How is the Impersonation API Used?

Tolleiv Nietsch - 2026-03-12 - How To ...

Overview

The Bare.ID impersonation API allows administrators or authorized service accounts to sign in to a specific application on behalf of another user. This feature is useful for reproducing support cases, inspecting user sessions, or performing administrative tasks in the user context — without the need of knowing the user's password.

Authentication is performed via a valid access token of an authorized account.

Prerequisites: Access to a Valid Access Token

Before impersonation can be carried out, a valid access token must be available that authorizes the caller to perform impersonation. This is obtained via the Bare.ID token endpoint. The full documentation on authentication can be found here: API Authentication

As a required permission, the access token must include the client role `impersonation` of the system application `realm-management` assigned to the user.

Technical Flow of Impersonation

The flow consists of a few central steps:

1. Create/request the impersonation including token via the API
2. Provide a link in the user's browser context that initiates the impersonation process
3. Start a session in the browser via an HTTP redirect call and forward the user into the application

1. Requesting the Impersonation Token

The impersonation request is made via the 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 ''
```

Here, `USER_UUID` is the UUID of the user on whose behalf you want to use an application. `CLIENT_ID` identifies the application into which the caller wants to sign in on behalf of the third party.

The response contains a JSON object with token and target URL:

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

2. Establishing the Browser Session

To create a valid session in the browser on behalf of a third party, the received token must be passed to the likewise received impersonation URL in that browser context.

This can be done via GET as a query parameter or via POST in a `application/x-www-form-urlencoded` body:

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

or

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

The server responds with a redirect to the target application and automatically sets the appropriate session cookies. After the redirect, the session can be used like a regular user login.

Usage in the Frontend

When impersonation is used in the browser or a web-based application, a valid session in the context of the Bare.ID instance is automatically created. Subsequent calls to other applications can reuse the same session without requiring an additional login. However, critical actions such as registering MFA devices or changing the password are not possible.

Typical Use Cases

- Reproducing support or user issues
- Testing app features in the user context
- Administrative control over user workflows

Responsibilities and Logging

Every impersonation performed is recorded in the affected user’s audit log. In the activity overview, the process appears as an admin login and shows the responsible “impersonator”.

If impersonation is performed via a client’s service account, this service account is listed as the “impersonator” in the log. In this case, it is the client’s responsibility to add internal audit data to be able to trace which actual user triggered the action.

Notes

- Impersonation tokens are short-lived (60s) and must not be stored or shared.
- This feature is not intended for permanent user simulation or for bypassing authentication processes.

Further Information

- Bare.ID API-documentation – user/v1
- Bare.ID manual – user-management
- Bare.ID manual – API Authentication
- Bare.ID manual – OIDC & SAML