INLAND REVENUE
AUTHORITY
OF SINGAPORE

IRAS API SERVICES
INTERFACE SPECIFICATIONS

Property Tax Services

Last updated on : 31 January 2017

Version No: 1.0

Disclaimers: *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 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.*

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 easier integration between IRAS' service with external developers. 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 an account at https://apiservices.iras.gov.sg/ to subscribe to IRAS API services for Production.

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

3. API Services

IRAS will provide several API services for public consumption. The following sections describe the request and response for each of the services.

The table below shows the list of Property Tax API services currently available in IRAS.

S/No	Name of API Services	Description	URL
1	Property Tax Balance Enquiry	This service enables a taxpayer to retrieve the property tax payable on his property by simply entering the property address and his or her co-owner's identification number.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/PTTaxBal/PtyTaxBalSearch
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/PTTaxBal/PtyTaxBalSearch

3.1 General Information

3.1.1 Production Usage

Approval is required to use the services.

Developers are required to test in the Sandbox environment and send the sample test data to the Authority before the Authority will grant approval for production usage .

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 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
- JSON strings are to be enclosed with double quotes (”) and NOT single quotes (‘)

3.1.3 Common Request Payload

All request payloads share the following common field:

clientID	String	This has to match the client id that is passed in via the HTTP headers

3.1.4 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. **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. **400348** : Property record not found. **400349** : You do not have access to this property. **10454** : Property is not in Valuation List.
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 Property Tax Balance Enquiry Service

3.2.1 Request Payload

clientID	As per section 3.1.3	
criteria	String	Search Criteria: 1 = **Search by building address** (*blkHouseNo, streetName, ownerTaxRefID are required fields*) 2 = **Search by property tax reference number** (*pptyTaxRefNo is a required field*) 3 = **Search by postal code** (*postalCode, ownerTaxRefID are required fields*) *Note: The rest of the fields not mentioned are optional.*
blkHouseNo	String	Blk/House Number
streetName	String	Street Name
postalCode	String	Postal Code
storeyNo	String	Storey Number
unitNo	String	Unit Number
ownerTaxRefID	String	Owner Tax Reference ID
pptyTaxRefNo	String	Property Tax Reference Number

Sample JSON request payload

```
{
  "clientID": "{YOUR_CLIENT_ID}",
  "pptyTaxRefNo": "4143149E",
  "postalCode": "",
  "criteria": "2",
  "blkHouseNo": "",
  "streetName": "",
  "storeyNo": "",
  "unitNo": "",
  "ownerTaxRefID": ""
}
```

3.2.2 Response Payload

data	Object	The object payload containing the property tax balance information
data.propertyDescription	String	The description of the property.
data.propertyTaxReferenceNo	String	The property tax reference number.
data.outstandingBalance	Number	The outstanding property tax balance. *Note: **data.outstandingBalance** <= 0 indicates that there is no outstanding balance.*
data.paymentByGiro	String	A “Yes” or “No” indicator of whether payment is by GIRO.
returnCode	As per section 3.1.4	
info		

info.message	
Info.messageCode	
info.fieldInfoList	
info.fieldInfoList.field	
Info.fieldInfoList.message	

Sample success JSON response payload

```
{
  "returnCode": 10,
  "info": {
    "fieldInfoList": []
  },
  "data": {
    "outstandingBalance": "2844.00",
    "paymentByGiro": "No",
    "propertyDescription": "",
    "propertyTaxReferenceNo": "4143149E"
  }
}
```

Sample error JSON response payload

```
{
  "returnCode": 30,
  "info": {
    "fieldInfoList": [
      {
        "field": "PptyTaxRefNo",
        "message": "Value is not valid"
      }
    ],
    "message": "Arguments Error",
    "messageCode": 850301
  }
}
```

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 Property Tax Balance Enquiry Service

Input	Expected Output

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "",
"postalCode": "",
"criteria": "1",
"blkHouseNo": "151B",
"streetName": "KINGS RD",
"storeyNo": "09",
"unitNo": "05",
"ownerTaxRefID": "T9100002J"
}
	{
	"returnCode": 10,
	"info": {
	"fieldInfoList": []
	},
	"data": {
	"outstandingBalance": "2844.00",
	"paymentByGiro": "Yes",
	"propertyDescription": "151B KINGS RD #09-05",
	"propertyTaxReferenceNo": "0200320A"
	}
	}

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "",
"postalCode": "",
"criteria": "1",
"blkHouseNo": "151B",
"streetName": "KINGS RD",
"storeyNo": "09",
"unitNo": "08",
"ownerTaxRefID": "T9100004G"
}
	{
	"returnCode": 10,
	"info": {
	"fieldInfoList": []
	},
	"data": {
	"outstandingBalance": "-500.00",
	"paymentByGiro": "Yes",
	"propertyDescription": "151B KINGS RD #09-08",
	"propertyTaxReferenceNo": " 0200323U"
	}
	}

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "",
"postalCode": "268159",
"criteria": "3",
"blkHouseNo": "",
"streetName": "",
"storeyNo": "09",
"unitNo": "07",
"ownerTaxRefID": "T9100002J"
}
	{
	"returnCode": 20,
	"info": {
	"fieldInfoList": [],
	"message": "Property record not found",
	"messageCode": 400348
	}
	}

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "",
"postalCode": "",
"criteria": "1",
"blkHouseNo": "151B",
"streetName": "KINGS RD",
"storeyNo": "09",
"unitNo": "05",
"ownerTaxRefID": "T9100003I"
}
	{
	"returnCode": 20,
	"info": {
	"fieldInfoList": [],
	"message": "You do not have access to this property",
	"messageCode": 400349
	}
	}

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "",
"postalCode": "268159",
"criteria": "3",
	{
	"returnCode": 10,
	"info": {
	"fieldInfoList": []
	},

"blkHouseNo": "",
"streetName": "",
"storeyNo": "09",
"unitNo": "05",
"ownerTaxRefID": "T9100002J"
}
	"data": {
	"outstandingBalance": "2844.00",
	"paymentByGiro": "Yes",
	"propertyDescription": "151B KINGS RD #09-05",
	"propertyTaxReferenceNo": "0200320A"
	}
	}

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "0200320A",
"postalCode": "",
"criteria": "2",
"blkHouseNo": "",
"streetName": "",
"storeyNo": "",
"unitNo": "",
"ownerTaxRefID": ""
}
	{
	"returnCode": 10,
	"info": {
	"fieldInfoList": []
	},
	"data": {
	"outstandingBalance": "2844.00",
	"paymentByGiro": "Yes",
	"propertyDescription": "",
	"propertyTaxReferenceNo": "0200320A"
	}
	}

{
"clientID": "YOUR_CLIENT_ID",
"pptyTaxRefNo": "0200321P",
"postalCode": "",
"criteria": "2",
"blkHouseNo": "",
"streetName": "",
"storeyNo": "",
"unitNo": "",
"ownerTaxRefID": ""
}
	{
	"returnCode": 20,
	"info": {
	"fieldInfoList": [],
	"message": "Property is not in Valuation List",
	"messageCode": 10454
	}
	}

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["X-IBM-Client-Id "] = "{YOUR_CLIENT_ID}";
           //Step 2: Enter the Client-Secret given by IRAS
           httpWebRequest.Headers["X-IBM-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());
           }
        }
}
```