

```
<?php
class Validation {

    // form object
    private $objForm;

    // for storing all error ids
    private $_error = array();

    // validation messages
    public $_message = array(
        "first_name"    => "Please provide your first name",
        "last_name"     => "Please provide your last name",
        "address_1"     => "Please provide the first line of your address",
        "address_2"     => "Please provide the second line of your address",
        "town"          => "Please provide your town name",
        "county"        => "Please provide your county name",
        "post_code"     => "Please provide your post code",
        "country"       => "Please select your country",
        "email"         => "Please provide your valid email address",
        "email_duplicate" => "This email address is already taken",
        "login"         => "Username and / or password were incorrect",
        "password"      => "Please choose your password",
        "confirm_password" => "Please confirm your password",
        "password_mismatch" => "Passwords did not match",
        "category"      => "Please select the category",
        "name"          => "Please provide a name",
        "description"   => "Please provide a description",
        "price"         => "Please provide a price",
        "name_duplicate" => "This name is already taken"
    );

    // list of expected fields
    public $_expected = array();

    // list of require fields
    public $_required = array();

    // list of special validation fields
    // array('field_name' => 'format')
    public $_special = array();

    // post array
    public $_post = array();

    // fields to be removed from the $_post array
    public $_post_remove = array();

    // fields to be specifically formatted
    // array('field_name' => 'format')
    public $_post_format = array();

    public function __construct($objForm) {
        $this->objForm = $objForm;
    }
}
```

```
public function process() {
    if ($this->objForm->isPost() && !empty($this->_required)) {
        // get only expected fields
        $this->_post = $this->objForm->getPostArray($this->_expected);
        if (!empty($this->_post)) {
            foreach($this->_post as $key => $value) {
                $this->check($key, $value);
            }
        }
    }
}

public function add2Errors($key) {
    $this->_errors[] = $key;
}

public function check($key, $value) {
    if (!empty($this->_special) && array_key_exists($key, $this->_special)) {
        $this->checkSpecial($key, $value);
    } else {
        if (in_array($key, $this->_required) && empty($value)) {
            $this->add2Errors($key);
        }
    }
}

public function checkSpecial($key, $value) {
    switch($this->_special[$key]) {
        case 'email':
            if (!$this->isEmail($value)) {
                $this->add2Errors($key);
            }
            break;
    }
}

public function isEmail($email = null) {
    if (!empty($email)) {
        $result = filter_var($email, FILTER_VALIDATE_EMAIL);
        return !$result ? false : true;
    }
    return false;
}
```

```
public function isValid() {
    $this->process();
    if (empty($this->_errors) && !empty($this->_post)) {
        // remove all unwanted fields
        if (!empty($this->_post_remove)) {
            foreach($this->_post_remove as $value) {
                unset($this->_post[$value]);
            }
        }
        // format all required fields
        if (!empty($this->_post_format)) {
            foreach($this->_post_format as $key => $value) {
                $this->format($key, $value);
            }
        }
        return true;
    }
    return false;
}

public function format($key, $value) {
    switch($value) {
        case 'password':
            $this->_post[$key] = Login::string2hash($this->_post[$key]);
            break;
    }
}

public function validate($key) {
    if (!empty($this->_errors) && in_array($key, $this->_errors)) {
        return $this->wrapWarn($this->_message[$key]);
    }
}

public function wrapWarn($mess = null) {
    if (!empty($mess)) {
        return "<span class=\"warn\">{$mess}</span>";
    }
}
}
```