# Wie wird die Impersonation Schnittstelle genutzt?

Tolleiv Nietsch - 2025-12-17 - How To ...

## Überblick

Die Impersonation-Schnittstelle von Bare.ID ermöglicht Administrator:innen oder autorisierten Service-Accounts, sich im Namen eines anderen Benutzers an einer spezifischen Applikation anzumelden. Diese Funktion ist hilfreich, um Supportfälle nachzuvollziehen, Benutzersitzungen zu prüfen oder administrative Aufgaben im Benutzerkontext auszuführen — ohne das Passwort des Nutzers kennen zu müssen.

Die Authentifizierung erfolgt über einen gültigen Access-Token eines berechtigten Kontos.

## Voraussetzungen: Zugriff auf einen gültigen Access-Token

Bevor eine Impersonation ausgeführt werden kann, muss ein gültiger Access-Token vorliegen, der den Ausführenden zur Impersonation berechtigt. Dieser wird über den Bare.ID-Token-Endpunkt bezogen. Die vollständige Dokumentation zur Authentifizierung findet sich hier: API-Authentifizierung – Bare.ID Handbuch

Als notwendige Berechtigung des Access-Tokens muss die Client-Role `impersonation` der System-Applikation `realm-management` dem Nutzer zugeordnet sein.

## Technischer Ablauf der Impersonation

Der Ablauf besteht aus wenigen zentralen Schritten:

1. Erzeugen/Anfragen der Impersonation inkl. Token über die API
2. Bereitstellung eines Links im Browser des Nutzers der den Impersonation-Vorgang initiiert
3. Starten einer Session im Browser mittels HTTP Redirect-Aufruf und Weiterleitung in die Anwendung

### 1. Anfordern des Impersonation-Tokens

Die Anfrage zur Impersonation erfolgt über die Bare.ID-API:

```
curl -X 'POST' \
'https://api.bare.id/user/v1/[INSTANCE_UUID]/impersonation-token?userUuid=[USER_UUID]&
clientId=[CLIENT_ID]' \
-H 'accept: application/json' \
-H "authorization: Bearer [ACCESS_TOKEN]" \
-d ''
```

Dabei ist die `USER_UUID` die UUID desjenigen Nutzers, in dessen Namen ich eine Applikation verwenden möchte. Die `CLIENT_ID` identifiziert diejenige Applikation, in der Aufrufende sich im Namen des Dritten einloggen möchte.

Die Antwort enthält ein JSON-Objekt mit Token und Ziel-URL:

```
{
  "token": "abc123...",
  "url": "https://[BASE_URL]/impersonation"
}
```

### 2. Aufbau der Browser-Session

Um im Browser eine gültige Session im Namen eines Dritten zu erzeugen, muss in diesem Browser-Kontext der erhaltene Token an die ebenso erhaltene Impersonation URL übergeben werden.

Dies kann per GET als Query-Parameter oder im POST-Body application/x-www-form-urlencoded erfolgen:

```
curl --verbose "$IMPERSONATION_URL?token=$IMPERSONATION_TOKEN"
```

oder

```
curl --verbose --data "token=$IMPERSONATION_TOKEN" -X POST "$IMPERSONATION_URL"
```

Der Server antwortet mit einem Redirect zur Ziel-Applikation und setzt dabei automatisch die passenden Session-Cookies. Nach dem Redirect kann die Session wie eine reguläre Benutzer-Anmeldung verwendet werden.

### Verwendung im Frontend

Wird die Impersonation im Browser oder einer webbasierten Anwendung genutzt, entsteht automatisch eine gültige Session im Kontext der Bare.ID Instanz. Dadurch können nachfolgende Aufrufe zu weiteren Applikationen dieselbe Sitzung verwenden, ohne dass ein erneuter Login erforderlich ist. Kritische Aktionen, wie zum Beispiel das Registrieren von MFA-Devices oder das Ändern des Passwortes sind jedoch nicht möglich.

## Typische Anwendungsfälle:

- Nachvollziehen von Support- oder Nutzerproblemen
- Testen von App-Funktionen im Benutzerkontext
- Administrative Kontrolle über Benutzer-Workflows

## Verantwortlichkeiten und Logging

Jede durchgeführte Impersonation wird im Audit-Log des betroffenen Nutzers erfasst. In der Aktivitätsübersicht erscheint der Vorgang als Admin-Login und zeigt den verantwortlichen “Impersonator” an.

Wenn die Impersonation über einen Service-Account eines Clients erfolgt, wird dieser Service-Account im Log als “Impersonator” aufgeführt. In diesem Fall ist es Aufgabe des Clients, interne Audit-Daten zu ergänzen, um nachvollziehen zu können, welcher tatsächliche Benutzer die Aktion ausgelöst hat.

## Hinweise

- Impersonation-Tokens sind kurzlebig (60s) und dürfen nicht gespeichert oder weitergegeben werden.
- Dieses Feature ist nicht zur dauerhaften Benutzer-Simulation oder zum Umgehen von Authentifizierungsprozessen gedacht.

## Weiterführende Links

- Bare.ID API-Dokumentation – user/v1
- Bare.ID Handbuch – Nutzerkonten-Management
- Bare.ID Handbuch – API-Authentifizierung