OAuth 2.0 is an elaborate framework, which continuously evolves to address current needs and security considerations. The framework is even evolving into a consolidated OAuth 2.1 specification. This cheat sheet offers an overview of current security best practices for developers building OAuth 2.x client applications.

**OAuth 2.0 best practices for developers**

**Use the Authorization Code flow in every redirect scenario**

**Always use Proof Key for Code Exchange (PKCE)**
- The client includes a challenge based on a secret in Step 1
- The client includes the secret verifier in Step 10

**When using refresh tokens, apply additional protection**
- Rotate refresh tokens and act upon double use of a token
- Invalidate refresh tokens for web applications when...
  - the user explicitly logs out of the security token service
  - the user’s session with the security token service expires
- Invalidate refresh tokens when the user’s password changes

**Include an audience in the flow and in the access tokens**
- This restricts who accepts the access token in Step 12

**Restrict the capabilities of bearer access tokens**
- Keep the lifetime of access tokens as short as possible
- Use scopes to restrict the permissions associated with a token

**References**
- OAuth 2.0 threat model and security considerations
- OAuth 2.0 Security Best Current Practice
- The OAuth 2.1 Authorization Framework (draft)