IRAS API SERVICES INTERFACE SPECIFICATIONS

Stamp Certificate Authenticity Check

Last updated on: 6 Apr 2018
Version No.: 1.0

Table of Contents

1. Introduction 3
2. Registration at API Portal 3
3. API Services 3
   3.1 General Information 4
      3.1.1 Production Usage 4
      3.1.2 Common Interface Information 4
      3.1.3 Common Response Payload 4
   3.2 Stamp Certificate Authenticity Check 5
      3.2.1 Request Payload 5
      3.2.2 Response Payload 6
   3.3 Sandbox Testing 8
      3.3.1 Stamp Certificate Authenticity Check 8
4. Sample Code (C#) 13

1. Introduction

The Inland Revenue Authority of Singapore (IRAS) provides application programming interface (API) services to allow application developers to submit and retrieve tax related matters using HTTP requests. Most of the APIs will be in the form of a JSON web service which reduces client/server coupling and thus enabling easier integration between IRAS' service with external developers.

There will be a variety of services available in due time. While some services require a simple GET, others may be secured and require credentials that can be passed via HTTP header parameters which are as follows :

X-IBM-Client-Id	String containing the client ID of the application invoking IRAS API. This value will be provided to the application vendor by IRAS. E.g. a1234b5c-1234-abcd-efgh-a1234b5cdef
X-IBM-Client-Secret	String containing the client secret of the application invoking IRAS API. This value will be provided to the application vendor by IRAS. E.g. a12345bC67e8fG9a12345bC67e8fG9a12345bC67e8fG9

This document serves to help developers consume the API services provided by IRAS.

2. Registration at API Portal

Application developers are required to register for an account at https://apisandbox.iras.gov.sg/ to subscribe to IRAS API services for Sandbox Testing and at https://apiservices.iras.gov.sg/ for Production.

A computer-generated email will be sent to the subscriber's email account for account activation of the API Portal.

3. API Services

IRAS will provide several API services for the authenticity of stamp certificate. The following sections describe the request and response for each of the services.

The table below shows the list of Stamp Certificate Authenticity Check API services currently available in IRAS.

S/No	Name of API Services	Description	URL
1	Stamp Certificate Authenticity Check	This API can be used to check the authenticity of a stamp certificate.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/SD/SCAuthenticity
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/SD/SCAuthenticity

3.1 General Information

3.1.1 Production Usage

Approval is **NOT** required to use this service.

The following parameters must be populated in the HTTP header:

X-IBM-Client-Id	String containing the client ID of the application invoking IRAS API. This value will be provided to the application vendor by IRAS.
X-IBM-Client-Secret	String containing the client secret of the application invoking IRAS API. This value will be provided to the application vendor by IRAS.

3.1.2 Common Interface Information

- JSON is case sensitive by specifications.
- All date strings are to be represented in compliance to the ISO-8601 standard.
- All string fields are subject to validation of the following acceptable characters that is allowed (in red):
  - [a-zA-Z0-9‘@#()\-,./&+_] (**Note**: whitespace is included)
- All properties follow the camel-case convention.
- Unless stated as optional, all JSON object properties must be specified.
- Unless otherwise specified, all JSON services are invoked using HTTP verb POST.
- All input data format are as specified like the following:

3.1.3 Common Response Payload

All response payloads share the following common fields:

data	Object	The data property will be populated differently based on the API that is being invoked.
returnCode	Integer	**10** : Success - The request was successfully processed.

		20 : Warning - The request was successfully processed. However, there are non-fatal issues. Please refer to the “info” object for diagnostic information. **30** : Failure – The request was not processed. Refer to “info” object for error information.
info	Object	This complex object holds any diagnostic information that will allow developers to debug their failed requests.
info.message	String	Diagnostic message in the event of warning or error.
Info.messageCode	Integer	Integer code signifying the type of error or warning. **850300 :** Request object is null – The incoming JSON request is null. **850301** : Arguments error – There is an error with one of the arguments provided. **850302** : Generic error – There is an exception within the service. **850303** : Service is inactive. **850304** : Service is not authorized for usage based on the provided credentials. **850305** : Invalid test user – The input fields provided are not valid for sandbox testing.
info.fieldInfoList	Array	An array for FieldInfo objects.
info.fieldInfoList.field	String	Name of the field that resulted in a warning / error.
Info.fieldInfoList.message	String	Diagnostic message provided to aid consumer’s developers.

3.2 Stamp Certificate Authenticity Check

3.2.1 Request Payload

docRefNo	Number	Document reference number of the document.
stampCertRef	String	Stamp certificate reference.

Sample JSON request payload

```
{
  "docRefNo": 2017092900013,
  "stampCertRef": "061211-00MA4-1-001314317"
}
```

3.2.2 Response Payload

data	Object	The object payload containing the information for the queried document reference number and stamp certificate reference.
data.certType	String	Stamp certificate type.
data.assmtType	String	The assessment type of the stamp certificate.
data.stampCertRef	String	The stamp certificate reference
data.stampCertIssueDate	String	The date the stamp certificate is issued.
data.docRefNo	String	Document reference number of the document.
data.docVerNo	String	Version number of the document.
data.appRefNo	String	Applicant's reference number.
data.docDescription	String	Description of the document stamped.
data.dateOfDoc	String	The date of the document.
data.sdAmount	String	Stamp duty amount. Not applicable for sale and purchase.
data.penalty	String	Penalty.
data.buyerSD	String	Buyer's stamp duty. Only applicable for sale and purchase.
data.addBuyerSD	String	Additional buyer's stamp duty. Only applicable to sale and purchase.
data.duplicate	String	Duplicate(s)
data.adjudicationFee	String	Adjudication fee
data.valuationFee	String	Valuation fee
data.fines	String	Fines
data.totalAmtPayable	String	Total amount payable.
data.propertyList	Array	A listing of the line items under property.
data.propertyList.blkHseNo	String	Address of the property.
data.propertyList.street	String	Address of the property.
data.propertyList.unitLevel	String	Address of the property.
data.propertyList.postalCode	String	Address of the property.
data.vacantLandList	Array	A listing of the line items under vacant land.
data.vacantLandList.mkTSNo	String	MK/ TS number.
data.vacantLandList.lotNo	String	Lot number.
data.vacantLandList.plPTParcelNo	String	PL/ PT parcel number
data.vacantLandList.streetName	String	Street name of the vacant land.
data.stockSharesList	Array	A listing of the line items under stocks and shares.
data.stockSharesList.nameOfCompany	String	Name of company.

data.stockSharesList.entityType	String	The entity type of the company.
data.stockSharesList.entityID	String	ID entity of the company
data.stockSharesList.noStocksShares	String	Number of stocks and shares.
data.securities	Array	A listing of the line items under securities.
returnCode	As per Section 3.1.3	
info		
info.message		
info.messageCode		
info.fieldInfoList		
info.fieldInfoList.field		
info.fieldInfoList.message		
info.fieldInfoList.recordID		

Sample success JSON response payload


```
{
   "data":    {
      "addBuyerSD": "0",
      "adjudicationFee": "0",
      "assmtType": "Original",
      "certType": "Original",
      "dateOfDoc": "2017-09-01",
      "docDescription": " Equitable Mortgage (Ad valorem)",
      "docRefNo": "2017092900013",
      "docVerNo": "1.0",
      "fines": "0",
      "penalty": "10",
      "sdAmount": "34",
      "stampCertIssueDate": "2017-09-29",
      "stampCertRef": "061211-00MA4-1-001314317",
      "totalAmtPayable": "44",
      "valuationFee": "0",
      "propertyList": [      {
         "unitLevel": "#04-111",
         "street": "YISHUN ROAD",
         "postalCode": "760758",
         "blkHseNo": "386"
      }],
      "stockSharesList": [
            {
                "entityID": "12121",
                "entityType": "OTHERS",
                "nameOfCompany": "ABC",
                "noStocksShares": "100"
            }
        ],
      "vacantLandList": [],
      "appRefNo": "XY1234",
        "securities": [
            "1212",
            "11212"
```

```
        ]
      "returnCode": "10",
      "info": {"fieldInfoList": []}
}
```

Sample error JSON response payload

```
{
   "returnCode": "30",
   "info":    {
      "message": "Arguments Error",
      "messageCode": "850301",
      "fieldInfoList":       [
                  {
            "field": "docRefNo",
            "message": "Value cannot be null"
         },
                  {
            "field": "stampCertRef",
            "message": "Value cannot be null"
         }
      ]
   },
   "data":    {
      "propertyList": [],
      "stockSharesList": [],
      "vacantLandList": [],
      "securities": []
   }
}
```

3.3 Sandbox Testing

As explained in section 2, developers can first create an account in the Sandbox environment to make API calls to our Sandbox URL. This allows the developers to mimic the characteristics of our production environment and create a simulated response from our API.

3.3.1 Stamp Certificate Authenticity Check

Input	Expected Output
{ "docRefNo": 2017080200001, "stampCertRef": "221161-76SA1-1-001655717" }	{ "data": { "addBuyerSD":"0", "adjudicationFee":"0", "assmtType":"Revised Amended", "buyerSD":"21300", "certType":"Original", "docDescription":"Sale and Purchase Agreement (Ad valorem)", "docRefNo":"2017080200001", "docVerNo":"2.0", "fines":"0",

Input	Expected Output

	"penalty":"81",
	"stampCertIssueDate":"2017-08-02",
	"stampCertRef":"221161-76SA1-1-001655717",
	"totalAmtPayable":"21381",
	"valuationFee":"0",
	"propertyList":[
	{
	"street":"WOODLANDS DRIVE 53",
	"postalCode":"730555",
	"blkHseNo":"555"
	}
	],
	"stockSharesList":[],
	"vacantLandList":[],
	"securities":[]
	},
	"returnCode":"10",
	info":
	{
	"fieldInfoList":[]
	}
	}

Input	Expected Output

{
"docRefNo": 2017090500006,
"stampCertRef": "011043-36SR2-1-001665517"
}
	{
	"data":
	{
	"addBuyerSD":"0",
	"adjudicationFee":"50",
	"assmtType":"Relief",
	"certType":"Original",
	"dateOfDoc":"2017-09-01",
	"docDescription":"Transfer",
	"docRefNo":"2017090500006",
	"docVerNo":"2.0",
	"fines":"0",
	"penalty":"0",
	"sdAmount":"0",
	"stampCertIssueDate":"2017-09-05",
	"stampCertRef":"011043-36SR2-1-001665517",
	"totalAmtPayable":"60",
	"valuationFee":"10",
	"propertyList":[],
	"stockSharesList":[
	{
	"entityID":"",
	"entityType":"",
	"nameOfCompany":"TEST COMPANY",
	"noStocksShares":"1"
	}
	],
	"vacantLandList":[],
	"securities":[]
	},
	"returnCode":"10",
	"info":
	{
	"fieldInfoList":[]
	}
	}

Input	Expected Output

{
"docRefNo": 2011062000004,
"stampCertRef": "200200-11SH1-1-000555555"
}
	{
	"returnCode":"30",
	"info":
	{
	"message":"Document Reference Number and Stamp Certificate Reference do not exist in the system.",
	"messageCode":"850301",
	"fieldInfoList":[]
	},
	"data":
	{
	"propertyList":[],
	"stockSharesList":[],
	"vacantLandList":[],
	"securities":[]
	}
	}

{
"docRefNo" : 2017112800004,
"stampCertRef" : "200200-11SH1-1-000555555"
}
	{
	"returnCode":"30",
	"info":
	{
	"message":"Document Reference Number does not match Stamp Certificate Reference.",
	"messageCode":"850301",
	"fieldInfoList":[]
	},
	"data":
	{
	"propertyList":[],
	"stockSharesList":[],
	"vacantLandList":[],
	"securities":[]
	}
	}

Input	Expected Output

{
}
	{
	"returnCode":"30",
	"info":
	{
	"message":"Request object is null",
	"messageCode":"850300",
	"fieldInfoList":[]
	},
	"data":
	{
	"propertyList":[],
	"stockSharesList":[],
	"vacantLandList":[],
	"securities":[]
	}
	}

{
"stampCertRef" : "220160-01SA1-1-001764917"
}
	{
	"returnCode":"30",
	"info":{
	"message":"Arguments Error",
	"messageCode":"850301",
	"fieldInfoList":[
	{
	"field":"docRefNo",
	"message":"Value cannot be null"
	}]
	},
	"data":
	{
	"propertyList":[],
	"stockSharesList":[],
	"vacantLandList":[],
	"securities":[]
	}
	}

Input	Expected Output

<pre>{
"docRefNo" : 2017112800004
}</pre>
	<pre>{
	"returnCode":"30",
	"info":{
	"message":"Arguments Error",
	"messageCode":"850301",
	"fieldInfoList":[{
	"field":"stampCertRef",
	"message":"Value cannot be null"}]
	},
	"data":{
	"propertyList":[],
	"stockSharesList":[],
	"vacantLandList":[],
	"securities":[]
	}
	}</pre>

4. Sample Code (C#)

```
using System;
using System.Net;
using System.IO;
using System.Text;

// jsonData – contains data from Section 3.1.1 of this document
public static void callWebAPI(string jsonData, string url)
{
        try
        {
            var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
            httpWebRequest.ContentType = "application/json;";
            httpWebRequest.Method = "POST";

            //Step 1: Enter the Client-Id given by IRAS
            httpWebRequest.Headers["Client-Id"] = "{YOUR_CLIENT_ID}";
            //Step 2: Enter the Client-Secret given by IRAS
            httpWebRequest.Headers["Client-Secret"] ="{YOUR_CLIENT_SECRET}";

            // Step 3: Call API using POST
            using (var streamWriter = new StreamWriter(httpWebRequest.GetRequestStream()))
            {
                streamWriter.Write(jsonData);
                streamWriter.Flush();
                streamWriter.Close();
            }

            // Step 3a: Output response
            var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
            using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
            {
                var result = streamReader.ReadToEnd();
                //print the received reponse
                Console.WriteLine(result);
            }
        }
```

```
        catch (WebException e)
        {
            if (!string.IsNullOrEmpty(e.Message))
            {
                // Step 3b: Print general errors
                Console.WriteLine("Exception - ");
                Console.WriteLine(e.Message);
            }

            if (e.Response != null)
            {
                // Step 3c: Print Output response exception
                Stream receiveStream = e.Response.GetResponseStream();
                StreamReader readStream = new StreamReader(receiveStream, Encoding.UTF8);
                // print the error received from Server
                Console.WriteLine("Response error received - ");
                Console.WriteLine(readStream.ReadToEnd());
            }
        }
}
```

The information provided is intended for better general understanding and is not intended to comprehensively address all possible issues that may arise. The contents are correct as at 6 Mar 2018 and are provided on an "as is" basis without warranties of any kind. IRAS shall not be liable for any damages, expenses, costs or loss of any kind however caused as a result of, or in connection with your use of this document.

While every effort has been made to ensure that the above information is consistent with existing policies and practice, should there be any changes, IRAS reserves the right to vary our position accordingly.