



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() : (Exchange SPA ==> Exchange SPA)

Description: User action

Customer selects the preferred IDP from the list

1.5 - 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"}
```

1.6 - 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

1.7 - authenticateUser() : (Identity Provider ==> Identity Provider)

Description:

Authenticate the IDP User.

1.8 - 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

1.9 - tdifIDPUserInfoRequest() : (ISAM AAC ==> Identity Provider)

1.10 - addIDPClaims() : (ISAM AAC ==> EXT-Issuer API)

Description: API

Store IDP 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

1.11 - generateLinkableIDPAuthId() : (EXT-Customer API ==> Exchange SPA)

Generate IDP_AUTH_ID for the linkable flow

1.12 - getIDPAuthUserInfo() : (Exchange SPA ==> EXT-Customer API)

Description: API

Retrieve IDP Auth User Information

Security:

Bearer-Token: ISAM_LEVEL3_JWT

Request:

GET: <<idhub>>/api-ext-customer-ui/v1/idpauthentications/<<idp_auth_id>>/<<rp_auth_id>>/userinfo

Response:

HTTP/1.1 200 OK

BODY:

```
{ "idpAuthId": "<<idp_auth_id>>", "sessionId": "<<session_id>>", "claims": [ { "claims": {} } ] }
```

1.13 - getRPDetails() : (Exchange SPA ==> EXT-Customer API)

Description: API

Retrieve details for a given relying party

Security:

Bearer-Token: ISAM_LEVEL3_JWT

Request:

GET: <<idhub>>/api-ext-customer-ui/v1/relyingparty/7

Response:

HTTP/1.1 200 OK

BODY:

```
{ "id": "rp biz client",
  "name": "DSS GRS",
  "description": "Dept Social Services Grants Registration System",
  "properties": { "displayName": "Grants Registration Portal",
    "serviceName": "Grants Registration Portal" }
```

1.14 - checkmyGovFlowRequired() : (Exchange SPA ==> Exchange SPA)

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

1.15 - confirmmyGovConnection() : (Exchange SPA ==> Exchange Connector)

1.16 - getExchangeLink(reylingParty) : (Exchange Connector ==> Exchange Application Server)

Description: API

Verify if myGov is linked to Exchange

1.17 - verifyAccount() : (Exchange Connector ==> myGov Integration)

Description: API

Verify if the linked account for the given MBUN is active and return the matching email address.

Security:

Bearer-Token: **EXCHANGE_MYGOV_JWT**

Request:

GET: /authenticator/verify

Response:

HTTP/1.1 200 OK

BODY:

{"email":"<<email>>"}

1.18 - clearExistingmyGovExchangeLink() : (Exchange Connector ==> Exchange Connector)

1.19 - verifyAccount() : (Exchange Connector ==> myGov Integration)

Description: API

Verify if account exists in myGov for the given DiD Email address

Security:

Bearer-Token: **EXCHANGE_MYGOV_JWT**

Request:

GET: /authenticator/verify

Response:

HTTP/1.1 200 OK

1.20 - confirmmyGovAccountCreation() : (Exchange Connector ==> Exchange Connector)

1.21 - displaymyGovLoginPage() : (Exchange Connector ==> Exchange SPA)

Display myGov login page for the User to sign-in with their myGov credentials

1.22 - loginUsingmyGovCredentials(myGov_Creds) : (Exchange SPA ==> myGov Login)

Log into myGov using myGov credentials

1.23 - myGovOIDCAuthReq() : (ISAM AAC ==> myGov Login)

Description: OIDC

Create OIDC Auth request for myGov Linkable flow

/sso/sps/oidc/rp/IDENTITYHUB/redirect/MYGOVIDP?state=<<state>>&code=<<code>>

1.24 - myGovOIDCAuthResponse() : (myGov Login ==> ISAM AAC)

Description: OIDC response

Send the Auth response to ISAM AAC for myGov Linkable flow

Request:

/sso/sps/oidc/rp/IDENTITYHUB/redirect/MYGOVIDP?state=<<state>>&code=<<code>>

Response:

HTTP/1.1 302

Redirect from IDP

1.25 - getIDToken() : (ISAM AAC ==> myGov Login)

Description: API

Obtain the Identity Token from myGov

Security:

Bearer-Token: **EXCHANGE_MYGOV_JWT**

Request:

POST: <<mygov>>/core/connect/token

BODY:

{"code":"<<mygov_auth_code>>"}

Response:

200

BODY:

```
{ "mbun": "<<mbun>>",
  "acr": "2",
  "amr": [ "PASSWORD", "SECRET QUESTION"],
  "gsk": "bV_xyaFisYBJBkZKMGay7Ot",
  "lt": "permanent"
}
```

1.26 - getUserInfo() : (ISAM AAC ==> myGov Login)

1.27 - checkDiDEmailMatchesmyGovEmail() : (Exchange Connector ==> Exchange Connector)

Description: Internal API

Verify myGov email address with Digital Identity email address

1.28 - promptForConsent() : (Exchange Connector ==> Exchange SPA)

Redirect to SPA to display the consent page

1.29 - customerConsents() : (Customer ==> Exchange SPA)

Description: User Action

Customer provides consent to share details with myGov

1.30 - 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

1.31 - confirmmyGovConnection() : (Exchange SPA ==> Exchange Connector)

1.32 - getmyGovAccountLinks() : (Exchange Connector ==> myGov Integration)

Description: API

Verify if a link exists between myGov and the Relying Party (e.g. Exchange)

Security:

Bearer-Token: EXCHANGE_MYGOV_JWT

Request:

GET: /accounts/links

PARAM: relyingPartyId

Response:

HTTP/1.1 200 OK

BODY: RelyingPartyLink object

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

1.33 - createUpdateProfile() : (Exchange Connector ==> 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

1.34 - createUpdateLink(rpLinkID) : (Exchange Connector ==> 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

1.35 - generateLink(relyingParty) : (Exchange Connector ==> Exchange Application Server)

Generate link for myGov in Exchange

1.36 - createMyGovAccount() : (Exchange Connector ==> myGov Integration)

Description: API

Create a new myGov account for the User in the myGov system

Security:

Bearer-Token: EXCHANGE_MYGOV_JWT

Request:

POST (Create): /accounts/

BODY: myGovAccount object

```
{ "myGovAccount ": "<<myGovAccount>>" }
```

Response:

HTTP/1.1 201: Successfully created

1.37 - createUpdateProfile() : (Exchange Connector ==> 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

1.38 - createUpdateLink(rpLinkID) : (Exchange Connector ==> 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

1.39 - promptForConsent() : (Exchange Connector ==> Exchange SPA)

Redirect to SPA to display the consent page

1.40 - customerConsents() : (Customer ==> Exchange SPA)

Description: User Action

Customer provides consent to share details with the relying party

1.41 - 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

1.42 - getRPFinalisedAuthRequest() : (Exchange SPA ==> EXT-Customer API)

Description: API

Retrieve Relying Party Finalised Auth Request

Security: ISAM_LEVEL3_JWT

Request:

GET: <<idhub>>/api-ext-customer-ui/v1/rpauthentications/<<rp_auth_id>>/finalised

Response:

HTTP/1.1 200 OK

BODY:

```
{ "rpAuthId": "<<rp_auth_id>>",  
  "fullURL": "<<fullURL>>",  
  "sessionId": "<<session_id>>" }
```

1.43 - kickOffRPAuthorizeFlow() : (Exchange SPA ==> ISAM AAC)

Description: API

Invoke this API to kickoff the RP Authorise flow

Security:

Bearer-Token: ISAM_LEVEL3_JWT

Request:

POST:

<<idhub>>/sso/sps/oauth/oauth20/authorize?response_type=code&client_id=rp_biz_client&scope=<,scope>>&redirect_uri=<<redirect_uri>>&state=<,state>>&acr_values=<<acr_values>>&prompt=none

Response:

{ "location": "https://<<idhub>>/proto-rp/exchange?state=<<state>>&code=<<code>>" }

1.44 - confirmRPCConnection() : (Exchange SPA ==> Exchange Connector)

Description: Internal API

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

1.45 - getExchangeLink(reylingParty) : (Exchange Connector ==> Exchange Application Server)

Description: API

Verify if the RP link exists in Exchange

1.46 - getmyGovAccountLinks() : (Exchange Connector ==> myGov Integration)

Description: API

Verify if a link exists between myGov and the Relying Party (e.g. Exchange)

Security:

Bearer-Token: EXCHANGE_MYGOV_JWT

Request:

GET: /accounts/links

PARAM: relyingPartyId

Response:

HTTP/1.1 200 OK

BODY: RelyingPartyLink object

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

```

    "created": "<<created>>",
    "lastModified": "<<lastModified>>"
  }
}

```

1.47 - generateLink(relyingParty) : (Exchange Connector ==> Exchange Application Server)

Generate relying party pairwise identifier in Exchange

1.48 - createUpdateLink(rpLinkID) : (Exchange Connector ==> 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

1.49 - tdifRPAuthoriseResponse() : (ISAM AAC ==> Relying Party)

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