### 1.4.1 OAuth 2.0

With OAuth 2.0, sharing user data to third-party applications is easy, does not require sharing user credentials, and allows control over what data is shared. Four (4) roles defined in OAuth 2.0:

- **Resource owner**: The end user that owns resources that an application wants to access.
- **Resource server**: The service hosting the protected resources.
- **Client**: The application that would like to access the resource.
- **Authorization server**: The server issuing access to the client, which is the role of Keycloak.

In an OAuth 2.0 protocol flow, the client requests access to a resource on behalf of a resource owner from the authorization server. The authorization server issues limited access to the resource at the resource server by including access token in the request.

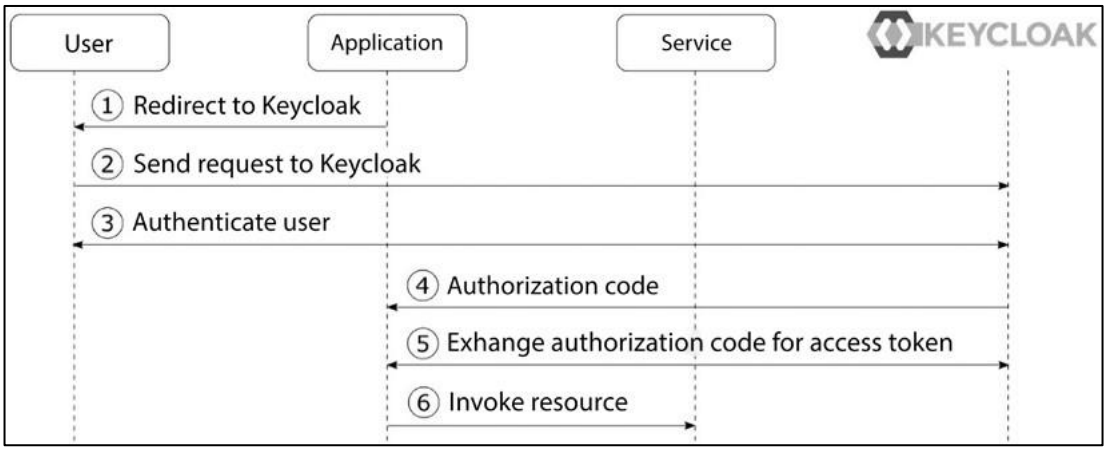


**Figure 2: Simplified OAuth 2.0 Authorization Code Grant Type**

The steps in the diagram are as follows:

1. The application sends an authentication request to the user's browser to be redirected to Keycloak for authorization.
2. The browser redirects the user to Keycloak’s authorization page.
3. If the user is not authenticated with Keycloak, Keycloak authenticates the user
4. The application gets an authorization code from Keycloak.
5. The application then exchanges the code for an access token from Keycloak.
6. The application uses the access token to access the protected resource.

Access tokens are passed around from the application to services, usually having a short lifetime. To get new access tokens without repeating the whole process, a refresh token is used.