

IRAS API SERVICES INTERFACE SPECIFICATIONS

Auto-Inclusion Scheme (AIS) For Employment Income – Submission of Employment Income Records (CorpPass)

Last updated on : 03 Nov 2020

Version No: 1.2

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.

Version Control

Version History

Revision Date	Version Number	Change Log Summary
04 Dec 2019	1.0	Baseline
01 July 2020	1.1	Added Network Security and Message Security sections Updated error message under Section 3.7.8
03 Nov 2020	1.2	Removed Extra spaces in the Sample Json Added Common Interface Information Section Added Common Response Payload Section Updated Request and Response Payload Sections Added Table Headers to tables

Table of Contents

Version Control	2
Table of Contents	3
1. Introduction.....	4
2. Create Account, Register App and Subscribe	5
3. API Services	6
3.1 Network Security	6
3.2 Message Security	7
3.3 General Usage.....	8
3.4 Sandbox Usage	10
3.5 Production Usage	10
3.6 Common Interface Information.....	11
3.7 Common Response Payload	12
4. Submission of Employment Income Records (CorpPass) API Service	13
4.1 API Endpoint.....	13
4.2 JSON Request Payload	13
4.3 JSON Response Payload	18
4.3.1 Success Response	20
4.3.2 Error Response: Content Validation	23
4.3.3 Error Response: Header or Trailer	25
4.3.4 Error Response: Exceed Max Records.....	26
4.3.5 Error Response: Server Error	27
4.3.6 Error Response: Request Denied.....	28
4.3.7 Error Response: Request Denied.....	29
4.3.8 Error Response: Request Denied.....	30
4.3.9 Response: Not Found.....	31
5. Status Codes for API Response.....	32
6. Sample Code (C#)	33

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 RESTful 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, 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 as a guide for developers to consume IRAS API services.

2. Create Account, Register App and Subscribe

Developers are required to test APIs in the Sandbox environment first, before using them in Production. Create a developer account at <https://apisandbox.iras.gov.sg/> to subscribe to IRAS APIs for Sandbox Testing, and a developer account at <https://apiservices.iras.gov.sg/> to subscribe to IRAS APIs for Production.

Note: In order for IRAS to identify your API subscriptions, please follow the steps below to create an account:

Step 1: Create Account

- Create an account at the IRAS API Marketplace with your personal particulars. A computer-generated email will be sent to you for account activation.
- After account activation and login, click on your Username at the top-right corner of the page > My Organization > Edit organization.

Note: Organization name should reflect your actual company / organisation name (followed by the ROC or UEN where applicable), e.g. ABC Pte Ltd_201712345X.

* *Take note of the underscore after the company / organisation name.*

Step 2: Register App

- Click on Apps on the Menu Bar > Register New Application
- Ensure that the App Title is recognisable by software users, as they will be asked to give consent for the App to call this API.

Step 3: Subscribe

- Click on API Products > Select the Submission of Employment Income Records (CorpPass) API Product > Click Subscribe
- Select your registered App > Click Subscribe

Refer to <https://go.gov.sg/iras-apiuserguide> for more details on the steps above.

3. API Services

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

Name of API Service	Description	Endpoint
Submission of Employment Income Records (CorpPass)	This CorpPass authenticated API allows payroll software developers to incorporate within their payroll software the feature for employers to validate and submit employment income information (IR8A, IR8S, Appendix 8A and Appendix 8B) to IRAS directly.	<p>For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/EmplIncomeRecords/Submit</p> <p>For Actual Submission (Production): https://apiservices.iras.gov.sg/iras/prod/EmplIncomeRecords/Submit</p> <p>This service is available only when the AIS submission portal is open. AIS Employers can refer to IRAS' website for the submission period: https://www.iras.gov.sg/irashome/Businesses/Employers/Auto-Inclusion-Scheme--AIS--for-Employment-Income/#title6</p>

To consume this API service, you must be subscribed to CorpPass Authentication API. Refer to IRAS CorpPass Authentication API Services Interface Specification for more information.

3.1 Network Security

The API endpoint has to be triggered from a Server-to-Server connection. Transport Layer Security (TLS) will be used to secure end-to-end communications between the server invoking the API and IRAS API Gateway.

Your server must be able to:

1. Support the following protocols: HTTP/2, TLS 1.2/1.3, and
2. Trigger HTTP/GET and HTTP/POST requests.

3.2 Message Security

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.
access_token	CorpPass Access Token. Can be blank if validateOnly parameter is set to 'true'. Refer to 4.3 . Refer to IRAS CorpPass Authentication API Services Interface Specification on obtaining the access token. The required access scope for this API is <i>EmplIncomeSub</i> .
Content-Type	application/json

Sample HTTP Header in a HTTP Post Request

```
POST /iras/prod/EmpIncomeRecords/Submit HTTP/1.1
Host: https://apiservices.iras.gov.sg
X-IBM-Client-Id: 40e7be2f-0b4f-4985-bcc9-cdfd38c5b5c8
X-IBM-Client-Secret: vD0kR8iT3kR1hB8dPlqS3hC4lJ1aAlfV4pQ0uW0hI3uI5bW4rU
access_token: eyAidHlwIjogIkpXVCIsICJhbGciOiAiU1MyNTYiIH0.....
Content-Type: application/json
Accept: application/json
```

3.3 General Usage

Each submission allows up to 800 records in total (regardless of form type) and cannot be more than 8 MB in total data size.

Examples of Acceptable Submissions (total size < 8 MB)

- 800 IR8A records
- 500 IR8A records + 100 IR8S records + 50 Appendix 8A records + 50 Appendix 8B records
- 100 IR8A records + 30 IR8S records + 10 Appendix 8A records

Examples of Rejected Submissions

- 700 IR8A records and total size > 8 MB
- 801 IR8A records
- 201 IR8A records + 200 IR8S records + 200 Appendix 8A records + 200 Appendix 8B records

For submissions of more than 800 records or 8 MB in total data size, your application should 'chunk' it into 2 submissions.

Example of IR8A submission for 1,000 employees

- 1st submission – IR8A for 800 employees (total size < 8 MB)
- 2nd submission – IR8A for 200 employees (total size < 8 MB)

In the event that a 'chunked' submission fails, error details will be returned in the response message, and the failed submission is not accepted by IRAS.

Example:

- 1st submission – IR8A for 800 employees (Passed)
- 2nd submission – IR8A for 200 employees (Failed)

Application user will have to resubmit the 2nd submission data chunk.

Submission Guidelines:

1. If an employee has excess CPF contribution, Benefits-in-Kind and/or Stock Options Gains, submit all the relevant form types (IR8A / IR8S / Appendix 8A / Appendix 8B) of this employee within the same request.
2. Ensure the amounts reflected in the IR8A and the supporting forms (IR8S / Appendix 8A / Appendix 8B) tally.
3. Do **NOT** submit duplicate data.
4. Original data (i.e. Batch Indicator = O) must be submitted **before** submitting Amendment data.

5. When submitting Amendment data, only prepare the affected employees' records and provide the difference in amounts. Leave other numeric fields not affected by the error blank.

If the amendment for the supporting forms (IR8S/ Appendix 8A/ Appendix 8B) affects the amounts submitted for Form IR8A, an IR8A amendment data has to be submitted within the same request as well.

3.4 Sandbox Usage

The “Submission of Employment Income Records (CorpPass) (Sandbox)” plan in the sandbox environment is designed to mimic the actual Production environment so that developers can test the API integration before submitting actual data to the Production environment. IRAS’ approval is not required for subscription to this plan.

To perform the sandbox testing, developers will require the testing entity assigned by IRAS, their client ID and client secret, and the API sandbox URL (refer to section 3). Developers can refer to IRAS’ website for details on the testing process:

<https://www.iras.gov.sg/irashome/Businesses/Employers/Auto-Inclusion-Scheme--AIS-/Vendors-Supporting-AIS-Employers--Submission/#title4>.

Developers who have passed the sandbox testing can proceed to subscribe to the “Submission of Employment Income Records (CorpPass)” plan in the API Portal for Production.

Note: You must be subscribed to CorpPass Authentication (Sandbox) API before you can commence sandbox testing.

3.5 Production Usage

The “Submission of Employment Income Records (CorpPass)” plan in Production requires approval by IRAS, and it is subjected to successful completion of sandbox testing and subscription to “CorpPass Authentication” API. The subscription approval turnaround time is 10 days.

After approval is granted by IRAS, client application can POST a JSON request object to the production URL (refer to section 3).

Use of IRAS API(s) are governed by the [Additional API Terms of Use](#) found at the API Marketplace.

The use of IRAS API(s) in the Production environment is only intended for the purpose as indicated in the API service description. Any other forms of use, **including all forms of tests**, are strictly prohibited. In the event such activities are found to have occurred in the Production environment, you will be deemed to have breached the API Terms of Use, which may result in the suspension or termination of your account.

3.6 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 Type and Size	Description	Example
String(12)	A string containing maximum 12 characters.	"180084010K"
Number(4)	A numeric value containing maximum 4 whole numbers.	1990

3.7 Common Response Payload

All response payloads share the following common fields:

Parameter Name	Data Type	Description
data	Object	The data property will be populated differently based on the API that is being invoked.
returnCode	Integer	<p>10 : Success - The request was successfully processed.</p> <p>20 : Warning - The request was successfully processed. However, there are non-fatal issues. Please refer to the “info” object for diagnostic information.</p> <p>30 : Failure – The request was not processed. Refer to “info” object for error information.</p>
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	<p>Integer code signifying the type of error or warning.</p> <p>850300 : Request object is null – The incoming JSON request is null.</p> <p>850301 : Arguments error – There is an error with one of the arguments provided.</p> <p>850302 : Generic error – There is an exception within the service.</p> <p>850303 : Service is inactive.</p> <p>850304 : Service is not authorized for usage based on the provided credentials.</p> <p>850305 : Invalid test user – The input fields provided are not valid for sandbox testing.</p> <p>850806 : Invalid document reference number – The provided document reference number is not found in our system.</p> <p>850807 : DRN does not match contracting party's ID – The provided document reference number does not match the contracting party's identity number.</p>
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.

© Inland Revenue Authority of Singapore

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.

4. Submission of Employment Income Records (CorpPass) API Service

4.1 API Endpoint

Method	Endpoint
POST	/EmplIncomeRecords/Submit

4.2 JSON Request Payload

Client invoking this service will be expected to submit the following JSON request object with the following fields in the HTTP request. Note that the field names are case sensitive.

The JSON request need to be modified with escape strings and serialised in order to be acceptable by API web service. Refer to Section 5 for sample code.

Parameter Name	Description
validateOnly	<p>Boolean Indicator to indicate whether to validate JSON message only.</p> <p>If validateOnly = true, the API will perform validation of the employment income information without submission.</p> <p>If validateOnly = false, the API will perform validation of the employment income information and submission to IRAS. The access_token must be valid.</p> <p>For optimal user experience, it is strongly encouraged that the data to be submitted are validated successfully prior to obtaining CorpPass access token, as the token is only valid for a limited period of time. This can be done by submitting with validateOnly parameter set to true.</p>
bypass	<p>Boolean Indicator to indicate whether to bypass warning message thrown and proceed with submission.</p> <p>If bypass = true, hit warning and proceed with submission.</p> <p>If bypass = false, respond as error with no submission.</p>
ir8aInput	<p>String containing the content of the IR8A This will be the same as the one that is generated for IRAS' Validation and Submission Application.^#</p>
ir8sInput	<p>String containing the content of the IR8S This will be the same as the one that is generated for IRAS' Validation and Submission Application.^ #</p>
a8aInput	<p>String containing the content of the Appendix 8A</p>

	This will be the same as the one that is generated for IRAS' Validation and Submission Application.^ #
a8bInput	String containing the content of the Appendix 8B This will be the same as the one that is generated for IRAS' Validation and Submission Application.^ #
inputType	String to indicate if the content is a XML or TEXT format. Expected value includes "XML" and "TEXT".

^ Refer to the current file format published at <https://www.iras.gov.sg/IRASHome/Businesses/Employers/Auto-Inclusion-Scheme--AIS-Technical-File-Format/Specifications/>

At least one of the form inputs must be filled. All inputs must be of the same input type (TEXT / XML).

JSON request object schema

```
{
  "properties": {
    "validateOnly": {
      "type": "boolean",
      "description": "Validate Only Indicator"
    },
    "bypass": {
      "type": "boolean",
      "description": "Bypass Indicator"
    },
    "ir8aInput": {
      "type": "string",
      "description": "ir8a XML or TEXT string"
    },
    "ir8sInput": {
      "type": "string",
      "description": "ir8s XML or TEXT string"
    },
    "a8aInput": {
      "type": "string",
      "description": "a8a XML or TEXT string"
    },
    "a8bInput": {
      "type": "string",
      "description": "a8b XML or TEXT string"
    },
    "inputType": {
      "type": "string",
      "description": "XML or TEXT indicator"
    }
  }
}
```

Sample JSON request object

```

{
  "inputType": "xml",
  "a8bInput": "",
  "a8aInput": "",
  "ir8sInput": "",
  "ir8aInput": "<?xml version='1.0' encoding='utf-8' standalone='no'?><IR8A
xmlns='http://www.iras.gov.sg/IR8ADef'><IR8AHeader><ESubmissionSDSC
xmlns='http://tempuri.org/ESubmissionSDSC.xsd'><FileHeaderST><RecordType>0</Recor
dType><Source>6</Source><BasisYear>2019</BasisYear><PaymentType>08</PaymentTyp
e><OrganizationID>U</OrganizationID><OrganizationIDNo>T16ZZ0100B</OrganizationIDNo
><AuthorisedPersonName>SAMPLE
1</AuthorisedPersonName><AuthorisedPersonDesignation/><EmployerName>ABC PTE
LTD</EmployerName><Telephone>61234567</Telephone><AuthorisedPersonEmail>employ
ers@email.com</AuthorisedPersonEmail><BatchIndicator>O</BatchIndicator><BatchDate>2
0190701</BatchDate><DivisionOrBranchName/></FileHeaderST></ESubmissionSDSC></IR
8AHeader><Details><IR8ARecord><ESubmissionSDSC
xmlns='http://tempuri.org/ESubmissionSDSC.xsd'><IR8AST><RecordType
xmlns='http://www.iras.gov.sg/IR8A'>1</RecordType><IDType
xmlns='http://www.iras.gov.sg/IR8A'>1</IDType><IDNo
xmlns='http://www.iras.gov.sg/IR8A'>T8000001J</IDNo><NameLine1
xmlns='http://www.iras.gov.sg/IR8A'>SAMPLE</NameLine1><NameLine2
xmlns='http://www.iras.gov.sg/IR8A'></AddressType
xmlns='http://www.iras.gov.sg/IR8A'>L</AddressType><BlockNo
xmlns='http://www.iras.gov.sg/IR8A'>4</BlockNo><StName
xmlns='http://www.iras.gov.sg/IR8A'>Redhill</StName><LevelNo
xmlns='http://www.iras.gov.sg/IR8A'>7</LevelNo><UnitNo
xmlns='http://www.iras.gov.sg/IR8A'>3</UnitNo><PostalCode
xmlns='http://www.iras.gov.sg/IR8A'>150004</PostalCode><AddressLine1
xmlns='http://www.iras.gov.sg/IR8A'></AddressLine2
xmlns='http://www.iras.gov.sg/IR8A'></AddressLine3
xmlns='http://www.iras.gov.sg/IR8A'><TX_UF_POSTAL_CODE
xmlns='http://www.iras.gov.sg/IR8A'><CountryCode
xmlns='http://www.iras.gov.sg/IR8A'><Nationality
xmlns='http://www.iras.gov.sg/IR8A'>301</Nationality><Sex
xmlns='http://www.iras.gov.sg/IR8A'>M</Sex><DateOfBirth
xmlns='http://www.iras.gov.sg/IR8A'>19501018</DateOfBirth><Amount
xmlns='http://www.iras.gov.sg/IR8A'>150000</Amount><PaymentPeriodFromDate
xmlns='http://www.iras.gov.sg/IR8A'>20190101</PaymentPeriodFromDate><PaymentPerio
dToDate xmlns='http://www.iras.gov.sg/IR8A'>20191231</PaymentPeriodToDate><MBF
xmlns='http://www.iras.gov.sg/IR8A'>10</MBF><Donation
xmlns='http://www.iras.gov.sg/IR8A'>10</Donation><CPF
xmlns='http://www.iras.gov.sg/IR8A'>5000</CPF><Insurance
xmlns='http://www.iras.gov.sg/IR8A'>3000</Insurance><Salary
xmlns='http://www.iras.gov.sg/IR8A'>50000</Salary><Bonus
xmlns='http://www.iras.gov.sg/IR8A'>10000</Bonus><DirectorsFees
xmlns='http://www.iras.gov.sg/IR8A'>20000</DirectorsFees><Others
xmlns='http://www.iras.gov.sg/IR8A'>70000</Others><ShareOptionGainsS101g
xmlns='http://www.iras.gov.sg/IR8A'>2000</ShareOptionGainsS101g><ExemptIncome
xmlns='http://www.iras.gov.sg/IR8A'>3000</ExemptIncome><IncomeForTaxBorneByEmploy
er

```

© Inland Revenue Authority of Singapore

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.

```

xmlns="http://www.iras.gov.sg/IR8A">1000</IncomeForTaxBorneByEmployer><IncomeForTaxBorneByEmployee
xmlns="http://www.iras.gov.sg/IR8A">0</IncomeForTaxBorneByEmployee><BenefitsInKind
xmlns="http://www.iras.gov.sg/IR8A">Y</BenefitsInKind><S45Applicable
xmlns="http://www.iras.gov.sg/IR8A">Y</S45Applicable><IncomeTaxBorneByEmployer
xmlns="http://www.iras.gov.sg/IR8A">P</IncomeTaxBorneByEmployer><GratuityNoticePymExGratiaPaid
xmlns="http://www.iras.gov.sg/IR8A">Y</GratuityNoticePymExGratiaPaid><CompensationRetrenchmentBenefitsPaid
xmlns="http://www.iras.gov.sg/IR8A">Y</CompensationRetrenchmentBenefitsPaid><ApprovalObtainedFromIRAS
xmlns="http://www.iras.gov.sg/IR8A">Y</ApprovalObtainedFromIRAS><ApprovalDate
xmlns="http://www.iras.gov.sg/IR8A">20190324</ApprovalDate><CessationProvisions
xmlns="http://www.iras.gov.sg/IR8A">Y</CessationProvisions><IR8SApplicable
xmlns="http://www.iras.gov.sg/IR8A">Y</IR8SApplicable><ExemptOrRemissionIncomeIndicator
xmlns="http://www.iras.gov.sg/IR8A">1</ExemptOrRemissionIncomeIndicator><CompensationAndGratuity
xmlns="http://www.iras.gov.sg/IR8A"/><GrossCommissionAmount
xmlns="http://www.iras.gov.sg/IR8A">5000</GrossCommissionAmount><GrossCommissionPeriodFrom
xmlns="http://www.iras.gov.sg/IR8A">20190101</GrossCommissionPeriodFrom><GrossCommissionPeriodTo
xmlns="http://www.iras.gov.sg/IR8A">20190131</GrossCommissionPeriodTo><GrossCommissionIndicator
xmlns="http://www.iras.gov.sg/IR8A">M</GrossCommissionIndicator><Pension
xmlns="http://www.iras.gov.sg/IR8A">1000</Pension><TransportAllowance
xmlns="http://www.iras.gov.sg/IR8A">2000</TransportAllowance><EntertainmentAllowance
xmlns="http://www.iras.gov.sg/IR8A">3000</EntertainmentAllowance><OtherAllowance
xmlns="http://www.iras.gov.sg/IR8A">4000</OtherAllowance><GratuityNoticePymExGratia
xmlns="http://www.iras.gov.sg/IR8A">5000</GratuityNoticePymExGratia><RetrenchmentBenefits
xmlns="http://www.iras.gov.sg/IR8A">6000</RetrenchmentBenefits><RetrenchmentBenefitsUpto311292
xmlns="http://www.iras.gov.sg/IR8A">7000</RetrenchmentBenefitsUpto311292><RetrenchmentBenefitsFrom1993
xmlns="http://www.iras.gov.sg/IR8A">8000</RetrenchmentBenefitsFrom1993><EmployerContributionToPensionOrPFOOutsideSg
xmlns="http://www.iras.gov.sg/IR8A">9000</EmployerContributionToPensionOrPFOOutsideSg><ExcessEmployerContributionToCPF
xmlns="http://www.iras.gov.sg/IR8A">10000</ExcessEmployerContributionToCPF><ShareOptionGainsS101b
xmlns="http://www.iras.gov.sg/IR8A">11000</ShareOptionGainsS101b><BenefitsInKindValue
xmlns="http://www.iras.gov.sg/IR8A">12000</BenefitsInKindValue><EmployeesVoluntaryContributionToCPF
xmlns="http://www.iras.gov.sg/IR8A">13000</EmployeesVoluntaryContributionToCPF><Designation
xmlns="http://www.iras.gov.sg/IR8A">CLERK</Designation><CommencementDate
xmlns="http://www.iras.gov.sg/IR8A">19680101</CommencementDate><CessationDate
xmlns="http://www.iras.gov.sg/IR8A">20190801</CessationDate><BonusDeclarationDate
xmlns="http://www.iras.gov.sg/IR8A">20190712</BonusDeclarationDate><DirectorsFeesApprovalDate

```

© Inland Revenue Authority of Singapore

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.


```

xmlns="http://www.iras.gov.sg/IR8A">20190101</DirectorsFeesApprovalDate><RetirementBenefitsFundName
xmlns="http://www.iras.gov.sg/IR8A">SRS</RetirementBenefitsFundName><DesignatedPensionOrProvidentFundName
xmlns="http://www.iras.gov.sg/IR8A">CPF</DesignatedPensionOrProvidentFundName><BankName xmlns="http://www.iras.gov.sg/IR8A">3</BankName><PayrollDate
xmlns="http://www.iras.gov.sg/IR8A">20191212</PayrollDate><Filler
xmlns="http://www.iras.gov.sg/IR8A">ABC PAYROLL VENDOR PTE LTD
</Filler><GratuityOrCompensationDetailedInfo
xmlns="http://www.iras.gov.sg/IR8A"/><ShareOptionGainsDetailedInfo
xmlns="http://www.iras.gov.sg/IR8A"/><Remarks
xmlns="http://www.iras.gov.sg/IR8A"/></IR8AST></ESubmissionSDSC></IR8ARecord></Details><IR8ATrailer><ESubmissionSDSC
xmlns="http://tempuri.org/ESubmissionSDSC.xsd"><IR8ATrailerST><RecordType>2</RecordType><NoOfRecords>1</NoOfRecords><TotalPayment>150000</TotalPayment><TotalSalary>50000</TotalSalary><TotalBonus>10000</TotalBonus><TotalDirectorsFees>20000</TotalDirectorsFees><TotalOthers>70000</TotalOthers><TotalExemptIncome>3000</TotalExemptIncome><TotalIncomeForTaxBorneByEmployer>1000</TotalIncomeForTaxBorneByEmployer><TotalIncomeForTaxBorneByEmployee>0</TotalIncomeForTaxBorneByEmployee><TotalDonation>10</TotalDonation><TotalCPF>5000</TotalCPF><TotalInsurance>3000</TotalInsurance><TotalMBF>10</TotalMBF><Filler/></IR8ATrailerST></ESubmissionSDSC></IR8ATrailer>
</IR8A>",
  "bypass": true,
  "validateOnly": false
}

```

Sample JSON request object

```

{
  "inputType": "text",
  "a8bInput": "",
  "a8aInput": "",
  "ir8sInput": "",
  "ir8aInput": "06201908UT16ZZ0100B SAMPLE 1 ABC PTE
LTD 61234567 employers@email.com
O20190701 IR8A
\r\n11T8000001J SAMPLE L4 Redhill
7 3 150004
301M19501018150000 201901012019123110 10 5000 3000 50000 10000 20000
70000 2000 3000 1000 0 YYPYYY20190624YY1 500000
2019010120190131M100000 200000 300000 400000 500000 600000 700000
800000 900000 1000000 1100000 1200000 13000 CLERK
19680101201908012019071220190101SRS CPF
320191231
\r\n21 150000 50000 10000 20000 70000 3000 1000 0 10
5000 3000 10
\r\n",
  "bypass": true,
  "validateOnly": false
}

```

© Inland Revenue Authority of Singapore
 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.

4.3 JSON Response Payload

For API services that process or persist data, a JSON response object will be returned to the client after service is invoked in the HTTP response. Clients invoking these API services will be required to consume this JSON response object to retrieve the status and output. The schema of the response object varies across services and will be covered in detail in each specific service in the later sections.

Depending on the request to this service, the following different responses can be expected from the service based on the input provided to it from the request.

Parameter Name	Description
statusCode	A string representing the status of the service output returned to the client. The default value is 200 (OK). For a listing of valid status codes, see Status Codes section below.
msgError	A string containing general error message.
ir8a	An array containing the <u>output, warning, error</u> related to the ir8a form submitted.
ir8s	An array containing the <u>output, warning, error</u> related to the ir8s form submitted.
a8a	An array containing the <u>output, warning, error</u> related to the a8a form submitted.
a8b	An array containing the <u>output, warning, error</u> related to the a8b form submitted.

Detailed description of the output, warning and error are listed the table below.

<p>output</p>	<p>An array containing pipe delimited string/s containing the Submission Reference Number and the processed information of the submission, as shown in sample below. Client is expected to store the Submission Reference Number for future reference.</p> <p>The output is delimited in the following order:</p> <ul style="list-style-type: none"> • Submission Reference Number • Organisation Name • Organisation Reference Number • Basis Year + 1 • File Type - "O" for original submission, "A" for amendment • Date and time of submission in ddMMyyHHmm format • User ID • Number of Records
<p>error</p>	<p>An array list of error messages. Each array item will contain a JSON error object with the following fields:</p> <ul style="list-style-type: none"> • RecordType – a string containing the type of record that caused the error, record type will be specific to each service. • RecordField – a string containing the field that caused the error. • RecordIdentifier – a string containing the ID of the record or row that caused the error • error – a string containing the error message or number. <p>(0 = Header, 1 = Detail, 2 = Trailer)</p>
<p>warning</p>	<p>An array list of warning messages. Each array item will contain a JSON error object with the following fields:</p> <ul style="list-style-type: none"> • RecordType – a string containing the type of record that caused the error, record type will be specific to each service. • RecordField – a string containing the field that caused the error. • RecordIdentifier – a string containing the ID of the record or row that caused the error • error – a string containing the warning message or number. <p>(0 = Header, 1 = Detail, 2 = Trailer)</p>

4.3.1 Success Response

A successful call to the service means that the AIS submission is accepted. A JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value "200", representing STATUS_OK.
msgError	Empty
ir8a	An array containing the output, warning, error related to the ir8a form submitted.
ir8s	An array containing the output, warning, error related to the ir8s form submitted.
a8a	An array containing the output, warning, error related to the a8a form submitted.
a8b	An array containing the output, warning, error related to the a8b form submitted.

Sample JSON response object

```
{
  "statusCode": "200",
  "ir8a": {
    "output": "ES19300819IR8A5378635154|ABC PTE LTD|T16ZZ0100B|2020|O|3008191208|S1111111C|1",
    "errors": [],
    "warnings": [{}]
  },
  "ir8s": {
    "errors": [],
    "warnings": []
  },
  "a8a": {
    "errors": [],
    "warnings": []
  },
  "a8b": {
    "errors": [],
    "warnings": []
  }
}
```

OR

If bypass=true, warning message will be shown and submission is allowed.

```
{
  "statusCode": "200",
  "ir8a": {
    "output": "ES19300819IR8A5378635154|ABC PTE LTD|T16ZZ0100B|2020|O|3008191208|S1111111C|1",
    "errors": [],
    "warnings": [
      {

```

© Inland Revenue Authority of Singapore
 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.

```

        "recordType": "1",
        "recordField": "Detail",
        "recordIdentifier": "T16ZZ0100B",
        "error": "To help ensure the data entry is accurate and correct, the system has
highlighted the following for your information: \r\n-IR8A file contains 1 record(s) with
employees' income tax to be paid by the employer. \r\nYou may ignore this warning if you
confirm the records are in order."
    }
]
},
"ir8s": {
    "errors": [],
    "warnings": []
},
"a8a": {
    "errors": [],
    "warnings": []
},
"a8b": {
    "errors": [],
    "warnings": []
}
}
}

```

For successful production submission, an email will also be triggered to both the company contact representative as well as the individual that submit the salary file through payroll software.

The format of the email is shown below:

Dear Sir / Madam

Organisation Name: ABC PTE LTD

Organisation Tax Ref No.: 200312345A

The following file(s) has been successfully submitted to IRAS and will be processed within 5 working days. We will contact you if there is any issue with the submitted file(s).

Submissions for Employees' Employment Income earned in the Year 2018 (YA2019) Submitted by SXXXX111K				
Form Type	Date and Time of Submission	e-Submission Reference Number	File Type	Number of Records Submitted
Form IR8A	-	-	-	-
Appendix 8A	01/02/2019 10:40 AM	ES18010219IR8A1234567890	Original	1
Appendix 8B	-	-	-	-
Form IR8S	-	-	-	-

To help ensure data accuracy and completeness, we would like to bring your attention to the following point(s) on your submission:

1) There are 1 record(s) with no Form IR8A in this submission. Form IR8A is required for all employees.
You may ignore this warning if you intend to submit or have submitted the IR8A in a separate submission.

This is a system-generated email. If you did not perform this submission, please clarify with your payroll software vendor and notify IRAS immediately at ais@iras.gov.sg.

Thank you for supporting the Auto-Inclusion Scheme for Employment Income.

Yours faithfully

The Auto-Inclusion Team

Individual Income Tax Division - Employee Branch

Inland Revenue Authority of Singapore

ais@iras.gov.sg (email) | 1800 356 8015 (tel) | www.iras.gov.sg | www.facebook.com/irassg | www.twitter.com/iras_sg

A Partner of the Community in Nation-Building and Inclusive Growth

Confidentiality: Privileged and/or confidential information may be contained in this email or its attachments. If this email has been sent to you by mistake, please notify the sender and delete it immediately. The retention or communication of this email's contents or its attachments to an unauthorised person is prohibited and may be an offence under the Official Secrets Act (Cap 213).

4.3.2 Error Response: Content Validation

When the records in the content has failed validations, the call to the service is unsuccessful. A JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value "400", representing STATUS_BAD_REQUEST.
msgError	String which may contain a simplified error message or an error number.
ir8a	An array containing the <u>output, warning, error</u> related to the ir8a form submitted.
ir8s	An array containing the <u>output, warning, error</u> related to the ir8s form submitted.
a8a	An array containing the <u>output, warning, error</u> related to the a8a form submitted.
a8b	An array containing the <u>output, warning, error</u> related to the a8b form submitted.

Sample JSON response object

```
{
  "statusCode": "400",
  "msgError": "",
  "ir8a": {
    "output": "",
    "warnings": [],
    "errors": [
      {
        "recordType": "2",
        "recordField": "BasisYear",
        "recordIdentifier": "200312345A",
        "error": "Invalid"
      }
    ]
  },
  "ir8s": {
    "output": "",
    "warnings": [],
    "errors": []
  },
  "a8a": {
    "output": "",
    "warnings": [],
    "errors": []
  },
  "a8b": {
    "output": "",
    "warnings": [],
    "errors": []
  }
}
```

© Inland Revenue Authority of Singapore
 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.

If bypass=false, warning message will be shown and submission will not be allowed.

```

{
  "statusCode": "400",
  "msgError": "",
  "ir8a": {
    "output": "",
    "warnings": [
      {
        "recordType": 1,
        "recordField": "IDNo",
        "recordIdentifier": "S1234567A",
        "error": "ir8s is required"
      }
    ],
    "errors": [ ]
  },
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}

```


4.3.3 Error Response: Header or Trailer

When the Header or Trailer in the content has failed validations, the call to the service is unsuccessful. A JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value "400", representing STATUS_BAD_REQUEST.
msgError	String which may contain a simplified error message or an error number.
ir8a	An array containing the <u>output, warning, error</u> related to the ir8a form submitted.
ir8s	An array containing the <u>output, warning, error</u> related to the ir8s form submitted.
a8a	An array containing the <u>output, warning, error</u> related to the a8a form submitted.
a8b	An array containing the <u>output, warning, error</u> related to the a8b form submitted.

Sample JSON response object

```
{
  "statusCode": "400",
  "msgError": "",
  "ir8a": {
    "output": "",
    "warnings": [{}],
    "errors": [{
      "recordType": "2",
      "recordField": "BasisYear",
      "recordIdentifier": "200312345A",
      "error": "Invalid"
    }
  ]
},
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}
```

4.3.4 Error Response: Exceed Max Records

When the number of records submitted is more than 800, the call to the service is unsuccessful. A JSON response object that contains the following fields will be returned in the HTTP response. There will be only one error record in an **error** array, as shown in sample below.

Parameter Name	Description
statusCode	A string containing the value "413", representing STATUS_REQUEST_TOO_LARGE.
msgError	String which may contain a simplified error message or an error number.
ir8a	An array containing the <u>output, warning, error</u> related to the ir8a form submitted.
ir8s	An array containing the <u>output, warning, error</u> related to the ir8s form submitted.
a8a	An array containing the <u>output, warning, error</u> related to the a8a form submitted.
a8b	An array containing the <u>output, warning, error</u> related to the a8b form submitted.

Sample JSON response object

```
{
  "statusCode": "413",
  "msgError": "NoOfRecords Exceed 800",
  "ir8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}
```

4.3.5 Error Response: Server Error

When an exception occurs during the processing of the request, a JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value "500", representing STATUS_SERVER_ERROR.
msgError	String which may contain a simplified error message or an error number.

Sample JSON response object

```
{
  "statusCode": "500",
  "msgError": "Internal Server Error Occurred. Please try again later.",
  "ir8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}
```

4.3.6 Error Response: Request Denied

When the *access_token* given is not valid, a JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value "401", representing STATUS_DENIED.
msgError	String which may contain a simplified error message or an error number.

Sample JSON response object

```
{
  "statusCode": "401",
  "msgError": "The access_token given is invalid, please re-login via CorpPass.",
  "ir8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}
```

4.3.7 Error Response: Request Denied

When the *access_token* given has expired, a JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value “401”, representing STATUS_DENIED.
msgError	String which may contain a simplified error message or an error number.

Sample JSON response object

```
{
  "statusCode": "401",
  "msgError": "The access_token given has expired, please re-login via CorpPass.",
  "ir8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}
```

4.3.8 Error Response: Request Denied

When the *access_token* given lacks the required authorisation, a JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description
statusCode	A string containing the value “401”, representing STATUS_DENIED.
msgError	String which may contain a simplified error message or an error number.

Sample JSON response object

```
{
  "statusCode": "401",
  "msgError": "Please check if you have been authorised with an Approver role to make the submission.",
  "ir8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "ir8s": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8a": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  },
  "a8b": {
    "output": "",
    "warnings": [ ],
    "errors": [ ]
  }
}
```

4.3.9 Response: Not Found

When the API has been unpublished or turned off, a JSON response object that contains the following fields will be returned in the HTTP response.

Parameter Name	Description+
httpCode	A string containing the value "404", representing STATUS_SERVER_ERROR.
httpMessage	String which may contain a simplified error message or an error number.
moreInformation	String which may contain more details about the error

Sample JSON response object

```
{
  "httpCode": "404",
  "httpMessage": "Not Found",
  "moreInformation": "The requested URL was not found on this server"
}
```

5. Status Codes for API Response

The statusCode field will always contain an integer representing the processed state of the request. The list of possible status codes and what they represent are listed below.

Status	Status Code	Remarks
STATUS_OK	200	The request completed successfully.
STATUS_NO_CONTENT	204	The server has fulfilled the request, but there is no new information to send back.
STATUS_MOVED	301	The requested service has been assigned to a new permanent Uniform Resource Identifier (URI), and any future references to this service should be done using one of the returned URIs.
STATUS_REDIRECT	302	The requested service resides temporarily under a different URI.
STATUS_NOT_MODIFIED	304	The requested did not make any modification.
STATUS_BAD_REQUEST	400	The request could not be processed by the server due to invalid inputs.
STATUS_DENIED	401	The requested service requires user authentication/authorisation.
STATUS_GONE	410	The requested service is no longer available at the server, and no forwarding address is known.
STATUS_REQUEST_TOO_LARGE	413	The server cannot process the request because the submitted entity is larger than the server is able to process.
STATUS_SERVER_ERROR	500	The server encountered an unexpected condition that prevented it from fulfilling the request.

6. Sample Code (C#)

Escape strings and serialize the data before calling API

```

// Step 1 Read IR8A file

string input = File.ReadAllText(file);

// Step 2 Escape string

StringBuilder escapeString = new StringBuilder(input);

escapeString.Replace("\"", "\\\"");

escapeString.Replace("\r", "\\r");

escapeString.Replace("\n", "\\n");

escapeString.Replace("\t", "\\t");

input = escapeString.ToString();

// Step 3 Create submission object

ApiSubmission inputObj = new ApiSubmission()

{

    InputType = "XML",

    Bypass = true,

    ValidateOnly = true,

    IR8AInput = input

};

// Step 4 Serialize the object using JavaScriptSerializer

var serializer = new JavaScriptSerializer();

StringBuilder serializedData = new StringBuilder();

serializer.Serialize(inputObj, serializedData);

//Step 5 Send serialized data - function from specs

callWebAPI(serializedData, accessToken);

```

Calling the Web API

```

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

public static void callWebAPI(string jsonData, string accessToken)
{
    //Step 0 : Call ServerCertificateValidationCallback
    ServicePointManager.ServerCertificateValidationCallback += (sender, cert, charin,
sslIPolicyErrors) => true;

    // Step 1: Construct URL
    String url = "https://apisandbox.iras.gov.sg/iras/sb/EmplIncomeRecords/Submit";

    try
    {
        var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
        httpWebRequest.ContentType = "application/json";
        httpWebRequest.Method = "POST";

        //Step 2: Enter the Client-Id given by IRAS
        httpWebRequest.Headers["X-IBM-Client-Id"] = "<Your Client ID>";
        //Step 3: Enter the Client-Secret given by IRAS
        httpWebRequest.Headers["X-IBM-Client-Secret"] = "<Your Client Secret>";

        //Step 4: Enter the Client-Secret given by IRAS
        httpWebRequest.Headers["access_token"] = accessToken;

        // Step 5: Call API using POST
        using (var streamWriter = new StreamWriter(httpWebRequest.GetRequestStream()))
        {
            streamWriter.Write(jsonData);
            streamWriter.Flush();
            streamWriter.Close();
        }

        // Step 5a: 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 5b: Print general errors
    Console.WriteLine("Exception - ");
    Console.WriteLine(e.Message);
}

if (e.Response != null)
{
    // Step 5c: 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());
}
}
```