

## Instantiation

## Authenticating

# Auth

The Auth class is used to authenticate user credentials (username / password) against an access data list (stored in a database table, etc).

Namespace: \

File location: lib/auth.php

---

## Instantiation

```
$auth = new \Auth (string $storage [, array $args = NULL ]);
```

You can map the class internal keys `id` and `pw` with your data table field names (i.e. 'username', 'password'). Given an example for F3's Jig (simple file-based db):

```
$db = new \DB\Jig('data/');  
$db_mapper = new \DB\Jig\Mapper($db, 'users');  
$auth = new \Auth($db_mapper, array('id' => 'username', 'pw' => 'password'));
```

Currently available authentication storage resources are:

- Jig
- SQL
- MongoDB
- LDAP
- SMTP

## SMTP adapter

---

Instantiation with SMTP as storage adapter has the following syntax:

```
$auth = new \Auth('smtp', [  
    'host' => 'smtp.domain.com',  
    'port' => 25,  
    'scheme' => null,  
]);
```

## Authenticating

## Login

---

### Used to test submitted credentials against an authentication data list

```
bool login ( string $id, string $pw [, string $realm = NULL ] )
```

The `login()` is the core method of the Auth class. By passing login credentials to it, it will authenticate the user against the auth objects authentication storage resource. Returns `true` on success, `false` on failure.

Example:

```
$db = new \DB\Jig('data/');  
$user = new \DB\Jig\Mapper($db, 'users');  
$auth = new \Auth($user, array('id'=>'user_id', 'pw'=>'password'));  
$login_result = $auth->login('admin','secret_pwd'); // returns true on successful log  
in
```

## Basic

---

### HTTP basic authentication mechanism

```
bool basic ( [ string $func = NULL ] )
```

The `basic()` method provides a way to authenticate a user without using webpage forms. The browser will display a login dialog box, prompting the user to enter their username and password credentials.

`$func` is the name of a callback function that can be used to manipulate the submitted password before it is compared against the password kept in the data storage resource, for example if you hash your passwords before storing them in your database. The submitted password is passed to your callback function as an argument.

```
$db = new \DB\SQL('mysql:host=localhost;dbname=project1', 'root', 'topsecret123');  
$user = new \DB\SQL\Mapper($db, 'users');  
$auth = new \Auth($user, array('id'=>'user_id', 'pw'=>'password'));  
$auth->basic(); // a network login prompt will display to authenticate the user
```

