

# Objektorientierung mit Models

== Erläuterung ==

Der Code ist zur Hilfenahme entstanden, als die Frage geklärt werden sollte ob man die Validierung im Controller oder im Model praktizieren sollte: [Überprüfung von Werten in \\$\\_POST Feldern](#). Mehr Informationen zum Basismodel sollen noch folgen. Jeder ist aber angehalten den Artikel zu verbessern.

Das hier erläuterte Verfahren nutzt object-relational mapping bzw Objektrelationale Abbildung - kurz ORM.

== Basismodel ==

## Quellcode

```
1. abstract class Model {  
2.     protected $primarykey = 'id';  
3.     protected $data = array();  
4.     abstract public function getColumns();  
5.     function __construct($data = array()) {  
6.         $this->data = $data;  
7.     }  
8.     public static function findById($id, $className) {  
9.         $sql = "SELECT *  
10.        FROM ".$this->getTablename()."  
11.       WHERE ".$this->primarykey." = :id ";  
12.       $stmt = MyDB::getInstance()->prepare($sql);  
13.       $stmt->execute(array(  
14.           ':id' => $id  
15.       ));  
16.       return new $className($stmt->fetch());  
17.   }  
18.   protected function _findById() {  
19.   }  
20.   public function __set($name, $value) {  
21.       $this->data[$name] = $value;  
22.   }  
23.   public function __get($name) {  
24.       return $this->data[$name];  
25.   }  
26.   /**  
27.    * empty() on a __get function will not work as expected if isset has not been overriden  
28.    */  
29.   public function __isset($key) {  
30.       if (isset($this->data[$key])) {  
31.           return (false === empty($this->data[$key]));  
32.       } else {  
33.           return null;  
34.       }  
35.   }  
36.   public function validate() {  
37.   }  
38.   public function getTablename() {  
39.       return get_class($this);  
40.   }  
41.   public function delete() {  
42.       $primaryval = $this->{$this->primarykey};  
43.       $sql = 'DELETE FROM '.$this->getTablename().' WHERE '.$this->primarykey.' = :primaryval';  
44.       $stmt = MyDB::getInstance()->prepare($sql);  
45.       $stmt->execute(array(  
46.           ':primaryval' => $primaryval  
47.       ));  
48.   }
```

```

58. ':primaryval' => $primaryval
59. });
60. }
61. public function save($override = null) {
62.     if($override !== null) {
63.         $this->data = array_merge($this->data, $override);
64.     }
65. }
66. $this->validate();
67. $params = array();
68. $primaryval = $this->{$this->primarykey};
69. // insert or update
70. if($primaryval === null) {
71.     $sql = 'INSERT INTO '.$this->getTablename().' ('.implode(',', $this->getColumns()).') VALUES (';
72.     foreach($this->getColumns() as $key) {
73.         $sql .= ':'.$key.',';
74.         $params[':'.$key] = $this->$key;
75.     }
76.     $sql = rtrim($sql,',');
77.     if(count($params)) {
78.         $stmt = MyDB::getInstance()->prepare($sql);
79.         $stmt->execute($params);
80.         // remember auto increment key
81.         $this->{$this->primarykey} = MyDB::getInstance()->lastInsertId();
82.     }
83. } else {
84.     $sql = 'UPDATE '.$this->getTablename().' SET ';
85.     foreach($this->getColumns() as $key) {
86.         $sql .= $key.' = :'.$key.',';
87.         $params[':'.$key] = $this->$key;
88.     }
89.     $sql = rtrim($sql,','). WHERE '.$this->primarykey.' = :primaryval';
90.     if(count($params)) {
91.         $params[':primaryval'] = $primaryval;
92.         $stmt = MyDB::getInstance()->prepare($sql);
93.         $stmt->execute($params);
94.     }
95. }
96. }
97. }
98. }
99. }
100. }

```

Alles anzeigen

== Beispielimplementierung eines Benutzers ==

### Quellcode

```

1. class User extends Model {
2.     public function getTablename() {
3.         return get_class($this); // user
4.     }
5.     public function getColumns() {
6.         return array('email', 'password');
7.     }
8.     public function validate() {
9.         if(...) {
10.             throw new ValidateException('message');
11.         }
12.     }
13. }

```

14. }

Alles anzeigen

== Beispiele ==

==== Neuen Benutzer anlegen ===

### Quellcode

```
1. <form method="post">
2. email: <input type="text" name="email" /><br/>
3. password: <input type="password" name="password" /><br/>
4. <input type="submit">
5. </form>
6. <?php
7. if(count($_POST)) {
8. $user = new User($_POST);
9. $user->save();
10. }
11. ?>
```

Alles anzeigen

==== Benutzer bearbeiten ===

### Quellcode

```
1. <form method="post" action="edit.php?userid=123">
2. email: <input type="text" name="email" /><br/>
3. password: <input type="password" name="password" /><br/>
4. <input type="submit">
5. </form>
6. <?php
7. if(count($_POST) && isset($_GET['userid'])) {
8. $user = User::findById($_GET['userid']);
9. $user->save($_POST);
10. }
11. ?>
```

Alles anzeigen