Magic

The Magic class is a PHP magic wrapper and implements the ArrayAccess interface (http://php.net/manual/en/class.arrayaccess.php).

Namespace: `\`
File location: `lib/magic.php`

Instantiation

The Magic class is an abstract class (http://www.php.net/manual/en/language.oop5.abstract.php), which means that you need to extend it with another class, that then can take advantage of the magic wrapper.

It implements the ArrayAccess interface and wraps some magic methods (http://www.php.net/manual/en/language.oop5.magic.php) towards them. This way you can use an underlying object the same way you do it with any standard array. Let's have a look at this example:

```
class User extends Magic {

    protected $data;

    function exists($key) {
        return array_key_exists($key,$this->data);
    }

    function set($key, $val) {
        $this->data[$key] = $val;
    }

    function &get($key) {
        return $this->data[$key];
    }

    function clear($key) {
        unset($this->data[$key]);
    }
}
```

Now you can also access an user object like an Array or set the user data with objects properties and vice versa.

```
$user = new User();
$user->name = 'John';
$user['age'] = 28;
$user->set('mail','john@email.com');

echo $user['name']; // John
echo $user->get('age'); // 28
echo $user->mail; // john@email.com
```

Abstract Methods

The Magic class being abstract, your new descendant class using Magic need to implement all these functions:

exists

Return TRUE if key is not empty

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

This function returns TRUE if key is not empty

set

Bind value to key

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

This function allows you to bind a value to a key

get

Retrieve contents of key

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

This function allows you to retrieve the contents of a key

clear

Unset key

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

This function allows you to unset a key

Parent Methods

The following methods are needed for internal implementation. You don't need them in your magic child class. Stick to the abstract methods.

visible

Return TRUE if property has public visibility

```
private bool visible ( string $key )
```

This function returns TRUE if property has public visibility. This is important to decide if we call the magic setter/getter or just bypass the public property.

offsetexists

Convenient method for checking property value

```
mixed offsetexists ( string $key )
```

This function is a convenient method for retrieving the value of a property

__isset

Alias for offsetexists()

```
mixed __isset ( string $key )
```

This function is an alias for the offsetexists() function above

offsetset

Convenient method for assigning property value

```
mixed offsetset ( string $key, scalar $val )
```

This function is a convenient method for setting the value of a property

__set

Alias for offsetset()

```
mixed __set ( string $key, scalar $val )
```

This function is an alias for the offsetset() function above

offsetget

Convenient method for retrieving property value

```
mixed offsetget ( string $key )
```

This function is a convenient method for getting the value of a property

__get

Alias for offsetget()

```
mixed __get ( string $key )
```

This function is an alias for the offsetget() function above

offsetunset

Convenient method for checking property value

```
NULL offsetunset ( string $key )
```

This function is a convenient method for checking the value of a property

__unset

Alias for offsetunset()

```
NULL __unset ( string $key )
```

This function is an alias for the offsetunset() function above