IRAS API SERVICES INTERFACE SPECIFICATIONS

Retrieve Property Tax Consolidated Statement

Last updated on: 25 Apr 2018

Version No.: 1.0

Table of Contents

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 easy 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 create a developer's account at https://apisandbox.iras.gov.sg/ and subscribe to IRAS API services for Sandbox Testing,
- To create a developer's account at https://apiservices.iras.gov.sg/ and subscribe to IRAS API services for Production usage.

Note: In order for IRAS to identify your API subscriptions, please enter your details as follows when creating an account:

- "First name" field: To enter "Name of organisation"
- "Last name" field: To enter "Tax Reference No. of organisation"

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

3. API Services

The following sections describe the request and the response of the API service.

S/No	Name of API Services	Description	URL
1	Property Consolidated Statement	This API can be used to submit Property Consolidated Statement	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/PropertyConsolidatedStatement/retrieve
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/PropertyConsolidatedStatement/retrieve

3.1 General Information

Each retrieval of property tax consolidated statement information cannot be more than 2 MB in data size.

3.1.1 Production Usage

Approval is 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:

Data Format Specification

Data Type and Size	Description	Example
String(12)	A string containing maximum 12 characters.	“S1234567Z”
Number(4)	A numeric value containing maximum 4 whole numbers.	1990

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 Retrieve Property Consolidated Statement

3.2.1 Request Payload

refNo	String(30)	Reference number from successful submission of “Corporate Rental Submission” API. Compulsory.
propertyTaxRef	String(8)	One of the Property Tax Reference number within the consolidated statement. Compulsory.

Sample JSON request payload

```
{
  "propertyTaxRef": "0200320A",
  "refNo": "PTSTMT000001"
}
```

3.2.2 Response Payload

data	Object	The object payload containing the date of successful retrieval.
data.propertyTaxRef	String(8)	Tax reference number of the property
data.propertyDesc	String(100)	Property description
data.annualValue	String(20)	Annual value of the property (in 2 decimal point)
data.dateEffAV	String(8)	Effective date of the annual value (in YYYYMMDD)
data.dateNotice	String(8)	Date of the notice (in YYYYMMDD)
data.dinNotice	String(20)	Document Identification Number for the notice
data.taxBalAmt	String(20)	Tax balance of the property (in 2 decimal point)
data.indicatorFollowingYr	String(5)	Indicator for the following year
data.paymentMode	String(10)	Mode of payment
data.statusRevision	String(20)	Revision status
returnCode	As per Section 3.1.3	
info		
info.message		
info.messageCode		
info.fieldInfoList		
info.fieldInfoList.field		
info.fieldInfoList.message		

Sample success JSON response payload


```
{
"returnCode":"10",
 "data":[{
"annualValue":"3500.02",
"dateEffAV":"20160101",
"dateNotice":"20161018",
"dinNotice":"106-22-040423088-0-3",
"indicatorFollowingYr":"No",
"paymentMode":"Non-GIRO",
"propertyDesc":"ABC LANE #111-02",
"propertyTaxRef":"0200320A",
"statusRevision":"",
"taxBalAmt":"4892.05"
}],
"info":{"fieldInfoList":[]}
}
```


Sample error JSON response payload


```
{
  "returnCode": "30",
  "info": {
    "messageCode": 850301,
    "message": "Arguments Error",
    "fieldInfoList": [
      {
        "field": "propertyTaxRef",
        "message": "Invalid ID",
      }
    ]
  }
}
```


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.

Input	Expected Output

{
"propertyTaxRef": "0200320A",
"refNo": "PTSTMT000001"
}
	{
	"returnCode":"10",
	"data":[{
	"annualValue":"3500.02",
	"dateEffAV":"20160101",
	"dateNotice":"20161018",
	"dinNotice":"106-22-040423088-0-3",
	"indicatorFollowingYr":"No",
	"paymentMode":"Non-GIRO",
	"propertyDesc":"ABC LANE #111-02",

	"propertyTaxRef":"0200320A", "statusRevision":"", "taxBalAmt":"4892.05" }], "info":{"fieldInfoList":[]} }

{
"propertyTaxRef": "0200322G",
"refNo": "PTSTMT000002"
}
	{
	"returnCode":"10",
	"data":[{
	"annualValue":"3500.02",
	"dateEffAV":"20160101",
	"dateNotice":"20161018",
	"dinNotice":"106-22-040423088-0-3",
	"indicatorFollowingYr":"No",
	"paymentMode":"Non-GIRO",
	"propertyDesc":"CDE STREET #10-02",
	"propertyTaxRef":"0200322G",
	"statusRevision":"",
	"taxBalAmt":"4892.05"
	}],
	"info":{"fieldInfoList":[]}
	}

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 03 Oct 2017 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.