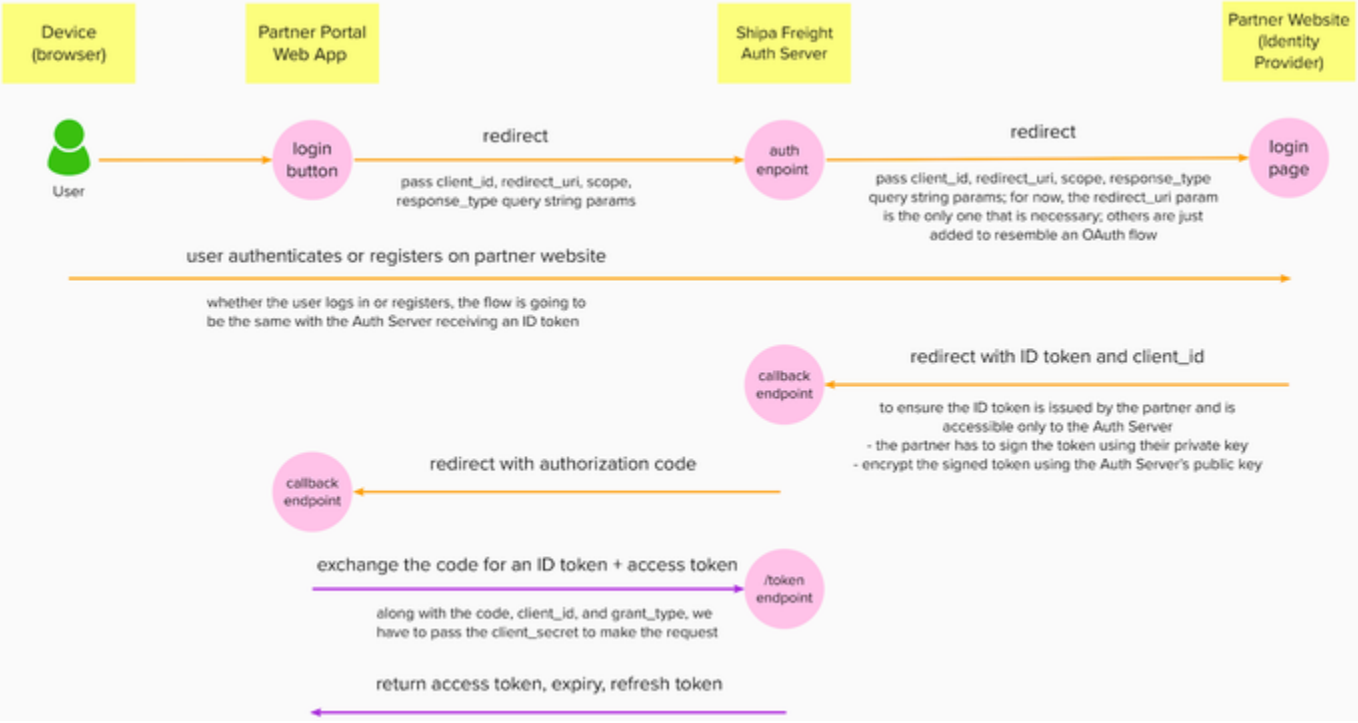
Shipa Freight Auth Server Partner Integration Spec v1.3

Introduction

In order for us to enable Single Sign-On (SSO) with partners on the partner portals, we have developed an OpenID Connect (OIDC) Authentication server (SF Auth) built on top of the OAuth 2.0 framework. Since we do not expect partners to be OAuth / OIDC compliant, we have come up with a flow to ensure that authentication happens in a seamless and secure manner. At no point in the flow will we ever have access to the users' credentials. User login and registration will continue to happen on partner websites.

When a user visits the partner portal (e.g. https://partner.shipafreight.com), they will see a login button which will redirect them to the Auth Server. The Auth Server will automatically redirect the user to a login page on the partner website, along with a few URL query parameters. This login page can be an existing login page or a new login page built for users of the partner portal. To allow users to register, it is ideal to have a register link on the login page.

Login and registration should continue to be the same for partners. However, once login / registration completes, the partner website will redirect back to the Auth Server with an ID token, which will be sufficient to authenticate the user on the partner portal. If the user is authenticating for the first time, an entry will be created in the database. If some fields that are required for quoting and booking are missing in the ID token payload, the user will be prompted with a welcome modal on their first login on the portal where they will be asked to provide additional details.



OAuth Flow

The following is a breakdown of the different steps that will be expected to be implemented by the partner website to complete the integration with the Auth Server

Step 0 - User accesses Auth Server

- To start the authentication process, users have to first access the Auth Server either through the login page on the partner portal or directly using the Direct Auth link
- The Auth Server creates a session for the user, sets cookies in the user's browser, and then redirects to a configured URL for that partner, passing a redirect URL as a query string parameter
- Once authentication is completed at the partner website, a redirect back to the redirect URL with the ID token will complete the authentication process

Step 1 - Partner website validates incoming request

- The Auth Server will redirect to a login page on the partner website with 3 URL query parameters, namely **client_id**, **response_type**, and **redirect_uri**

- It is recommended to validate the values of these 3 query parameters, particularly the **redirect_uri** parameter (*refer to the example implementation*)
- **client_id** is the ID of the Auth Server
- **response_type** tells the partner website what data to return, which will be restricted to **id_token**
- **redirect_uri** tells the partner website the URL to redirect back to once login or registration succeeds or fails; it is highly recommended to validate that it points to the Auth Server
    - **redirect_uri** will be URL-encoded

Step 2 - Partner website creates the ID token payload

- after login or registration succeeds, an ID token payload with the user's details has to be constructed (*refer to the example implementations*)
- the following are the required ID token claims as specified in the OpenID Connect Core 1.0 spec
    - **iss** - issuer of the token - ***required***
        - a case sensitive HTTPS URL of the partner e.g. https://partner.com
    - **sub** - subject of the token - ***required***
        - same as the `email` field below
    - **aud** - audience (recipient for which the token is intended) - ***required***
        - the `client_id` which is passed to the partner by Auth Server (*not the same as the* `client_id` *query parameter in the direct auth link*)
    - **iat** - issued at time (time at which the token was issued) - ***required***
        - a number representing the number of seconds from 1970–01–01T0:0:0Z as measured in UTC
    - **exp** - expiration time (time at which the token expires) - ***required***
        - a number representing the number of seconds from 1970–01–01T0:0:0Z as measured in UTC
- the following are the additional user claims that Auth Server recognizes
    - **email** - the user's email address - ***required***
    - **firstName** - the user's first name - ***optional***
    - **lastName** - the user's last name - ***optional***
    - **companyName** - the user's company name - ***optional***
    - ***taxId*** *- the user's tax ID* - ***optional***
    - **countryCode** - the user's two-digit country code - ***optional***
        - the full list can be found under Appendix: List of Supported Country Codes
    - **phoneNumber** - the user's phone number in full international format - **optional**
        - the full format includes a plus sign (+) followed by the country code, city code, and local phone number
- if any of the above optional user fields are invalid, they will be ignored and not saved in the user's record; on first log in to the partner portal, the user will be prompted to provide the missing fields

Step 3 - Partner website signs and encrypts the ID token payload

- signing will ensure that the ID token is coming from the partner and not from an attacker
    - partners will have to generate or use an existing RSA key pair, and then use the private key to sign the ID token payload
    - it is critical to secure the private key and not to expose it anywhere; failure to protect the private key could open the door to a number of attack vectors
    - since the Auth Server relies on the partner's public key to verify tokens, our **v1** implementation requires that we be notified when partner keys are changed in order to keep the integration running
- encrypting will ensure that the ID token is only visible to the Auth Server and not to an attacker
    - after the ID token has been signed, it has to be encrypted using the Auth Server's encryption public key

Step 4 - Partner website redirects back to the Auth Server with the ID token

- after encryption, the partner website has to redirect back to the **redirect_uri** query parameter, passing along the encrypted signed ID token as an **id_token** query parameter
    - for example, if the **redirect_uri** parameter is https://auth.shipafrieght.com/auth, the redirect URL would then be https://auth.shipafrieght.com/auth?id_token=ID_TOKEN_VALUE

Optional Step 4 - Partner website redirects back to the Auth Server with an error

- to handle failures gracefully, the partner website can redirect back to the Auth Server with an error code and a description using the **error** and **error_description** query parameters
- the following are the accepted values for the **error** query parameter:
    - **access_denied** - the user or the partner server denied the request
    - **server_error** - the partner website encountered an unexpected condition that prevented it from fulfilling the request

- **temporarily_unavailable** - the partner website is currently unable to handle the request due to a temporary overloading or maintenance of the server
- **user_canceled_request** - the user canceled the sign-in request
- **invalid_client** - the specified client isn't valid
- **invalid_request** - the request parameters aren't valid

Integration Modes

Direct Auth Link

- Partners can use direct auth links to authenticate users without requiring users to visit the login page on the partner portal first
  - the URL pattern is as follows
    - https://auth-staging.shipafreightservices.com/auth?scope=openid&response_type=code&client_id=CLIENT_ID&redirect_uri=REDIRECT_URI (staging)
    - https://auth.shipafreight.com/auth?scope=openid&response_type=code&client_id=CLIENT_ID&redirect_uri=REDIRECT_URI (production)
  - **CLIENT_ID** would be the client ID of the partner portal (*we will be providing this*)
  - **REDIRECT_URI** would be a callback URL on the partner portal (*we will be providing this*)
  - the direct auth link also supports passing all of the five UTM parameters, namely **utm_source**, **utm_medium**, **utm_campaign**, **utm_search** and **utm_content**

Testing

- To ensure that the correct ID token is generated, it's helpful to use https://auth-server-test-idp.herokuapp.com/auth/decode-token to verify the token
- To test end-to-end, we would need the URL of the partner login page and the partner public key to configure the Auth Server. This will redirect users to the partner login page. With the correct ID token passed when redirecting back to the Auth Server, the partner public key will be used to verify the token. If verification passes, users will be logged in to the partner portal.

Appendix: Auth Server Public Keys

- the encryption public keys are available in JWK (JSON Web Key) format at the following endpoints
  - https://auth.shipafreight.com/jwks (production)
  - https://auth-staging.shipafreightservices.com/jwks (staging)
- a key in JWK format can be converted to PEM format and vice-versa

Appendix: Generating an RSA key pair

- OpenSSL

```
# generate a private key
openssl genrsa -out private-key.pem 2048

# generate corresponding public key
openssl rsa -in private-key.pem -pubout -out public-key.pem
```

- https://mkjwk.org

Appendix: Example Partner Implementations

NodeJS: https://github.com/shipafreight/auth-server-sample-idp-nodejs

PHP: https://github.com/shipafreight/auth-server-sample-idp-php

Appendix: List of Supported Country Codes

- Afghanistan AF
- Albania AL
- Algeria DZ
- American Samoa AS
- Andorra AD
- Angola AO
- Anguilla AI
- Antarctica AQ
- Antigua and Barbuda AG
- Argentina AR
- Armenia AM
- Aruba AW
- Australia AU
- Austria AT
- Azerbaijan AZ
- Bahamas (the) BS
- Bahrain BH
- Bangladesh BD
- Barbados BB
- Belarus BY
- Belgium BE
- Belize BZ
- Benin BJ
- Bermuda BM
- Bhutan BT
- Bolivia (Plurinational State of) BO
- Bosnia and Herzegovina BA
- Botswana BW
- Bouvet Island BV
- Brazil BR
- British Indian Ocean Territory (the) IO
- Brunei Darussalam BN
- Bulgaria BG
- Burkina Faso BF
- Burundi BI
- Cabo Verde CV
- Cambodia KH
- Cameroon CM
- Canada CA
- Cayman Islands (the) KY
- Central African Republic (the) CF
- Chad TD
- Chile CL
- China CN
- Christmas Island CX
- Cocos (Keeling) Islands (the) CC
- Colombia CO
- Comoros (the) KM
- Congo (the Democratic Republic of the) CD
- Congo (the) CG
- Cook Islands (the) CK
- Costa Rica CR
- Croatia HR
- Curaçao CW
- Cyprus CY

- Czechia CZ
- Côte d'Ivoire CI
- Denmark DK
- Djibouti DJ
- Dominica DM
- Dominican Republic (the) DO
- Ecuador EC
- Egypt EG
- El Salvador SV
- Equatorial Guinea GQ
- Eritrea ER
- Estonia EE
- Eswatini SZ
- Ethiopia ET
- Falkland Islands (the) [Malvinas] FK
- Faroe Islands (the) FO
- Fiji FJ
- Finland FI
- France FR
- French Guiana GF
- French Polynesia PF
- French Southern Territories (the) TF
- Gabon GA
- Gambia (the) GM
- Georgia GE
- Germany DE
- Ghana GH
- Gibraltar GI
- Greece GR
- Greenland GL
- Grenada GD
- Guadeloupe GP
- Guam GU
- Guatemala GT
- Guernsey GG
- Guinea GN
- Guinea-Bissau GW
- Guyana GY
- Haiti HT
- Heard Island and McDonald Islands HM
- Holy See (the) VA
- Honduras HN
- Hong Kong HK
- Hungary HU
- Iceland IS
- India IN
- Indonesia ID
- Iraq IQ
- Ireland IE
- Isle of Man IM
- Israel IL
- Italy IT
- Jamaica JM
- Japan JP
- Jersey JE

- Jordan JO
- Kazakhstan KZ
- Kenya KE
- Kiribati KI
- Korea (the Republic of) KR
- Kuwait KW
- Kyrgyzstan KG
- Lao People's Democratic Republic (the) LA
- Latvia LV
- Lebanon LB
- Lesotho LS
- Liberia LR
- Liechtenstein LI
- Lithuania LT
- Luxembourg LU
- Macao MO
- Madagascar MG
- Malawi MW
- Malaysia MY
- Maldives MV
- Mali ML
- Malta MT
- Marshall Islands (the) MH
- Martinique MQ
- Mauritania MR
- Mauritius MU
- Mayotte YT
- Mexico MX
- Micronesia (Federated States of) FM
- Moldova (the Republic of) MD
- Monaco MC
- Mongolia MN
- Montenegro ME
- Montserrat MS
- Morocco MA
- Mozambique MZ
- Myanmar MM
- Namibia NA
- Nauru NR
- Nepal NP
- Netherlands (the) NL
- New Caledonia NC
- New Zealand NZ
- Nicaragua NI
- Niger (the) NE
- Nigeria NG
- Niue NU
- Norfolk Island NF
- Northern Mariana Islands (the) MP
- Norway NO
- Oman OM
- Pakistan PK
- Palau PW
- Palestine, State of PS
- Panama PA

- Papua New Guinea PG
- Paraguay PY
- Peru PE
- Philippines (the) PH
- Pitcairn PN
- Poland PL
- Portugal PT
- Puerto Rico PR
- Qatar QA
- Republic of North Macedonia MK
- Romania RO
- Russian Federation (the) RU
- Rwanda RW
- Réunion RE
- Saint Barthélemy BL
- Saint Helena, Ascension and Tristan da Cunha SH
- Saint Kitts and Nevis KN
- Saint Lucia LC
- Saint Martin (French part) MF
- Saint Pierre and Miquelon PM
- Saint Vincent and the Grenadines VC
- Samoa WS
- San Marino SM
- Sao Tome and Principe ST
- Saudi Arabia SA
- Senegal SN
- Serbia RS
- Seychelles SC
- Sierra Leone SL
- Singapore SG
- Slovakia SK
- Slovenia SI
- Solomon Islands SB
- Somalia SO
- South Africa ZA
- South Georgia and the South Sandwich Islands GS
- South Sudan SS
- Spain ES
- Sri Lanka LK
- Suriname SR
- Svalbard and Jan Mayen SJ
- Sweden SE
- Switzerland CH
- Taiwan (Province of China) TW
- Tajikistan TJ
- Tanzania, United Republic of TZ
- Thailand TH
- Timor-Leste TL
- Togo TG
- Tokelau TK
- Tonga TO
- Trinidad and Tobago TT
- Tunisia TN
- Turkey TR
- Turks and Caicos Islands (the) TC

- Tuvalu TV
- Uganda UG
- Ukraine UA
- United Arab Emirates (the) AE
- United Kingdom of Great Britain and Northern Ireland (the) GB
- United States of America (the) US
- Uruguay UY
- Uzbekistan UZ
- Vanuatu VU
- Venezuela (Bolivarian Republic of) VE
- Viet Nam VN
- Virgin Islands (British) VG
- Virgin Islands (U.S.) VI
- Wallis and Futuna WF
- Yemen YE
- Zambia ZM
- Zimbabwe ZW
- Åland Islands AX