Authentifizierung über Client ID – Anleitung für Entwickler

Bahn-IT ServiceHub

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

Inhalt

1 Wie verwendet IAM den Token? 3

2 Authentifizierungsablauf in IAM 4

3 ID Token Validierung 5

4 Beispiele 6

4.1 Endpunkte 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 Wie verwendet IAM den Token?

Vorweg: IAM steht für „Identity and Access Management und fokussiert auf Verwaltung, Authentifizierung und Autorisierung von externen Usern für Anwendungen von INFRA BS. Die entsprechende Client-ID samt Secret Client kann beim zuständigen Technologiemanager angefordert werden.

Ein ID-Token ist ein Artefakt, dass beweist, dass der Benutzer authentifiziert wurde. Es wurde von OpenID Connect (OIDC) eingeführt, einen offenen Standard für die Authentifizierung, der von vielen Identitätsanbietern verwendet wird.

Ein ID-Token ist als JSON-Web-Token (JWT) kodiert, ein Standardformat, das es Ihrer Anwendung ermöglicht, den Inhalt leicht zu überprüfen und sicherzustellen, dass er vom erwarteten Aussteller stammt und nicht von jemand anderem geändert wurde.

Einfach ausgedrückt sieht ein Beispiel für ein ID-Token wie folgt aus:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi8vbXktZG9tYWluLmF1d
GgwLmNvbSIsInN1YiI6ImF1dGgwfDEyMzQ1NiIsImF1ZCI6IjEyMzRhYmNkZWYiLCJleHAiOjE
zMTEyODE5NzAsImlhdCI6MTMxMTI4MDk3MCwibmFtZSI6IkphbmUgRG9lIiwiZ2l2ZW5fbmFtZ
SI6IkphbmUiLCJmYW1pbHlfbmFtZSI6IkRvZSJ9.bq1-
jxlG9B_bielkqOnjTY9Di9FillFb6IMQINXoYsw
```

Natürlich ist dies für das menschliche Auge nicht lesbar, so dass Sie es entschlüsseln müssen, um zu sehen, welchen Inhalt das JWT enthält. Das ID-Token ist übrigens nicht verschlüsselt, sondern nur Base 64-kodiert. Sie können eine der vielen verfügbaren Bibliotheken verwenden, um es zu entschlüsseln, oder Sie können es selbst mit dem JWT.IO Debugger untersuchen.

Diese JSON-Eigenschaften werden als Claims bezeichnet und sind Erklärungen über den Benutzer und das Token selbst. Die Angaben über den Benutzer definieren die Identität des Benutzers.

In den OpenID Connect-Spezifikationen ist es nicht erforderlich, dass das ID-Token Angaben über den Benutzer enthält. In seiner minimalen Struktur enthält es keine Daten über den Benutzer, sondern nur Informationen über den Authentifizierungsvorgang.

Das ID-Token kann zusätzliche Informationen über den Benutzer enthalten, wie z. B. seine E-Mail-Adresse, sein Foto, seinen Geburtstag usw.

Schließlich, und das ist vielleicht das Wichtigste, wird der ID-Token vom Aussteller mit seinem privaten Schlüssel signiert. Dies garantiert Ihnen die Herkunft des Tokens und stellt sicher, dass er nicht manipuliert wurde. Sie können diese Dinge mit dem öffentlichen Schlüssel des Ausstellers überprüfen.

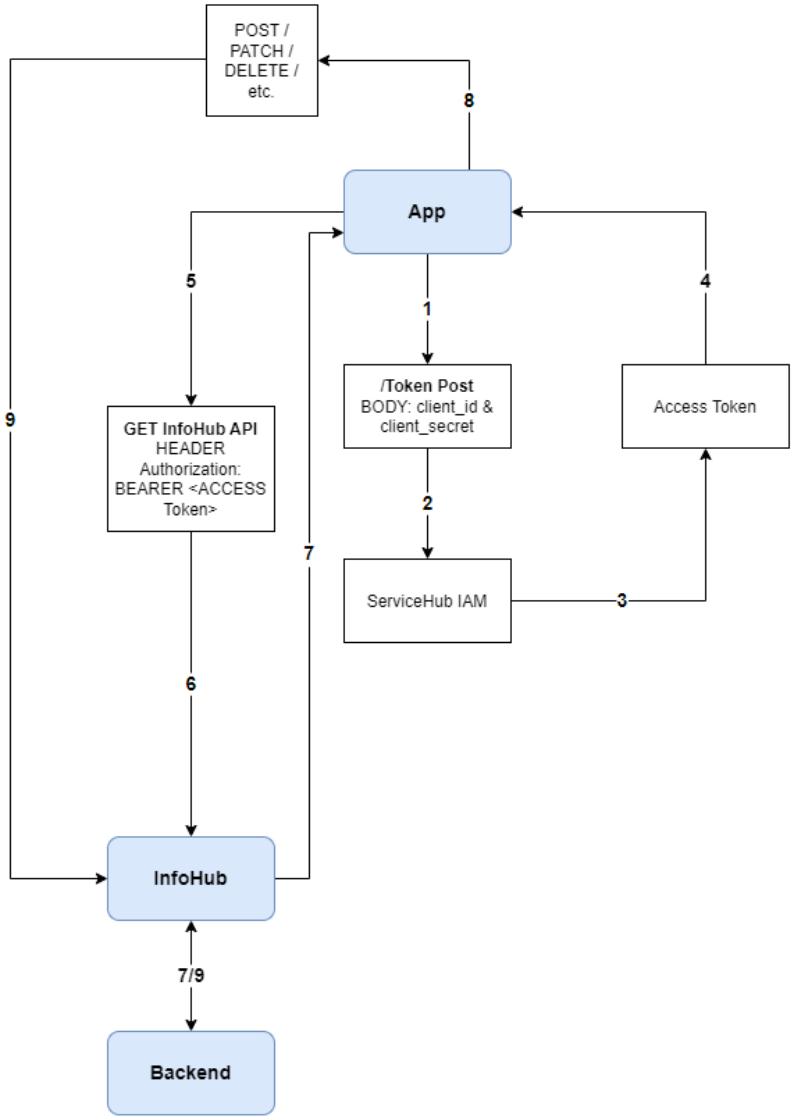
Erstens zeigt es, dass der Benutzer von einer Einrichtung, der Sie vertrauen (dem OpenID-Anbieter), authentifiziert wurde, so dass Sie den Claims über seine Identität vertrauen können.

Außerdem kann Ihre Anwendung das Erlebnis des Benutzers personalisieren, indem sie die Claims über den Benutzer verwendet, die im ID-Token enthalten sind. So können Sie beispielsweise den Namen des Benutzers auf der Benutzeroberfläche anzeigen oder ihm zum Geburtstag eine Nachricht mit den besten Wünschen schicken. Das Schöne daran ist, dass Sie keine zusätzlichen Anfragen stellen müssen, so dass Sie möglicherweise einen kleinen Leistungsgewinn für Ihre Anwendung erzielen.

Die wichtigste Erweiterung, die OpenID Connect gegenüber OAuth 2.0 vornimmt, um die Authentifizierung von Endbenutzern zu ermöglichen, ist die ID-Token-Datenstruktur. Das ID-Token ist ein Sicherheits-Token, das Angaben über die Authentifizierung eines Endbenutzers durch einen Autorisierungsserver bei Verwendung eines Clients und möglicherweise andere angeforderte Angaben enthält. Das ID-Token wird als JSON-Web-Token (JWT) [JWT] dargestellt.

2 Authentifizierungsablauf in IAM

Darstellung des Authentifizierungsablaufs als Flussdiagramm und Detailanleitung:



Schritt 1: App fordert via Token Post mit client_id und client_secret ACCESS Token an

Schritt 2: Anfrage landet bei ServiceHub IAM

Schritt 3: ServiceHub IAM stellt ACCESS Token aus

Schritt 4: ACCESS Token wird an App übermittelt

Schritt 5: App schickt BEARER Token zum InfoHub über Authorization HEADER

Schritt 6: BEARER Token wird am InfoHub validiert

Schritt 7: Response von Backend wird via InfoHub an App zurückgeschickt

Schritt 8: Senden von Daten (POST, PATCH, DELETE, etc.) an InfoHub Backend

Schritt 9: übermittelte Daten werden über InfoHub an das Backend geschickt

Der Datenabgleich erfolgt immer via InfoHub mit dem Backend.

3 ID Token Validierung

Wenn einer der in diesem Dokument definierten Validierungsverfahren fehlschlägt, MÜSSEN alle Operationen, die die nicht korrekt validierten Informationen erfordern, abgebrochen werden, und die nicht validierten Informationen dürfen NICHT verwendet werden.

Der Client MUSS das ID-Token in der Token-Antwort validieren. Zu diesem Zweck kann der Client das ID-Token an den Punktzeichen(„.“) aufteilen, das zweite Segment nehmen und es mit base64url dekodieren, um ein JSON-Objekt zu erhalten, das die ID-Token-Ansprüche enthält, die wie folgt validiert werden MÜSSEN:

1. Der Issuer Identifier für den OpenID Provider (der typischerweise während der Discovery ermittelt wird) MUSS genau mit dem Wert des issuer (Ausgeber) Claims übereinstimmen.
2. Der Kunde MUSS überprüfen, ob der aud (audience) Claim seinen client_id-Wert enthält, der bei dem durch den issuer(issuer) Claim als Publikum identifizierten Ermittenten registriert ist. Das ID-Token MUSS zurückgewiesen werden, wenn das ID-Token den Client nicht als gültiges Publikum aufführt oder wenn es zusätzliche Audiences enthält, denen der Client nicht vertraut.
3. Wenn das ID-Token mehrere Zielgruppen enthält, SOLLTE der Client überprüfen, ob ein azp-Claim vorhanden ist.
4. Wenn ein azp-Claim (autorisierte Partei) vorhanden ist, SOLLTE der Client überprüfen, ob seine client_id dem Claim-Wert entspricht.
5. Die aktuelle Zeit MUSS vor der Zeit liegen, die durch den exp-Anspruch repräsentiert wird (möglicherweise mit einem kleinen Spielraum, um die Zeitverschiebung zu berücksichtigen).
6. Der iat-Claim kann verwendet werden, um Token zurückweisen, die zu weit von der aktuellen Zeit entfernt ausgestellt wurden, um die Zeitspanne zu begrenzen, die Nonces gespeichert werden müssen, um Angriffe zu verhindern. Der zulässige Bereich ist kundenspezifisch.
7. Wenn der acr Claim angefordert wurde, SOLLTE der Client prüfen ob der geltend gemachte Claim-Wert angemessen ist. Die Bedeutung und Verarbeitung von acr-Claim-Werten liegt außerhalb des Rahmens dieses Dokuments.
8. Wenn eine max_age-Anfrage gestellt wird, SOLLTE der Client den auth_time-Anspruchswert überprüfen und eine erneute Authentifizierung anfordern, wenn er festellt, dass seit der letzten Endbenutzer-Authentifizierung zu viel Zeit verstrichen ist.
9. ID-Tokens KÖNNEN andere Claims enthalten. Alle verwendeten Claims, die nicht verstanden werden, MÜSSEN ignoriert werden.
10. ID-Token MÜSSEN mit JWS [JWS] signiert und optional sowohl signiert als auch mit JWS [JWS] bzw. JWE [JWE] verschlüsselt werden, wodurch Authentifizierung, Integrität, Nichtabstreitbarkeit und optional Vertraulichkeit gemäß Abschnitt 16.14 gewährleistet werden.
11. Wenn das ID-Token verschlüsselt ist, MUSS es signiert und dann verschlüsselt werden, wobei das Ergebnis ein verschachteltes JWT ist, wie in [JWT] definiert.
12. ID-Tokens MÜSSEN KEINEN Wert als alg verwenden, es sei denn, der verwendete Antworttyp gibt kein ID-Token vom Autorisierungsendpunkt zurück (z. B. bei Verwendung des Autorisierungscodeflusses) und der Client hat bei der Registrierung ausdrücklich die Verwendung von none angefordert.
13. ID-Tokens SOLLTEN NICHT die JWS- oder JWE-Header-Parameterfelder x5u, x5c, jku oder jwk verwenden. Stattdessen werden die Referenzen auf die verwendeten Schlüssel im Voraus mit Hilfe von Discovery- und Registrierungsparametern

4 Beispiele

4.1 Endpunkte

	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

Für den einfacheren Einstieg in die jeweiligen Flows haben wir euch hier eine fertige Collection bereitgestellt zum Download.

Bei Fragen wenden Sie sich ans Technologiemanagment (siehe ServiceHub Kontakte)

4.7 SoapUI / ReadyAPI

Für den einfacheren Einstieg in die jeweiligen Flows haben wird auch ein fertiges Projekt zum Download bereitgestellt.

Bei Fragen wenden Sie sich ans Technologiemanagment (siehe ServiceHub Kontakte)