

## Instantiation

## Methods

# OpenID : OpenID consumer

The OpenID class is a OpenID consumer.

Namespace: Web

File location: lib/web/openid.php

## Instantiation

### Return class instance

```
$openid = new OpenID ();
```

The OpenID class extends the Magic (magic) class.

## Methods

### auth

#### Initiate OpenID authentication sequence; Return FALSE on failure or redirect to OpenID provider URL

```
bool auth ( [ string $proxy = NULL [ , array $attr = array() [ , string|array $reqd = NULL ] ] ] )
```

This function allows you to initiate an OpenID authentication sequence; Returns FALSE on failure or redirect to the OpenID provider URL.

- \$proxy
- \$attr
- \$reqd OpenID required fields can be declared as comma-separated string or array.

Example:

```
$openid->auth($proxy, $attr, $reqd); // returns TRUE or FALSE
```

### verified

#### Return TRUE if OpenID verification was successful

```
bool verified ( [ string $proxy = NULL ] )
```

This function allows you check if the OpenID verification was successful.

Example:

```
$openid->verified($proxy); // returns TRUE or FALSE
```

response

---

### Return OpenID response fields

```
array response ( )
```

This function allows you to return the OpenID response fields.

Example:

```
$openid_response = $openid->response();
```

exists

---

### Return TRUE if OpenID request parameter exists

```
bool exists ( string $key )
```

This function allows you to check if an OpenID request parameter exists.

Example:

```
$exists = $openid->exists($key); // returns TRUE or FALSE
```

set

---

### Bind value to OpenID request parameter

```
string set ( string $key, string $val )
```

This function allows you to bind a value to an OpenID request parameter.

Example:

```
echo $openid->set('openid.mode', 'checkid_setup'); // displays 'checkid_setup'
```

get

---

### Return value of OpenID request parameter

```
mixed get ( string $key )
```

This function allows you to retrieve the value of an OpenID request parameter.

Example:

```
$is_valid = $openid->get('is_valid'); // returns TRUE or FALSE
```

## clear

---

### Remove OpenID request parameter

```
NULL clear ( string $key )
```

This function allows you to remove OpenID request parameter

Example:

```
$openid->clear('openid.sig');
```

## discover

---

### Determine OpenID provider

```
protected string|FALSE discover ( string $proxy )
```

This *protected* method is used internally and allows to determine the OpenID provider.