range - This can be the number 1 to 36 - meaning if 3, 3 months from today's date etc..
    g. Minimum Match - Number from 1 to 10, this is how many time you want your address to match a SACRRA Address

3. RESTful Service

This service is a standard RESTful Web service via an HTTPS tunnel. The section below covers the connections strings and the methods exposed via the service.

RESTful CONNECTION	**POST**
URL	**https://www.veriid.com/PBVerify/webservice/pbverify-kyc**
Expected Variables	**memberkey, password, consumer_details, match_criteria**

EXPOSED FUNCTIONS

FUNCTION	PARAMETER	TYPE	NOTES	RETURN
pbverify-kyc	**memberkey**	**String**	**Authentication Username**	**JSON OUTPUT**
	password	**String**	**Authentication Password**	
	consumer_details	**Array**	**Array of consumer details**	
	Match_criteria	**Array**	**Array of match criteria**	

4. RESTful JSON Response

pbverify-kyc - Failure

{
"Status": "Failure",
"Error": "Your Email Address or Password was blank, please try again",
“ErrorCode” : “E07”
}

pbverify-kyc – Success


```
{
        "Status": "Success",
        "KYC_Response": {
                "ConsumerDetail": {
                        "DisplayText": "Consumer Detail",
                        "Initials": "J",
                        "FirstName": "JUST",
                        "Surname": "GOOFY",
                        "IDNo": "7905015389079",
                },
                "SuccessfulAddressMatches": [{
                        "Address": "TRICELIA ST 18 WEST ACRES NELSPRUIT 1201",
                        "SubscriberName": "Telkom",
                        "LastUpdatedDate": "2017-05-25"
                },
                {
                        "Address": "TRICELIA ST 18 WEST ACRES NELSPRUIT 1201",
                        "SubscriberName": "Telkom",
                        "LastUpdatedDate": "2017-05-25"
                }],
                "MinimumMatchCriteria": {
                        "matchDateRange": "12 months",
                        "matchFound": 2,
                        "matchRequested": "2",
                        "matchSuccessful": "Yes"
                }
        }
}
```


5. PHP Client Connection

```
<?php

   ini_set('display_errors', 1);
   ini_set('display_startup_errors', 1);
   error_reporting(E_ALL);

   function getCall($memberkey, $password, $consumer_details, $match_criteria){

      $url = "https://www.veriid.com/PBVerify/webservice/pbverify-kyc";
      $data = array('memberkey' => $memberkey, 'password' => $password, 'consumer_details' => $consumer_details,
'match_criteria' => $match_criteria);


      $ch = curl_init();
      curl_setopt($ch, CURLOPT_URL, $url);
      curl_setopt($ch, CURLOPT_POST, count($data));
      curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));
      curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
      curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
      $output = curl_exec($ch);
      $info = curl_getinfo($ch);
      $error = curl_error($ch);
      curl_close($ch);

      echo "<pre>" .print_r($output, true) . "</pre>";

   }

   $consumer_details = array(
         "idNumber" => "7905565312088", // ID Number of person doing search for
        "firstname" => "Just", // First name
         "secondname" => "Very", // Second name
         "surname" => "Goofy", // Surname
         "enquiryReason" => "FICA", // FICA | Tracing |
         "yourReference" => "465465",
   );

   $match_criteria = array(
         "street" => "18A Tricelia”,
         "suburb" => “West Acres”,
         "city" => “White River”,
         "province" => “Mpumalnaga”,
         "pocode" => “1200”,
         "dateRange" => "12",
         "minimumMatch" => "2"
   );

   getCall("YOUR MEMBERKEY GOES HERE", "YOUR PASSWORD GOES HERE", $consumer_details, $match_criteria);
```

6. Management Console

http://www.veriid.com/API_Manager

- Use Username and Password Provide by Pitney Bowes to Login
- System can be used to maintain Company Information
- Quick view of last 25 live API calls
- Reporting to view between dates of API calls
- Check the status of each transaction with error codes.

7. Error Codes

- E01 – Invalid ID Number, please check ID Number and try again.
- E02 – Missing or empty from Consumer Details.
- E03 – Missing or empty from Match Criteria.
- E04 – Search Disabled.
- E05 – Insufficient funds. (COD Clients)
- E06 – No Response from SACRA members.
- E07 – Authentication error, please check your memberkey and password and try again
- E08 – First Name does not match.
- E09 – Second name does not match.
- E10 – Surname does not match