

Connecting with Go

This guide explains how to establish a connection between a Go application and a Keycloak identity provider using the OIDC (OpenID Connect) protocol. It walks through the necessary setup, configuration, and execution of a basic login flow to authenticate users through Keycloak.

Variables

Certain parameters must be provided to integrate a Go application with Keycloak. Below is a breakdown of each required variable, its purpose, and where to find it. Here's what each variable represents:

Variable	Description	Purpose
<code>CLIENT_ID</code>	Client ID from the Keycloak Admin Console	Identifies the Go app in the Keycloak realm
<code>CLIENT_SECRET</code>	Secret from the Credentials tab of the client	Authenticates the Go app with Keycloak
<code>ISSUER_URL</code>	Realm URL (e.g., <code>https://your-domain/realms/your-realm</code>)	Base URL for OIDC discovery and validation
<code>REDIRECT_URL</code>	The callback URL Keycloak redirects to after successful login	Required to complete the OIDC flow

These values are found under **Clients > [Your Client] > Settings / Endpoints** in the Keycloak Admin Console.

Prerequisites

Install Go

Check if Go is installed:

```
go version
```

If not installed, download it from <https://golang.org/dl> and install.

Install Required Packages

Install the required Go packages:

```
go get github.com/coreos/go-oidc/v3
go get golang.org/x/oauth2
```

Code

Once all prerequisites are set up, create a new file named `main.go` and add the following code:

```
package main

import (
    "context"
    "fmt"
    "log"
    "net/http"

    "golang.org/x/oauth2"
    "golang.org/x/oauth2/clientcredentials"
    "golang.org/x/oauth2/endpoints"
    "github.com/coreos/go-oidc/v3/oidc"
)

var (
    clientID      = "CLIENT_ID"
    clientSecret  = "CLIENT_SECRET"
    redirectURL   = "http://localhost:8080/callback"
    issuerURL     = "https://your-keycloak-domain/realms/your-realm"
)

func main() {
    ctx := context.Background()

    provider, err := oidc.NewProvider(ctx, issuerURL)
    if err != nil {
        log.Fatalf("Failed to get provider: %v", err)
    }

    verifier := provider.Verifier(&oidc.Config{ClientID: clientID})
```

```

config := oauth2.Config{
    ClientID:     clientID,
    ClientSecret: clientSecret,
    Endpoint:     provider.Endpoint(),
    Scopes:       []string{oidc.ScopeOpenID, "profile", "email"},
    RedirectURL:  redirectURL,
}

http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    url := config.AuthCodeURL("state", oauth2.AccessTypeOffline)
    http.Redirect(w, r, url, http.StatusFound)
})

http.HandleFunc("/callback", func(w http.ResponseWriter, r *http.Request) {
    ctx := r.Context()
    if r.URL.Query().Get("state") != "state" {
        http.Error(w, "state mismatch", http.StatusBadRequest)
        return
    }

    oauth2Token, err := config.Exchange(ctx, r.URL.Query().Get("code"))
    if err != nil {
        http.Error(w, "failed to exchange token: "+err.Error(), http.StatusInternalServerError)
        return
    }

    rawIDToken, ok := oauth2Token.Extra("id_token").(string)
    if !ok {
        http.Error(w, "no id_token field in oauth2 token", http.StatusInternalServerError)
        return
    }

    idToken, err := verifier.Verify(ctx, rawIDToken)
    if err != nil {
        http.Error(w, "failed to verify ID Token: "+err.Error(), http.StatusInternalServerError)
        return
    }

    var claims map[string]interface{}
    if err := idToken.Claims(&claims); err != nil {

```

```
    http.Error(w, "failed to parse claims: "+err.Error(), http.StatusInternalServerError)
    return
  }

  fmt.Fprintf(w, "Login successful! User info:\n\n%v", claims)
}

log.Println("Server started at http://localhost:8080")
log.Fatal(http.ListenAndServe(":8080", nil))
}
```

Replace:

- CLIENT_ID and CLIENT_SECRET with your Keycloak client credentials
- https://your-keycloak-domain/realms/your-realm with your realm's base URL
- http://localhost:8080/callback should be registered in Keycloak's **Valid Redirect URIs**

Execute

1. Run the application with:

```
go run main.go
```

2. In your browser, navigate to:

```
http://localhost:8080
```

3. You will be redirected to the Keycloak login screen. After logging in:

- The app will redirect to /callback
- If successful, you'll see your decoded user info printed on the screen

Revision #1

Created 17 June 2025 09:28:02 by kaiwalya

Updated 17 June 2025 09:29:54 by kaiwalya