AuthFlow Provider Templates

This directory contains OAuth2 authentication templates for various providers.

Available Providers

1. GitHub (`github/oauth2.json`)

- **Type**: OAuth2
- **Scopes**: `user:email`
- **Features**: PKCE support, state validation
- **Setup**: Create a GitHub OAuth App in your GitHub settings

2. Slack (`slack/oauth2.json`)

- **Type**: OAuth2
- **Scopes**: `users:read,channels:read,chat:write`
- **Features**: Team information, user identity
- **Setup**: Create a Slack App in your Slack workspace

3. Google (`google/oauth2.json`)

- **Type**: OAuth2
- **Scopes**: `openid email profile`
- **Features**: OpenID Connect, offline access, ID tokens
- **Setup**: Create a project in Google Cloud Console

4. Microsoft (`microsoft/oauth2.json`)

- **Type**: OAuth2
- **Scopes**: `openid profile email User.Read`
- **Features**: Microsoft Graph API, Azure AD integration
- **Setup**: Register an app in Azure Portal

5. Discord (`discord/oauth2.json`)

- **Type**: OAuth2
- **Scopes**: `identify email`
- **Features**: User profile, guild information
- **Setup**: Create an application in Discord Developer Portal

6. Notion (`notion/oauth2.json`)

- **Type**: OAuth2
- **Scopes**: `read`
- **Features**: Workspace integration, bot access
- **Setup**: Create an integration in Notion

Usage

1. Configure Provider

Update the `config` section in each provider template:

```json
{
  "config": {
    "clientId": "YOUR_CLIENT_ID",
    "clientSecret": "YOUR_CLIENT_SECRET",
    "redirectUri": "http://localhost:8080/oauth/callback"
  }
}
```

2. Customize Scopes

Modify the `defaultScope` field to request specific permissions:

```json
{
  "config": {
    "defaultScope": "user:email,repo:read"
  }
}
```

3. Security Configuration

Adjust security policies as needed:

```json
{
  "policy": {
    "security": {
      "requirePkce": true,
      "requireState": true,
      "allowedRedirectDomains": ["yourdomain.com"]
    }
  }
}
```

Provider-Specific Notes

GitHub

- Uses `https://api.github.com/user` for user info
- Stores user login as `user_id`
- Supports PKCE for enhanced security

Slack

- Uses `https://slack.com/api/users.identity` for user info
- Stores team information in connection
- Requires specific scopes for different features

Google

- Uses `https://www.googleapis.com/oauth2/v2/userinfo` for user info
- Supports OpenID Connect with ID tokens
- Requires `access_type=offline` for refresh tokens

Microsoft

- Uses `https://graph.microsoft.com/v1.0/me` for user info
- Supports Azure AD multi-tenant authentication
- Requires specific scopes for Microsoft Graph API

Discord

- Uses `https://discord.com/api/users/@me` for user info
- Stores Discord user ID as `user_id`
- Supports guild (server) information

Notion

- Uses `https://api.notion.com/v1/users/me` for user info
- Stores workspace information
- Requires specific API version headers

Testing

To test a provider:

1. Start the AuthFlow server:

```
cargo run --example workflow_server_demo --features "server sqlite encryption"
```

2. Use the provider template in your workflow
3. Navigate to the authorization URL
4. Complete the OAuth flow
5. Check the connection store for the persisted connection

Adding New Providers

To add a new provider:

1. Create a new directory under `templates/providers/`
2. Create an `oauth2.json` file with the provider configuration
3. Follow the existing template structure
4. Update this README with provider information

Security Considerations

- Always use HTTPS in production
- Implement proper state validation
- Use PKCE when supported
- Regularly rotate client secrets
- Monitor for suspicious activity
- Implement rate limiting
- Use secure redirect URIs