Authentication via Client ID – Instructions for developers

Bahn-IT ServiceHub

Exported on 2024-02-23 14:42:33

Inhalt

1 What is an ID token and what is it used for? 3

2 Authentication process in IAM 4

3 ID Token Validation 5

4 Examples 6

4.1 Endpoints 6
4.2 Access Token 7
4.3 ID Token 8
4.4 Userinfo 9
4.5 Refresh Token 10
4.6 Postman 10
4.7 SoapUI / ReadyAPI 10

1 What is an ID token and what is it used for?

First of all: IAM stands for "Identity and Access Management" and focuses on the administration, authentication and authorisation of external users for INFRA BS applications. The corresponding client ID and secret client can be requested from the responsible technology manager.

An ID token is an artefact that proves that the user has been authenticated. It was introduced by OpenID Connect (OIDC), an open standard for authentication that is used by many identity providers.

An ID token is encoded as a JSON Web Token (JWT), a standard format that allows your application to easily verify the content and ensure that it comes from the expected issuer and has not been altered by someone else.

In simple terms, an example of an ID token looks like this:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi8vbXktZG9tYWluLmF1d
GgwLmNvbSIsInN1YiI6ImF1dGgwfDEyMzQ1NiIsImF1ZCI6IjEyMzRhYmNkZWYiLCJleHAiOjE
zMTEyODE5NzAsImlhdCI6MTMxMTI4MDk3MCwibmFtZSI6IkphbmUgRG9lIiwiZ2l2ZW5fbmFtZ
SI6IkphbmUiLCJmYW1pbHlfbmFtZSI6IkRvZSJ9.bql-
jxlG9B_bielkqOnjTY9Di9FillFb6IMQINXoYsw
```

Of course, this is not human readable, so you have to decode it to see what content the JWT contains. By the way, the ID token is not encrypted, but it is base 64 encoded. You can use one of the many available librariesfor decoding, or you can examine it yourself with the JWT.IO debugger.

These JSON properties are called claims and are declarations about the user and the token itself. The information about the user defines the identity of the user.

In the OpenID Connect specifications, it is not necessary for the ID token to contain information about the user. In its minimal structure, it does not contain any data about the user, only information about the authentication process.

The ID token may contain additional information about the user, such as their email address, photo, birthday, etc.

Finally the ID token is signed by the issuer with the private key. This guarantees you the origin of the token and ensures that it has not been tampered. You can verify these things with the issuer's public key.

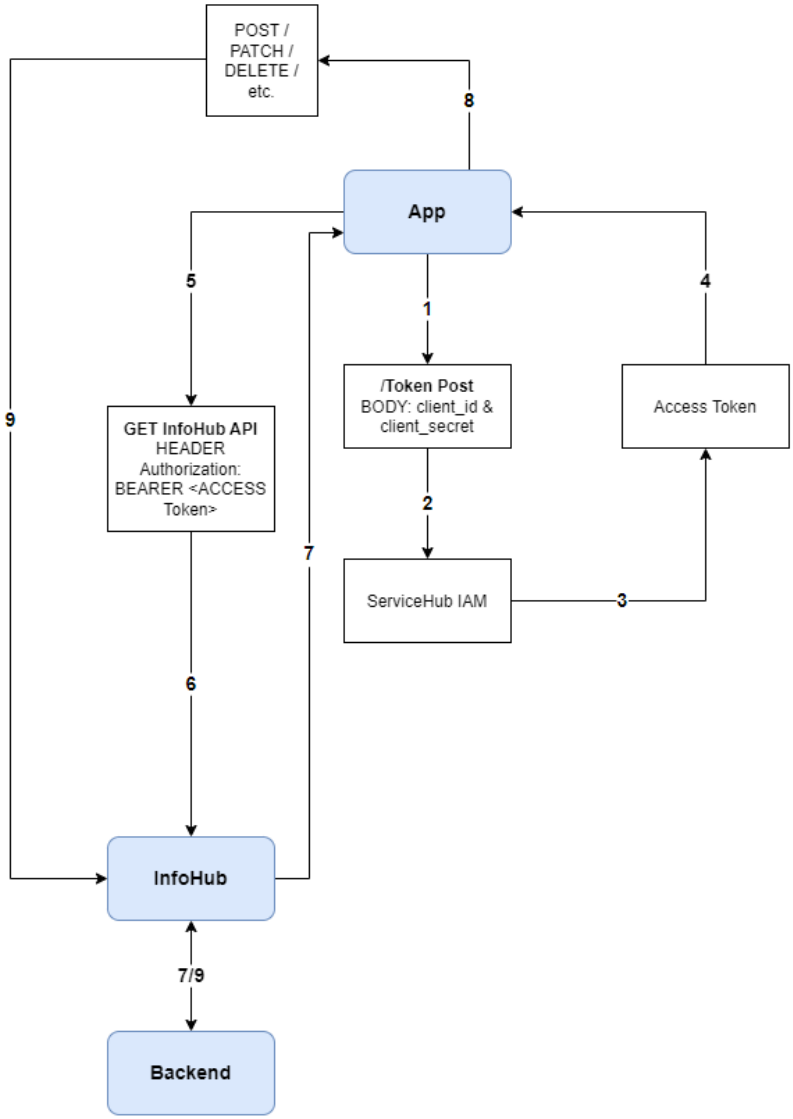
It shows that the user has been authenticated by an organisation you trust (the OpenID provider), so you can trust the claims about the identity.

Also, your application can personalise the user's experience by using the claims about the user that are contained in the ID token. For example, you can display the user's name on the user interface or send them a message wishing them a happy birthday. The nice thing about this is that you don't have to make any additional requests, so you may get a small performance gain for your application.

The main enhancement of OpenID Connect is to enable end-user authentication by the ID token data structure. The ID token is a security token that contains information about the authentication of an end user by an authorisation server when using a client and possibly other requested information. The ID token is represented as a JSON Web Token (JWT).

2 Authentication process in IAM

The authentication process including detailed instructions is as follows



Step 1: App requests ACCESS token via token post with client_id and client_secret

Step 2: Request ends up at ServiceHub IAM

Step 3: ServiceHub IAM issues an ACCESS Token

Step 4: ACCESS token is transmitted to app

Step 5: App sends BEARER token to InfoHub via Authorisation HEADER

Step 6: BEARER token is validated at the InfoHub

Step 7: Response from back-end is sent back to app via InfoHub

Step 8: Sending data (POST, PATCH, DELETE, etc.) to InfoHub backend

Step 9: Transmitted data is sent to the backend via InfoHub

The communication with the backend is always via the InfoHub

3 ID Token Validation

If any of the validation procedures defined in this document fail, all operations requiring the incorrectly validated information HAVE TO be cancelled and the unvalidated information MUST NOT be used.

The client HAS TO validate the ID token of the token response. To this end, the client may split the ID token at the dot characters("."), take the second segment and decode it using base64url to obtain a JSON object containing the ID token claims that HAVE TO be validated as follows:

1. The Issuer Identifier for the OpenID Provider (typically determined during discovery) HAS TO exactly match the value of the issuer claim.
2. The client HAS TO verify that the aud (audience) claim contains its client_id value registered with the investigator identified as the audience by the issuer(issuer) claim. The ID token HAS TO be rejected if the ID token does not list the client as a valid audience or if it contains additional audiences that the client does not trust.
3. If the ID token contains multiple audiences, the client SHOULD check to see if an azp claim is present.
4. If an azp claim (authorised party) is present, the client SHOULD check that its client_id matches the claim value.
5. The current time HAS TO be before the time represented by the exp claim (possibly with a small margin to account for the time shift).
6. The iat claim can be used to reject tokens issued too far from the current time to limit the amount of time nonces must be stored to prevent attacks. The permitted range is customer-specific.
7. If the acr claim has been requested, the client SHOULD check whether the claim value asserted is appropriate. The meaning and processing of acr claim values is outside the scope of this document.
8. When a max_age request is made, the client SHOULD check the auth_time claim value and request re-authentication if it determines that too much time has elapsed since the last end-user authentication.
9. ID tokens MAY contain other claims. Any claims used that are not understood HAVE TO be ignored.
10. ID tokens HAVE TO be signed with JWS [JWS] and optionally both signed and encrypted with JWS [JWS] or JWE [JWE], ensuring authentication, integrity, non-repudiation and optionally confidentiality
11. If the ID token is encrypted, it HAS TO be signed and then encrypted, with the result being an interleaved JWT as defined in [JWT].
12. ID tokens DO NOT HAVE TO use a value as an alg unless the response type used does not return an ID token from the authorisation endpoint (e.g. when using the authorisation code flow) and the client has explicitly requested the use of none during registration.
13. ID tokens SHOULD NOT use the JWS or JWE header parameter fields x5u, x5c, jku or jwk. Instead, the references to the keys used are communicated in advance using discovery and registration parameters.

4 Examples

4.1 Endpoints

	DEV	STAGING	VPROD	PROD
Issuer	https://servicehub-iam-dev.oebb.at/	https://servicehub-iam-staging.oebb.at/	https://servicehub-iam-vprod.oebb.at/	https://servicehub-iam.oebb.at/
JWKS Certs	https://servicehub-iam-dev.oebb.at/realms/bahnit/protocol/openid-connect/certs	https://servicehub-iam-staging.oebb.at/realms/bahnit/protocol/openid-connect/certs	https://servicehub-iam-vprod.oebb.at/realms/bahnit/protocol/openid-connect/certs	https://servicehub-iam.oebb.at/realms/bahnit/protocol/openid-connect/certs
Well-known	https://servicehub-iam-dev.oebb.at/realms/bahnit/.well-known/openid-configuration	https://servicehub-iam-staging.oebb.at/realms/bahnit/.well-known/openid-configuration	https://servicehub-iam-vprod.oebb.at/realms/bahnit/.well-known/openid-configuration	https://servicehub-iam.oebb.at/realms/bahnit/.well-known/openid-configuration
Account	https://servicehub-iam-dev.oebb.at/realms/bahnit/account/	https://servicehub-iam-staging.oebb.at/realms/bahnit/account/	https://servicehub-iam-vprod.oebb.at/realms/bahnit/account/	https://servicehub-iam.oebb.at/realms/bahnit/account/
Login	https://servicehub-iam-dev.oebb.at/realms/bahnit/protocol/openid-connect/auth	https://servicehub-iam-staging.oebb.at/realms/bahnit/protocol/openid-connect/auth	https://servicehub-iam-vprod.oebb.at/realms/bahnit/protocol/openid-connect/auth	https://servicehub-iam.oebb.at/realms/bahnit/protocol/openid-connect/auth
Token	https://servicehub-iam-dev.oebb.at/realms/bahnit/protocol/openid-connect/token	https://servicehub-iam-staging.oebb.at/realms/bahnit/protocol/openid-connect/token	https://servicehub-iam-vprod.oebb.at/realms/bahnit/protocol/openid-connect/token	https://servicehub-iam.oebb.at/realms/bahnit/protocol/openid-connect/token
Userinfo	https://servicehub-iam-dev.oebb.at/realms/bahnit/protocol/openid-connect/userinfo	https://servicehub-iam-staging.oebb.at/realms/bahnit/protocol/openid-connect/userinfo	https://servicehub-iam-vprod.oebb.at/realms/bahnit/protocol/openid-connect/userinfo	https://servicehub-iam.oebb.at/realms/bahnit/protocol/openid-connect/userinfo
Intro-spection	https://servicehub-iam-dev.oebb.at/realms/bahnit/protocol/openid-connect/token/introspect	https://servicehub-iam-staging.oebb.at/realms/bahnit/protocol/openid-connect/token/introspect	https://servicehub-iam-vprod.oebb.at/realms/bahnit/protocol/openid-connect/token/introspect	https://servicehub-iam.oebb.at/realms/bahnit/protocol/openid-connect/token/introspect
Logout	https://servicehub-iam-dev.oebb.at/realms/bahnit/protocol/openid-connect/logout	https://servicehub-iam-staging.oebb.at/realms/bahnit/protocol/openid-connect/logout	https://servicehub-iam-vprod.oebb.at/realms/bahnit/protocol/openid-connect/logout	https://servicehub-iam.oebb.at/realms/bahnit/protocol/openid-connect/logout

4.2 Access Token


```
{
    "sub": "a38bde5d-caba-4625-89be-8cf1f7f27d7d",
    "realm_roles": {
    "roles": [
        "IAM_INTERN_USER",
        "offline_access",
        "uma_authorization",
        "default-roles-bahnit"
    ]
},
"client_roles": {
    "mama_web.os.mama-os.oebb.at": {
        "roles": [
            "EVUAdministrator"
        ]
},
    "account": {
        "roles": [
            "manage-account",
            "manage-account-links",
            "view-profile"
        ]
    }
},
    "email_verified": false,
    "user_info": {
        "email_verified": false,
        "last_name": "Brechelmacher",
        "first_name": "Manuel",
        "email": "manuel.brechelmacher@oebb.at"
    },
"access_attributes": {
    "company_id": "00006",
    "user_id": "z177540"
},
"name": "Manuel Brechelmacher",
"preferred_username": "z177540",
"given_name": "Manuel",
"family_name": "Brechelmacher",
"email": "manuel.brechelmacher@oebb.at"
}
```

4.3 ID Token


```
{
  "exp": 1689679850,
  "iat": 1689679550,
  "auth_time": 0,
  "jti": "39ab844a-acf4-46c9-a995-ef45b7551e0c",
  "iss": "https://servicehub-iam.oebb.at/realms/bahnit",
  "aud": "sfit_web.sfit.oebb.at",
  "sub": "a38bde5d-caba-4625-89be-8cf1f7f27d7d",
  "typ": "ID",
  "azp": "sfit_web.sfit.oebb.at",
  "session_state": "d947f572-8e5a-4e4f-885f-48c6ee1b8d7f",
  "sid": "d947f572-8e5a-4e4f-885f-48c6ee1b8d7f",
  "realm_roles": {
    "roles": [
      "IAM_INTERN_USER",
      "offline_access",
      "uma_authorization",
      "default-roles-bahnit"
    ]
  },
  "client_roles": {
    "mama_web.os.mama-os.oebb.at": {
      "roles": [
        "EVUAdministrator"
      ]
    },
    "account": {
      "roles": [
        "manage-account",
        "manage-account-links",
        "view-profile"
      ]
    }
  },
  "email_verified": false,
  "access_attributes": {
    "company_id": "00006",
    "user_id": "z177540"
  },
  "name": "Manuel Brechelmacher",
  "preferred_username": "z177540",
  "given_name": "Manuel",
  "family_name": "Brechelmacher",
  "email": "manuel.brechelmacher@oebb.at"
}
```

4.4 Userinfo


```
{
  "sub": "a38bde5d-caba-4625-89be-8cf1f7f27d7d",
  "realm_roles": {
    "roles": [
      "IAM_INTERN_USER",
      "offline_access",
      "uma_authorization",
      "default-roles-bahnit"
    ]
  },
  "client_roles": {
    "mama_web.os.mama-os.oebb.at": {
      "roles": [
        "EVUAdministrator"
      ]
    },
    "account": {
      "roles": [
        "manage-account",
        "manage-account-links",
        "view-profile"
      ]
    }
  },
  "email_verified": false,
  "user_info": {
    "email_verified": false,
    "last_name": "Brechelmacher",
    "first_name": "Manuel",
    "email": "manuel.brechelmacher@oebb.at"
  },
  "access_attributes": {
    "company_id": "00006",
    "user_id": "z177540"
  },
  "name": "Manuel Brechelmacher",
  "preferred_username": "z177540",
  "given_name": "Manuel",
  "family_name": "Brechelmacher",
  "email": "manuel.brechelmacher@oebb.at"
}
```

4.5 Refresh Token


```
{
    "exp": 1689845717,
    "iat": 1689842117,
    "jti": "5af16e70-d281-4529-b3c1-4bb05ca96db1",
    "iss": "https://servicehub-iam-dev.oebb.at/realms/bahnit",
    "aud": "https://servicehub-iam-dev.oebb.at/realms/bahnit",
    "sub": "dd1371ad-58f4-4464-b8a6-90e5e7e32799",
    "typ": "Refresh",
    "azp": "test_service.postman.test.oebb.at",
    "session_state": "c28f72c1-0c43-4f9d-82ef-5e053a050d33",
    "scope": "openid profile",
    "sid": "c28f72c1-0c43-4f9d-82ef-5e053a050d33",
    "client_id": "test_service.postman.test.oebb.at",
    "username": "z177540",
    "active": true
}
```


4.6 Postman

To make it easier for you to get started with the respective flows, we have provided a ready-made collection (see attachment)

If you have any questions, please contact the technology management.

4.7 SoapUI / ReadyAPI

To make it easier for you to get started with the respective flows, we have provided a ready-made collection (see attachment)

If you have any questions, please contact the technology management.