



IRAS API SERVICES INTERFACE SPECIFICATIONS

Individual Income 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.

© Inland Revenue Authority of Singapore

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, including photocopying and recording without the written permission of the copyright holder, application for which should be addressed to the publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature.

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 Request Payload	4
	3.1.4 Common Response Payload.....	4
3.2	Income Expense Listing Service	6
	3.2.1 Request Payload.....	6
	3.2.2 Response Payload.....	6
3.3	Sandbox Testing.....	8
	3.3.1 Income Expense Listing Service.....	8
	Sample Code (C#).....	11

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 IIT API services currently available in IRAS.

S/No	Name of API Services	Description	URL
1	Income Expense Listing	This service allows users to retrieve the list of allowable and disallowable business expenses categorised by the profile of the taxpayers for calculating the adjusted profit/loss.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/ESubmission/IncExpListing For Production Usage: https://apiservices.iras.gov.sg/iras/prod/ESubmission/IncExpListing

3.1 General Information

3.1.1 Production Usage

Approval is **NOT** required to use the services.

Developers are encouraged to participate in the Sandbox Testing prior to use in Production.

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. 400033 : No match data found.
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 Income Expense Listing Service

3.2.1 Request Payload

clientID	As per section 3.1.3	
effectiveYA	Number	The effective Year of Assessment
profile	String	Specific profile name for filtering purposes <i>List of Profile Values</i> <ul style="list-style-type: none"> • Tutor • Commission agent • Taxi Driver <p><i>Note: Profile is optional. If profile is left empty, the entire listing of income and expenses will be retrieved</i></p>

Sample JSON request payload

```
{
  "clientID": "{YOUR_CLIENT_ID}",
  "profile": "",
  "effectiveYA": 2016
}
```

3.2.2 Response Payload

data	Object	The object payload containing the information for the line items queried for a specific Year of Assessment.
data.effectiveYA	Number	The effective Year of Assessment.
data.profile	String	The profile name that groups the set of data.incomeList and data.expenseList .
data.incomeList	Array	A listing of the line items under Income
data.incomeList.name	String	The name of the income line item
data.incomeList.allowable	String	"Y" = Allowable Income "N" = Disallowable Income
data.incomeList.description	String	A short description that describes this line item.
data.expenseList	Array	A listing of the line items under Expense
data.expenseList.name	String	The name of the expense line item
data.expenseList.allowable	String	"Y" = Allowable Expense "N" = Disallowable Expense
data.expenseList.description	String	A short description that describes this line item.
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,
  "data": {
    "effectiveYA": 2016,
    "profile": "",
    "expenseList": [
      {
        "allowable": "Y",
        "description": "Rental expense incurred for renting the business premises",
        "name": "Rental charges"
      },
      {
        "allowable": "N",
        "description": "Loss on disposal in respect of fixed assets sold",
        "name": "Loss on sale of fixed assets"
      }
    ],
    "incomeList": [
      {
        "allowable": "Y",
        "description": "Gross sales",
        "name": "Revenue"
      }
    ]
  },
  "info": {
    "fieldInfoList": []
  }
}

```

Sample error JSON response payload

```

{
  "returnCode": 30,
  "data": {
    "expenseList": [],
    "incomeList": []
  },
  "info": {
    "message": "Arguments Error",
    "messageCode": 850301,
    "fieldInfoList": [
      {
        "field": "EffectiveYA",
        "message": "Value cannot be null"
      }
    ]
  }
}

```

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 using a predefined set of inputs with a predefined output. This allows the developers to mimic the characteristics of our production environment and create a simulated response from our API.

3.3.1 Income Expense Listing Service

Input	Expected Output
<pre>{ "clientID": "YOUR_CLIENT_ID", "profile": "tutor", "effectiveYA": 2016 }</pre>	<pre>{ "returnCode": 10, "data": { "effectiveYA": 2016, "profile": "tutor", "expenseList": [{ "allowable": "Y", "description": "MRT, Taxi or Bus", "name": "Public Transport expense" }, { "allowable": "N", "description": "Expenses incurred directly or in the form of reimbursement on using private hire cars / private cars", "name": "Private Car expense" }, { "allowable": "Y", "description": "Telephone charges for the business", "name": "Telephone charges for the business" }, { "allowable": "N", "description": "Telephone charges for personal use", "name": "Telephone charges for personal use" }, { "allowable": "Y", "description": "Advertising fees", "name": "Advertising cost" }], "incomeList": [{ "allowable": "Y", "description": "Gross Income received", "name": "Tuition Fees" }] } }</pre>

	<pre>] }, "info": { "fieldInfoList": [] } } </pre>
<pre> { "clientID": "YOUR_CLIENT_ID", "profile": "Commission agent", "effectiveYA": 2016 } </pre>	<pre> { "returnCode": 10, "data": { "effectiveYA": 2016, "profile": "Commission agent", "expenseList": [{ "allowable": "Y", "description": "Advertising fees", "name": "Advertising cost" }, { "allowable": "Y", "description": "Telephone charges for the business", "name": "Telephone charges for the business" }, { "allowable": "N", "description": "Telephone charges for personal use", "name": "Telephone charges for personal use" }, { "allowable": "Y", "description": "Printng of name cards", "name": "Stationeries" }, { "allowable": "Y", "description": "Postages", "name": "Postages" }, { "allowable": "Y", "description": "MRT, Taxi or Bus", "name": "Public Transport expense" }, { "allowable": "N", "description": "Expenses incurred directly or in the form of reimbursement on using private hire / private cars", "name": "Private Car expense" }, { "allowable": "Y", "description": "Entertainment expenses for business related purposes only", "name": "Entertainment" }, { "allowable": "Y", </pre>

	<pre> "description": "GiftsAndGreetingcards", "name": "Gifts and Greeting cards" }, { "allowable": "Y", "description": "Employee/Staff costs", "name": "Gross employee salary, bonus, allowances" }, { "allowable": "Y", "description": "Compulosry CPF contributions by employer", "name": "Employer CPF Contributions" }, { "allowable": "N", "description": "CPF contribution for your employees above the statutory limit", "name": "Employers' CPF contribution exceeding CPF statutory rates" }, { "allowable": "N", "description": "Exceeding the cap of 1% or up to 2% of total employees' remuneration", "name": "Medical expenses for staff exceeding the cap" }, { "allowable": "N", "description": "Expenses that are personal and private in nature are not allowable", "name": "Personal expenses" }, { "allowable": "N", "description": "Fines and penalties", "name": "Penalties/fines/late payment charges" }], "incomeList": [{ "allowable": "Y", "description": "Gross commission income received", "name": "Commmission income" }] }, "info": { "fieldInfoList": [] } } </pre>
--	---

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 httpRequest = (HttpRequest)WebRequest.Create(url);
        httpRequest.ContentType = "application/json";
        httpRequest.Method = "POST";

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

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

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

```