IRAS API SERVICES INTERFACE SPECIFICATIONS

SingPass Authentication

Last updated on: 28 May 2018

Version No.: 1.1

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 and 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; and
- 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	SingPassAuth	This API can be used to retrieve the SingPass's Uniform Resource Locator (URL) in order to obtain the authorisation code from SingPass.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/Authentication/SingPassAuth
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/Authentication/SingPassAuth
2	SingPassToken	This API can be used to retrieve the authorisation token.	For Sandbox Testing: https://apisandbox.iras.gov.sg/iras/sb/Authentication/SingPassToken
			For Production Usage: https://apiservices.iras.gov.sg/iras/prod/Authentication/SingPassToken

3.1 General Information

The SingPassAuth and SingPassToken endpoints have to be triggered from a Server-to-Server connection. Approval is required to use these services.

3.1.1 Sandbox Usage

These services in the sandbox environment are designed to mimic the production environment so that developers can test the API integration before submitting actual data to the production environment.

Refer to Appendix 4.3 for the list of available test entities for SingPass Login.

3.1.2 Production Usage

Approval is required to use these services. After approval is granted by IRAS, client application can POST a JSON request object to the production URL.

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.

3.1.3 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

3.1.4 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. **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 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 an error.
Info.fieldInfoList.message	String	Diagnostic message provided to aid consumer’s developers.

3.2 SingPassAuth API Service

The SingPassAuth endpoint has to be triggered from a Server-to-Server connection using HTTP GET request. The response will be an URL which should be returned to the client for redirection to SingPass Login Page. The intended result after SingPass Login is to obtain the authorisation code to invoke the SingPassToken API service to retrieve the authorisation token. Approval is required to use this service.

The followings are the 5 steps involved to obtain the authorisation code from the Consent Platform:

1. Invocation of the SingPassAuth API service from IRAS (using HTTP GET request)
2. IRAS returns the SingPass Login Page's URL as part of the response message
3. Software redirects software user to the SingPass Login Page
4. Software user enters his/her SingPass credential
5. Software is redirected to SingPass Consent Page for software user to accept the agreement

3.2.1 Request Payload

Parameter Name	Data Type	Description
scope	String	The list of functions that the external application is requesting for. *Refer to Appendix 4.2 for the list of Scopes.*
callback_url	String	Callback url for Consent Platform to return with the authorisation code
state	String	Identifier to reconcile request and response. This will be sent back via the callback url

Sample HTTP GET Request

```
GET
/iras/prod/Authentication/SingPassAuth?scope=GSTReturnsSub+GSTTransLis
tSub&callback_url=http://www.iras.gov.sg/callback&state=390b25fa-4427-
4b10-9ae2-34d6e0cd91a1 HTTP/1.1
Host: https://apiservices.iras.gov.sg
X-IBM-Client-Id: 40e7be2f-0b4f-4985-bcc9-cdfd38c5b5c8
X-IBM-Client-Secret:
vD0kR8iT3kR1hB8dP1qS3hC4lJ1aA1fV4pQ0uW0hI3uI5bW4rU
Content-Type: application/json
Accept: application/json
```

3.2.2 Response Payload

data	Object	The object payload containing the information after successful invocation
data.url	String	SingPass Login Page's url
data.state	String	Value of state that was provided in the request
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":{
    "url":"https://stg-
saml.singpass.gov.sg/FIM/sps/SingpassIDPFed/saml20/logininitial?client
_id=a1234b5c-1234-abcd-efgh-
a1234b5cdef&scope=GSTReturnsSub+GSTTransListSub&redirect_uri=http://ww
w.iras.gov.sg/callback&state=390b25fa-4427-4b10-9ae2-34d6e0cd91a1",
    "state": "390b25fa-4427-4b10-9ae2-34d6e0cd91a1",
  },
  "info":{
    "fieldInfoList":[]
  }
}
```

Sample error JSON response payload

```
{
  "returnCode":"30",
  "data":{
    "state":"390b25fa-4427-4b10-9ae2-34d6e0cd91a1",
  },
  "info":{
    "messageCode":"850301",
    "message":"Arguments Error",
    "fieldInfoList":[
      {
        "field":"callback_url",
        "message":"The callback_url specified is not registered"
      }
    ]
  }
}
```

3.2.3 Response (SingPass Login Page)

The response given by SingPass will be a HTTP temporary redirect (302) response with the SingPass Consent Page URL and the following parameters in the redirection url.

Parameter Name	Data Type	Description
client_id	String	Unique identifier for external application
scope	String	The list of functions that the external application is requesting for. *Refer to Appendix 4.2 for the list of Scopes.*
callback_url	String	Callback url for Consent Platform to return with the authorisation code

state	String	Identifier to reconcile request and response. This will be sent back via the callback url

3.2.4 Response Payload (SingPass Consent Page)

The response given by SingPass will be a HTTP temporary redirect (302) response with the given callback URL and the following parameters. The authorisation code has to be received by a server for the subsequent token API service invocation.

Parameter Name	Data Type	Description
code	String	Authorisation Code
client_id	String	Unique identifier for external application
scope	String	The list of functions that the external application is requesting for. *Refer to Appendix 4.2 for the list of Scopes.*
state	String	Identifier to reconcile request and response. This will be sent back via the callback url

3.3 SingPassToken API Service

The SingPassToken endpoint has to be triggered from a Server-to-Server connection using HTTP GET request. The response will be the authorisation token that is necessary for authorised functional API service invocation. Approval is required to use this service.

3.3.1 Request Payload

Parameter Name	Data Type	Description
scope	String	The list of functions that the external application is requesting for. *Refer to Appendix 4.2 for the list of Scopes.*
callback_url	String	Callback url for validation purpose
code	String	Authorisation Code

Sample HTTP GET Request

```
GET
/iras/prod/Authentication/SingPassToken?scope=GSTReturnsSub+GSTTransLi
stSub&callback_url=http://www.iras.gov.sg/callback&code=322c89af-3921-
9f8e-1ab3-87f8de0bc8ce HTTP/1.1
Host: https://apiservices.iras.gov.sg
X-IBM-Client-Id: 40e7be2f-0b4f-4985-bcc9-cdfd38c5b5c8
X-IBM-Client-Secret:
vD0kR8iT3kR1hB8dP1qS3hC4lJ1aA1fV4pQ0uW0hI3uI5bW4rU
Content-Type: application/json
Accept: application/json
```

3.3.2 Response Payload

The response will be returned with the authorisation token. This token has to be passed as additional parameters for functional API service invocation that requires this authorisation token.

data	Object	The object payload containing the information after successful invocation
data.token	String	Authorisation Token
data.code	String	Value of state that was provided in the request
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": {
    "token" :
"eyAidHlwIjogIkpXVCIsICJhbGciOiAiUlMyNTYiIH0.eyAic3ViIjogIlM5ODEyMzgxR
CIsICJhdWRpdFRyYWNraW5nSWQiOiAiZjU0Njc3MGQtOTI4Ni00ODNlLTk0YzktZTllNDZ
lMTMwMzFiIiwgImlzcyI6ICJodHRwOi8vc3RnLWF1dGguYXBwcy5lLm5lY3Rhci5nb3Yuc
2c6ODAvY29uc2VudC9vYXV0aDIiLCAidG9rZW5OYW1lIjogImFjY2Vzc190b2tlbiIsICJ
0b2tlbl90eXBlIjogIkJlYXJlciIsICJhdXRoR3JhbnRJZCI6ICI4ZDRkNTEwNC0yMmMxL
TRkZDYtYTA5Mi0yYmU3YzZlNzkzNDMiLCAiYXVkIjogIm15aW5mbyIsICJuYmYiOiAxNTA
1ODkwNzUxLCAic2NvcGUiOiBbICJlZHVsZXZlbCIsICJob3VzaW5ndHlwZSIsICJyYWNlI
iwgInJlZ2FkZCIsICJzZXgiLCAiaGRidHlwZSIsICJtb2JpbGVubyIsICJjcGZiYWxhbmN
lcyIsICJoYW55dXBpbnlpbm5hbWUiLCAibWFycmllZG5hbWUiLCAiaGFueXVwaW55aW5hb
Glhc25hbWUiLCAiYXNzZXNzYWJsZWluY29tZSIsICJtYXJpdGFsIiwgImFsaWFzbmFtZSI
sICJuYXRpb25hbGl0eSIsICJkb2IiLCAibmFtZSIsICJjcGZjb250cmlidXRpb25zIiwgI
mVtYWlsIiBdLCAicmVhbG0iOiAiL215aW5mbyIsICJleHAiOiAxNTA1ODk0MzUxLCAiaWF
0IjogMTUwNTg5MDc1MSwgImV4cGlyZXNfaW4iOiAzNjAwMDAwLCAianRpIjogIjI2MmM3O
Dk5LWY4YmQtNDVlOC05Mzc1LWRmYTY5YWUwNTA3YSIgfQ.maD5lnrRA84hm2Lz5gdUlM6Z
VSsYxYlWodtMTesleDXJrD3hDeUVbmHGumeH4XogtmYacCWKh3geY5iItK4WLr_ft5oFzv
XilWrud7zahFOR5QuOxhALzl8d1erDN4jCQfbT6xq98B7nRCG9JAB1qvEvnSGQLGSWxH3j
4IxE5Jp5iP-3AzshXub-
8JukAqnWZLiQsoJCUu8GBg0RfG85wRKwImPmnPyCoAwqTv2MlqhZQr5oXkkeRrqITkZQME
3sFiqZmnExjHcml8Y__5h3FNvJSFdnnZSDW0471wBpYM9-jK7WdBR9rKS68sxdNa85y-
TQ7YXcb5AHYPro9GGY6w",
    "code":"322c89af-3921-9f8e-1ab3-87f8de0bc8ce"
  },
  "info": {
    "fieldInfoList": []
  }
}
```

Sample error JSON response payload


```
{
  "returnCode": "30",
  "data": {
    "code": "322c89af-3921-9f8e-1ab3-87f8de0bc8ce"
  },
  "info": {
    "messageCode": "850301",
    "message": "Arguments Error",
    "fieldInfoList": [
      {
        "field": "code",
        "message": "Authentication code verification failed"
      }
    ]
  }
}
```


3.3.2.1 JWT (JSON Web Token)

JSON Web Tokens consist of three parts separated by dots (.), they are:

1) Header
2) Payload
3) Signature

Therefore, a JWT typically looks like this: xxxxx.yyyyy.zzzzz

The header and the payload are Base64 encoded. The signature is created using these two, a secret and the hashing algorithm (as specified in the header: HMAC, SHA256 or RSA).

Refer to the example of the JSON Web Token below with the dots highlighted in red.


```
eyAidHlwIjogIkpXVCIsICJhbGciOiAiUlMyNTYiIH0.eyAic3ViIjogIlM5ODEyMzgxRC
IsICJhdWRpdFRyYWNraW5nSWQiOiAiZjU0Njc3MGQtOTI4Ni00ODNlLTk0YzktZTllNDZl
MTMwMzFiIiwgImlzcyI6ICJodHRwOi8vc3RnLWF1dGguYXBwcy5lLm5lY3Rhci5nb3Yuc2
c6ODAvY29uc2VudC9vYXV0aDIiLCAidG9rZW5OYW1lIjogImFjY2Vzc190b2tlbiIsICJ0
b2tlbl90eXBlIjogIkJlYXJlciIsICJhdXRoR3JhbnRJZCI6ICI4ZDRkNTEwNC0yMmMxLT
RkZDYtYTA5Mi0yYmU3YzZlNzkzNDMiLCAiYXVkIjogIm15aW5mbyIsICJuYmYiOiAxNTA1
ODkwNzUxLCAic2NvcGUiOiBbICJlZHVsZXZlbCIsICJob3VzaW5ndHlwZSIsICJyYWNlIi
wgInJlZ2FkZCIsICJzZXgiLCAiaGRidHlwZSIsICJtb2JpbGVubyIsICJjcGZiYWxhbmNl
cyIsICJoYW55dXBpbnlpbm5hbWUiLCAibWFycmllZG5hbWUiLCAiaGFueXVwaW55aW5hbG
lhc25hbWUiLCAiYXNzZXNzYWJsZWluY29tZSIsICJtYXJpdGFsIiwgImFsaWFzbmFtZSIs
ICJuYXRpb25hbGl0eSIsICJkb2IiLCAibmFtZSIsICJjcGZjb250cmlidXRpb25zIiwgIm
VtYWlsIiBdLCAicmVhbG0iOiAiL215aW5mbyIsICJleHAiOiAxNTA1ODk0MzUxLCAiaWF0
IjogMTUwNTg5MDc1MSwgImV4cGlyZXNfaW4iOiAzNjAwMDAwLCAianRpIjogIjI2MmM3OD
k5LWY4YmQtNDVlOC05Mzc1LWRmYTY5YWUwNTA3YSIgfQ.maD5lnrRA84hm2Lz5gdUlM6ZV
SsYxYlWodtMTesleDXJrD3hDeUVbmHGumeH4XogtmYacCWKh3geY5iItK4WLr_ft5oFzvX
ilWrud7zahFOR5QuOxhALzl8d1erDN4jCQfbT6xq98B7nRCG9JAB1qvEvnSGQLGSWxH3j4
IxE5Jp5iP-3AzshXub-
8JukAqnWZLiQsoJCUu8GBg0RfG85wRKwImPmnPyCoAwqTv2MlqhZQr5oXkkeRrqITkZQME
3sFiqZmnExjHcml8Y__5h3FNvJSFdnnZSDW0471wBpYM9-jK7WdBR9rKS68sxdNa85y-
TQ7YXcb5AHYPro9GGY6w
```

3.3.3 List of Diagnostic Messages

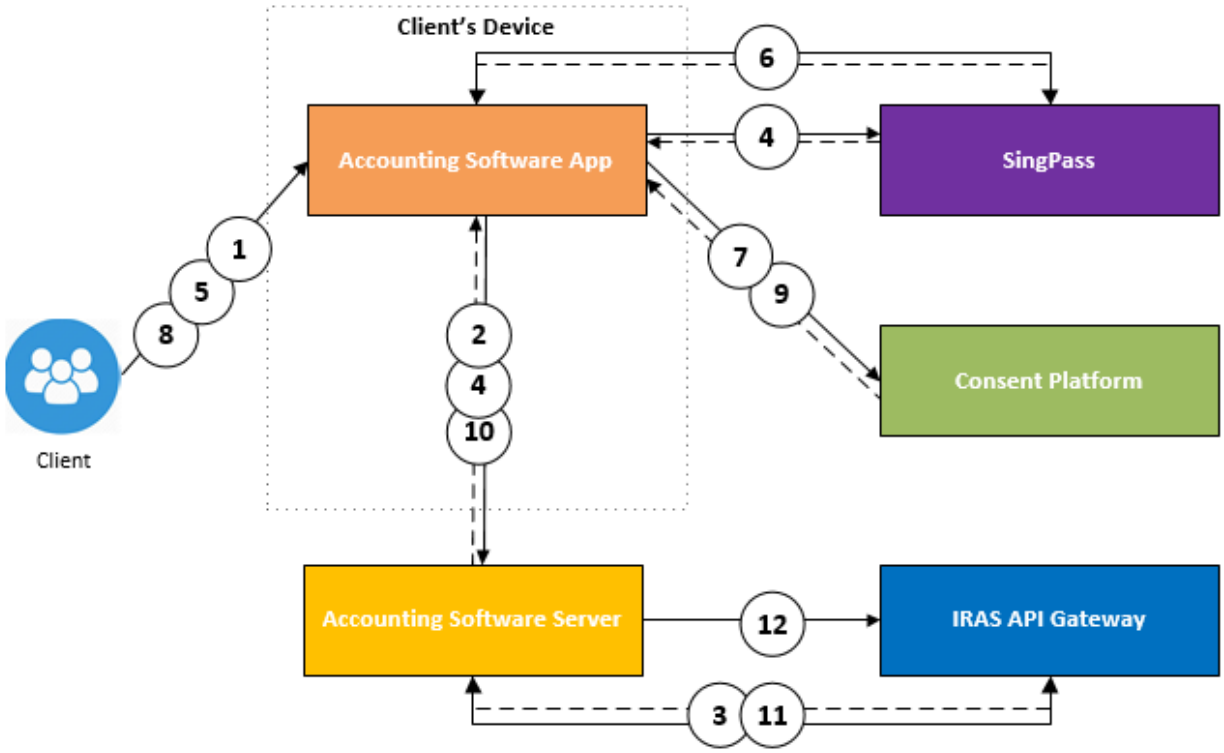
The table below contains the list of diagnostic messages that would be returned as a response in case of failure during the processing of the SingPass Authentication Request. As per section 3.1.4, these diagnostic messages will be part of the information within the info.fieldInfoList array.

field	message
callback_url	• API: Value cannot be null, empty, or white space • API: Callback_url mismatch with client registered callback url
scope	• API: Value cannot be null, empty, or white space • API: Scope mismatch with client registered scope
code	• API: Value cannot be null, empty, or white space

4 Appendix

4.1 Interaction Flow

The following interaction flow depicts the flow of events on the usage of the APIs.



1. **Client** clicks "Submit to IRAS" from **Accounting Software App**.
2. **Accounting Software App** to trigger **Accounting Software Server** to perform IRAS API SingPassAuth call.
3. The **IRAS API Gateway** will return a URL string of the SingPass Login Page with the callback URL and other parameters.
4. **Accounting Software Server** to send the SingPass URL string to **Accounting Software App** to redirect to **SingPass** Login Page for entry of credential.
5. **Client** clicks "Submit" at the **Accounting Software App** with SingPass Login Page.
6. Upon successful login to SingPass, **SingPass** will trigger a HTTP Temporary Redirection (302) to Consent Platform.
7. **Accounting Software App** is redirected to **Consent Platform** Page for agreement of the term of use.
8. **Client** clicks "Yes" at the **Accounting Software App** with Consent Platform Page.
9. **Consent Platform** will trigger a HTTP Temporary Redirection (302) with the callback URL with the Authorisation Code with other parameters.
10. The callback URL should be the **Accounting Software Server**'s domain so that the Authorisation Code and other parameters are sent to the server for further processing.

11. Using the Authorisation Code**, Accounting Software Server** will perform IRAS API SingPassToken call to **IRAS API Gateway** to request for an Authorisation Token.
12. **Accounting Software Server** can proceed to perform functional IRAS API calls to **IRAS API Gateway** with the Authorisation Token.

4.2 List of Scopes

The use of Scope allows proper permissioning of the functional API via the Authorisation Token. For request containing multiple Scopes, concatenate them with '+' char.

S/N	API	Scope
1	GST F5 Return Submission	GSTReturnsSub
2	GST Transaction Listing Submission	GSTTransListSub

4.3 Test Entities for SingPass

The table below shows the available test entities for SingPass.

S/N	SingPass ID	Password
1	S9812381D	MyInfo2o15
2	S6005054F	MyInfo2o15

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 28 May 2018 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.