Connecting with Keycloak Admin Rest API

This guide explains how to authenticate with and use the Keycloak Admin REST API from a backend application. It walks through the necessary setup, authentication flow, and execution of a sample API request to list users in a realm.

Variables

Certain parameters must be provided to access the Keycloak Admin REST API successfully. Below is a breakdown of each required variable, its purpose, and where to find it. Here’s what each variable represents:

Variable	Description	Purpose
`BASE_URL`	The base URL of the Keycloak server (e.g., https://your-domain)	All admin API requests are made under this URL
`REALM`	The realm name used to obtain an admin access token	Typically "master" if accessing all realms, or your target realm
`CLIENT_ID`	The client ID configured for admin access (must have sufficient privileges)	Authenticates the backend to obtain an access token
`CLIENT_SECRET`	The client secret associated with the client	Required to authenticate confidential clients
`ADMIN_USERNAME`	A Keycloak admin user with the manage-users or admin role	Used in password grant to fetch an access token
`ADMIN_PASSWORD`	The password for the above admin user	Used with the username to authenticate

These values can be found in the **Keycloak Admin Console** under **Clients > [Your Admin Client]** and **Users > [Admin User]**.

Prerequisites

Install Node.js and NPM

Check if Node.js is installed:

```
node -v
```

Verify npm installation:

```
npm -v
```

Install Required Package

We'll use Axios to make HTTP requests. Install it with:

```
npm install axios
```

Code

Once all prerequisites are set up, create a new file named admin-api.js and add the following code:

```
const axios = require("axios");

const BASE_URL = "https://your-keycloak-domain";
const REALM = "master";
const CLIENT_ID = "admin-cli";
const ADMIN_USERNAME = "your-admin-username";
const ADMIN_PASSWORD = "your-admin-password";

async function getAccessToken() {
  const response = await axios.post(
    `${BASE_URL}/realms/${REALM}/protocol/openid-connect/token`,
    new URLSearchParams({
      client_id: CLIENT_ID,
      grant_type: "password",
      username: ADMIN_USERNAME,
      password: ADMIN_PASSWORD,
    }),
    {
      headers: {
        "Content-Type": "application/x-www-form-urlencoded",
      },
    }
  );
  return response.data.access_token;
}
```

```
async function listUsers() {
  try {
    const token = await getAccessToken();
    const response = await axios.get(
      `${BASE_URL}/admin/realms/${REALM}/users`,
      {
        headers: {
          Authorization: `Bearer ${token}`,
        },
      }
    );

    console.log("Users in realm:", response.data);
  } catch (err) {
    console.error("Failed to list users:", err.response?.data || err.message);
  }
}

listUsers();
```

Replace:

- BASE_URL with your Keycloak server base URL
- ADMIN_USERNAME and ADMIN_PASSWORD with your actual admin user credentials
- REALM with master (or a custom realm if you configured admin access)

Execution

Open the terminal and navigate to the directory where admin-api.js is saved. Once in the correct directory, run the script with the command:

```
node admin-api.js
```

If the connection is successful:

1. The script will authenticate using the password grant type
2. It will retrieve a valid admin access token
3. It will fetch and display the list of users in the specified realm

If an error occurs (such as a 401 unauthorized), double-check your admin credentials and client permissions.

Revision #1
Created 17 June 2025 10:08:26 by kaiwalya
Updated 17 June 2025 10:14:49 by kaiwalya