



INLAND REVENUE
AUTHORITY
OF SINGAPORE

IRAS API SERVICES INTERFACE SPECIFICATIONS

Organisation Search

Last updated on: 30 June 2019

Version No: 2.0

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 AIS Organisation Search API Service.....	6
3.2.1 Request Payload	6
3.2.2 Response Payload	6
3.3 Commission Search API Service	8
3.3.1 Request Payload	8
3.3.2 Response Payload	8
3.4 Sandbox Testing	10
3.4.1 AIS Organisation Search API Service.....	10
3.4.2 Commission Search API Service	11
4. Sample Code (C#).....	14

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 of the API Portal.

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 Organisation Search API services currently available in IRAS.

S/No	Name of API Services	Description	URL
1	AIS Organisation Search	This API allows you to check if the organisation is participating in the Auto-Inclusion Scheme (AIS) for Employment Income.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/ESubmission/AISOrgSearch
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/ESubmission/AISOrgSearch
2	Commission Search	This API allows you to check if the organisation is participating in the e-Submission of Commission.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/ESubmission/CommSearch
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/ESubmission/CommSearch

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.
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 AIS Organisation Search API Service

3.2.1 Request Payload

clientID	As per section 3.1.3	
organizationID	String	The tax reference number of the organisation
basisYear	Number	The year of participation in e-submission of employment income <i>Note: The API only supports for basis year from 1997 onwards i.e. Basis year must be greater than or equal to 1997, else an error "Value is not valid" will be returned.</i>

Sample JSON request payload (where Current Year = 2019)

```
{
  "clientID": "YOUR_CLIENT_ID",
  "organizationID": "200312345A",
  "basisYear": 2019
}
```

3.2.2 Response Payload

data	Object	The object payload containing the successful retrieval results.
data.organizationInAIS	String	The indicator of whether the organisation is participating in e-submission of employment income. Value can be either "Y" or "N".
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": {
    "organizationInAIS": "Y"
  },
  "info": {
    "fieldInfoList": []
  }
}
```

Sample error JSON response payload

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

3.3 Commission Search API Service

3.3.1 Request Payload

clientID	As per section 3.1.3	
organizationID	String	The tax reference number of the organisation
basisYear	Number	The year of participation in e-submission of commission <i>Note: The API only supports for basis year from 1997 onwards i.e. Basis year must be greater than or equal to 1997, else an error "Value is not valid" will be returned.</i>

Sample JSON request payload (where Current Year = 2019)

```
{
  "clientID": "YOUR_CLIENT_ID",
  "organizationID": "200312345A",
  "basisYear": 2019
}
```

3.3.2 Response Payload

data	Object	The object payload containing the successful retrieval results.
data.organizationInAIS	String	The indicator of whether the organisation is participating in e-submission of commission search. Value can be either "Y" or "N".
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": {
    "organizationInAIS": "N"
  },
  "info": {
    "fieldInfoList": []
  }
}
```

Sample error JSON response payload

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

3.4 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.

The tables below show the Input and Expected Output based on Current Year = 2019.

3.4.1 AIS Organisation Search API Service

<u>Input</u>	<u>Expected Output</u>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "208000001M", "basisYear": 2019 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "208000001M", "basisYear": 1995 }</pre>	<pre>{ "returnCode": 30, "info": { "messageCode": 850301, "message": "Arguments Error", "fieldInfoList": [{ "field": "BasisYear", "message": "Value is not valid" }] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "T16ZZ0102E", "basisYear": 2018 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "95000001M", "basisYear": 1997 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "N" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "95000001M", "basisYear": 1990 }</pre>	<pre>{ "returnCode": 30, "info": { "messageCode": 850301, "message": "Arguments Error", "fieldInfoList": [{ "field": "BasisYear", "message": "Value is not valid" }] } }</pre>

<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "A9300001E", "basisYear": 2016 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "A9300001E", "basisYear": 2021 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "482222222R", "basisYear": 2017 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "N" }, "info": { "fieldInfoList": [] } }</pre>

3.4.2 Commission Search API Service

<u>Input</u>	<u>Expected Output</u>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "T16ZZ0101j", "basisYear": 2016 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "T16ZZ0101J", "basisYear": 2017 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "T16ZZ0101J", "basisYear": 1995 }</pre>	<pre>{ "returnCode": 30, "info": { "messageCode": 850301, "message": "Arguments Error", "fieldInfoList": [{ "field": "BasisYear", "message": "Value is not valid" }] } }</pre>

IRAS API Services Interface Specifications

<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "481111111E", "basisYear": 2016 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "481111111E", "basisYear": 2017 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "N" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "95000003E", "basisYear": 2000 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "N" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "95000003E", "basisYear": 2017 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "208000003N", "basisYear": 2019 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "208000003N", "basisYear": 1997 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "N" }, "info": { "fieldInfoList": [] } }</pre>
<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "A9300004Z", "basisYear": 2016 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</pre>

IRAS API Services Interface Specifications

<pre>{ "clientID": "YOUR_CLIENT_ID", "organizationID": "A9300004Z", "basisYear": 2017 }</pre>	<pre>{ "returnCode": 10, "data": { "organizationInAIS": "Y" }, "info": { "fieldInfoList": [] } }</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["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());
        }
    }
}

```