## Oauth2

## Open ID

<https://medium.com/@darutk/full-scratch-implementor-of-oauth-and-openid-connect-talks-about-findings-55015f36d1c3>

<https://bitbucket.org/PEOFIAMP/phpoidc/src/60a3da2c357f54f9fd74e5499d9d716f8b94acdf/phpOp/index.php?at=default&fileviewer=file-view-default>

<https://github.com/OAuth2-Framework/oauth2-framework/blob/master/src/Component/Server/Endpoint/Authorization/ParameterChecker/PromptParameterChecker.php>

<http://iiw.idcommons.net/images/2/27/UnregisteredClientExtension.pdf>

client profile env client\_type application\_type

web application prod confidential web

dev native

user-agent-based application prod public web

dev native

native application \* public native

**Resource owner**

grantAccess(resource)

getAuthorization(client) : bool

**Resource server**

hasAccess(access token) : resource

validate(access token) : access token

isAccessTokenExpired() : bool

isScopeGranted(access token, requested scope) : bool

ClientFactory

Create(

**Client**

sendRequest(request, accessToken) : resource

isTlsDeployed () : bool

* **Registered client**

getRedirectionEndpoints() : redirection endpoint[]

* + **Confidential**
    - **Web application**
  + **Public**
    - **User-agent-based application**
    - **Native application**
* **Unregistered client**

**Dynamic configuration**

checkRedirectionUri(client, redirect\_uri) : redirection endpoint

**Authorization server**

issueAccessToken() : accessToken

authenticate() : resource owner

**Server**

getFlows()->getResponseTypes + none =

[code => openidAuthorizationCodeFlow, none => noneResponseType, idtoken token => open id implicit flow, token => implicit flow ]

**Credentials**

* **Authorization grant types**

getUri()

getResponseTypes() : response types[]

handleRequest(request) : result

getGrantType() : grant type

* + **Authorization code**

getResponseTypes() : [code]

* + **Implicit**

getResponseTypes() : [token]

* + **Resource owner password credentials**
  + **Client credentials**
  + **Authorization code**

getResponseTypes() : [code]

* + **Implicit**

getResponseTypes() : [id token, id token token]

* + **Hybrid**

getResponseTypes() : [code token, code id\_token, code id\_token token]

* **Tokens**

getLifetime()

* + **Authorization code**
  + **Access token**
  + **Refresh token**

**Grant type**

handleAccessTokenRequest(request) : response

**Response type**

handleResponseTypeRequest(request) : result

getDefaultResponseMode() : response mode

* **Code**
* **Token**
* **Id\_token**
* **None**

**Responde mode**

handle(result) : response

* **Query**
* **Fragment**
* **Form post**

**Server endpoint**

* **Authorization**

validateRequest(request) : bool

verifyResourceOwnerIdentity(resource owner)

authenticateResourceOwner() : resource owner

obtainAuthorizationGrant(authorization request) : authorization grant

obtainAuthorizationDecision(resource owner) : bool

isTls (request) : bool

getScope(request) : array

redirect(client redirection)

* **Token**

obtainAccessToken(authorization grant) : [access token, refresh token]

obtainAccessToken(refresh token) : [access token, refresh token]

isTls(request)

authenticateClient() : client

**Client endpoint**

* **Redirection : Uri**

isTls () : bool

isTlsAvailable() : bool

::fromErrorResponse(Uri, Error response) : static

* **Invalid endpoint**

**Client authentication**

* **Client password**

**Authorization request**

Response type

Client id

Redirect uri

State

**Authorization response**

State

**Error response**

Error

Error description

Error uri

**Token request**

Client id (optional)

Scope

Grant type

**Token response**

Access token

Token type

Expires in

Refresh token

Scope

**Access token scope**

* **Scope**

getScope(request) : array

getPolicy() : Policy

getResourceOwnerInstructions() : scope granted[]

* **Policy**
  + **Default**
  + **Error**
  + **Ignore**

**Access token type**

* **Bearer**