



## 1.0 - accessRelyingPartyService() : (Customer ==> Relying Party)

### Description:

Customer Requesting to Access Relying Service.

## 1.1 - www\_authorise() : (Relying Party ==> ISAM WebSeal)

### Description: OIDC

Relying party issues an authorize request with the appropriate scope and claims

### Request:

https://<<idhub>>/authorize?response\_type=code&client\_id=rp&scope=openid profile email phone  
tdif business authorisations&redirect\_uri=<<hostname>>/rp/exchange&nonce=<<nonce>>&state=<<state>>&acr\_values=urn:id.gov.au:tdif:acr:ip2:cl2&claims={"id\_token":{"mygov\_linked":{"essential":true}},"userinfo":{"mygov\_linked":{"essential":true}}}

### Response:

HTTP/1.1 200 OK

## 1.2 - loadExchangeSPA() : (ISAM WebSeal ==> WebServer)

### Description:

Load Exchange SPA page

## 1.3 - displayIDPSelectionPage() : (Exchange SPA ==> Exchange SPA)

### Description: UI action

Display the IDP Selection page to the user

## 1.4 - customerSelectsIDP() : (Customer ==> Exchange SPA)

### Description: User action

Customer selects the preferred IDP from the list

## 1.5 - checkmyGovFlowRequired() : (Exchange SPA ==> Exchange SPA)

check if {"mygov\_linked":{"essential":true}} is present in the Relying Party Authorisation request

## 1.6 - kickOffLinkableIDPAuthorizeFlow() : (Exchange SPA ==> ISAM AAC)

### Description: API

Kickoff the IDP authorize flow

### Security:

Bearer-Token: ISAM\_LEVEL1\_JWT

### Request:

POST: /sso/sps/oidc/rp/IDENTITYHUB/kickoff/MYGOVIDP?acr\_values=<<acr\_values>>

### Response:

HTTP/1.1 200 OK

### BODY:

```
{"location":"/mga/sps/oauth/oauth20/authorize?scope=openid+email+link
&state=<<state>>
&client_id=rp_exchange_client
&nonce=<<nonce>>
&redirect_uri=/sso/sps/oidc/rp/IDENTITYHUB/redirect/MYGOVIDP
&acr_values=<<acr_values>>
&response_type=code"}
```

## 1.7 - tdifIDPAuthReq(tdif\_request) : (ISAM AAC ==> Identity Provider)

### Description: API

Send TDIF OIDC Authorize request to Selected IDP as a front channel redirect.

### Security:

ISAM\_LEVEL1\_JWT

### Request:

#### GET:

/proto-idp/oauth/authorize?claims=<<claims>>&state=<<state>>&client\_id=PROTOIDP&nonce=<<nonce>>&redirect\_uri=https://<<idhub>>/sso/sps/oidc/rp/IDENTITYHUB/redirect/PROTOIDP&acr\_values=<<acr\_values>>&response\_type=code

### Response:

HTTP/1.1 302 Found

## 1.8 - authenticateUser() : (Identity Provider ==> Identity Provider)

### Description:

Authenticate the IDP User.

## 1.9 - tdifIDPAuthResponse() : (Identity Provider ==> ISAM AAC)

### Description:

TDIF IDP Authorisation response

### Security:

ISAM\_LEVEL1\_JWT

### Request:

GET: /sso/sps/oidc/rp/IDENTITYHUB/redirect/PROTOIDP?code=<<code>>&state=<<state>>

**Response:**  
HTTP/1.1 302  
Redirect from IDP

### **1.10 - tdifIDPTokenRequest(client\_assertion\_type, code, grant\_type, redirect\_uri, state, client\_assertion, client\_id) : (ISAM AAC ==> Identity Provider)**

**Description:**  
Modify this to receive the new Claim EDI.

**Request:**  
REST API Request URL : <<idhub>>/idp/oauth/token  
Request Method : POST  
Request params:  
{client\_assertion\_type=<<client-assertion-type>>&code=APb2br&grant\_type=authorization\_code&redirect\_uri=<<redirect\_uri>>&state=2ilqGhrrGH&client\_assertion=<<client\_assertion>>&client\_id=IDP}

**Response:**  
{ "access\_token": "<<access\_token>>", "token\_type": "bearer", "expires\_in": 43199, "scope": "email openid phone tdif\_core", "id\_token": "<<id\_token>>" }

### **1.11 - tdifIDPUserInfoRequest() : (ISAM AAC ==> Identity Provider)**

**Description: API**  
Retrieve IDP User information

**Request:**

**Security:**  
**Header:** Bearer Token

**Response:**  
HTTP/1.1 200 OK

### **1.12 - addIDPClaims() : (ISAM AAC ==> EXT-Issuer API)**

**Description: API**  
Store the new IDP (EDI) claims in the database

**Security:**  
**Header:** ex-ext-iss-api-key: <<ISAM\_EXCHANGE\_JWT>>

**Request:**  
**PUT:** <<exchange>>/api-ext-issuer/v1/identityproviders/<<idp\_client\_id>>/authclaims

**BODY:**  
{ "claims": { "sub": "<<sub>>",  
"aud": "PROTOIDP",  
"acr": "urn:id.gov.au:tdif:acr:ip2:cl2",  
"auth\_time": 1564557530,  
"kid": "1",  
"iss": "https://<<idhub>>/proto-idp",  
"exp": "<<exp>>",  
"iat": "<<iat>>",  
"nonce": "<<nonce>>",  
"jti": "<<jti>>",  
"phone\_number": "000",  
"family\_name": "Mayweather",  
"email\_verified": true,  
"phone\_number\_verified": true,  
"updateAt": 1564557531302,  
"email": "test@email.com",  
"given\_name": "Mike",  
"birthdate": "<<birthdate>>" } }

**Response:**  
HTTP/1.1 200 OK

### **1.13 - searchCustomer : (EXT-Issuer API ==> EXT-Issuer API)**

### **1.14 - calculateRPKey : (EXT-Issuer API ==> EXT-Issuer API)**

### **1.15 - searchCustomer : (EXT-Issuer API ==> EXT-Issuer API)**

### **1.16 - calculateRPKey : (EXT-Issuer API ==> EXT-Issuer API)**

### **1.17 - searchCustomer : (EXT-Issuer API ==> EXT-Issuer API)**

### **1.18 - promptForConsent : (Exchange ==> Exchange SPA)** Redirect to SPA to display the consent page to share details with myGov

### **1.19 - promptForConsent : (Exchange SPA ==> Customer )** Redirect to SPA to display the consent page to share details with myGov

### 1.20 - customerConsents() : (Customer ==> Exchange SPA)

**Description:** User Action

Customer provides consent to share details with myGov

### 1.21 - storeRPCConsent() : (Exchange SPA ==> EXT-Customer API)

**Description:** API

Store Relying Party Consent

**Security:**

Bearer-Token: ISAM\_LEVEL3\_JWT

**Request:**

**PUT:** <<idhub>>/api-ext-customer-ui/v1/rpauthentications/<<rp\_auth\_id>>/consent

**BODY:**

```
{ "id": "<<rp_auth_id>>",
  "claimConsents": [],
  "abn": "<<abn>>",
  "triggerScope": "<<triggerScope>>" }
```

**Response:**

HTTP/1.1 201 Created

### 1.22 - confirmmyGovConnection : (Exchange SPA ==> Exchange)

**Description:** Internal API

Verify and confirm if a link exists between myGov and Exchange for the given user

### 1.23 - createUpdateLink(rpLinkID) : (Exchange ==> myGov Integration)

**Description:** API

Create a new myGov Relying Party Account Link / MBUN in the myGov system for a given relying party. The user account will be specified by the use of the "sub", "squ" and "sty" claims of the mgv-jwt JWT header

**Security:**

Bearer-Token: EXCHANGE\_MYGOV\_JWT

**Request:**

**POST (Create):** /accounts/links/

**BODY:** RelyingPartyLink object

```
{
  "relyingPartyId": "ATO|EXCHGE|...",
  "relyingPartyName": "Australian Taxation Office",
  "relyingPartyLinkDetails": {
    "id": "<<id>>",
    "status": "permanent|transient",
    "created": "<<created>>",
    "lastModified": "<<lastModified>>"
  }
}
```

**Response:**

HTTP/1.1 201 Relying party link successfully created

### 1.24 - createUpdateProfile() : (Exchange ==> myGov Integration)

**Description:** API

Create or Update User's myGov profile in myGov

**Security:**

Bearer-Token: EXCHANGE\_MYGOV\_JWT

**Request:**

**POST (Create):** /accounts/profile

**PUT (Update):** /accounts/profile

**BODY:** myGov Profile

```
{ "name": {
  "firstName": "string",
  "middleName": "string",
  "lastName": "string"
},
  "dateOfBirth": "2019-11-05"
}
```

**Response:**

HTTP/1.1 204 Successfully Created or Updated

### 1.25 - promptForConsent() : (Exchange ==> Exchange SPA)

Redirect to SPA to display the consent page to share details with the relying party

### 1.26 - promptForConsent : (Exchange SPA ==> Customer )

Redirect to SPA to display the consent page to share details with the relying party

### 1.27 - customerConsents() : (Customer ==> Exchange SPA)

**Description:** User Action

Customer provides consent to share details with the relying party

### 1.28 - storeRPCConsent() : (Exchange SPA ==> EXT-Customer API)

**Description:** API

Store Relying Party Consent

**Security:**

Bearer-Token: ISAM\_LEVEL3\_JWT

**Request:**

**PUT:** <<idhub>>/api-ext-customer-ui/v1/rpauthentications/<<rp\_auth\_id>>/consent

**BODY:**

```
{ "id": "<<rp_auth_id>>",  
  "claimConsents": [],  
  "abn": "<<abn>>",  
  "triggerScope": "<<triggerScope>>" }
```

**Response:**

HTTP/1.1 201 Created

### 1.29 - confirmRPConnection : (Exchange SPA ==> Exchange)

**Description:** Internal API

Verify and confirm if a link exists between the Relying Party and Exchange for the given user

### 1.30 - generateRPLink : (Exchange ==> Exchange)

Generate new RP Pairwise Identifier using the relevant sector

### 1.31 - saveRPLinkAndKey : (Exchange ==> Exchange)

Save the generated RP pairwise identifier

### 1.32 - createUpdateLink(rpLinkID) : (Exchange ==> myGov Integration)

**Description:** API

Create a new myGov Relying Party Account Link / MBUN in the myGov system for a given relying party. The user account will be specified by the use of the "sub", "squ" and "sty" claims of the mgv-jwt JWT header

**Security:**

Bearer-Token: EXCHANGE\_MYGOV\_JWT

**Request:**

**POST (Create):** /accounts/links/

**BODY:** RelyingPartyLink object

```
{  
  "relyingPartyId": "ATO|EXCHGE|...",  
  "relyingPartyName": "Australian Taxation Office,  
  "relyingPartyLinkDetails": {  
    "id": "<<id>>",  
    "status": "permanent|transient",  
    "created": "<<created>>",  
    "lastModified": "<<lastModified>>"  
  }  
}
```

**Response:**

HTTP/1.1 201 Relying party link successfully created

### 1.33 - tdifRPAuthoriseResponse() : (ISAM AAC ==> Relying Party)

Send an authorized response back to the Relying Party along with the RP claims