



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>>&ac
r_values=urn:id.gov.au:tdif:acr:ip2:cl2&claims={"id_token":{"mygov_linked":{"essential":true}}, "useri
nfo":{"mygov_linked":{"essential":true}}}

Response:

HTTP/1.1 200 OK

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

Description:

Load Exchange SPA page

1.3 - generateAndStoreRPAuthID() () : (Exchange SPA ==> SPA Session Storage)

Description: Function

Generate **RP_AUTH_ID** and store it in the session storage

1.4 - checkSessionStorage() : (Exchange SPA ==> SPA Session Storage)

Description: Function

Check if **ISAM_SSO_JWT** exists in the session storage.

1.5 - checkLocalStorage() : (Exchange SPA ==> SPA Local Storage)

Description: Function

Check if **ISAM_SSO_JWT** exists in the local storage.

1.6 - 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"}]}

1.7 - 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"}}
```

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

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

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

Description: Internal API

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

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

Description: API

Verify if myGov is linked to Exchange

1.11 - 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>>"}

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

1.13 - 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

BODY:

{"mbun": "<<mbun>>"} // if exists

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

Confirm new mygov account auto-creation

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

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

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

Log into myGov using myGov credentials

1.17 - 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>>

1.18 - 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

1.19 - 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_xyaFIsYBJbZKMGay7Ot",  
  "lt": "permanent"  
}
```

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

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

Connector)

Description: Internal API

Verify myGov email address with Digital Identity email address

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

Redirect to SPA to display the consent page

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

Description: User Action

Customer provides consent to share details with myGov

1.24 - 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

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

Description: Internal API

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

1.26 - 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>>"
  }
}
```

1.27 - 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

1.28 - 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

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

Generate link for myGov in Exchange

1.30 - 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

1.31 - 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

1.32 - 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

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

Redirect to SPA to display the consent page

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

Description: User Action

Customer provides consent to share details with the relying party

1.35 - 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

1.36 - getRPFfinalisedAuthRequest() : (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>>"
}

```

1.37 - 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>>"}

1.38 - 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

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

Description: API

Verify if the RP link exists in Exchange

1.40 - 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>>"
  }
}
```

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

Generate relying party pairwise identifier in Exchange

1.42 - 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

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

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