

DBA02

Erzeugt von Doxygen 1.8.5

Fre Aug 30 2013 09:39:10



# Inhaltsverzeichnis

<b>1</b>	<b>Hierarchie-Verzeichnis</b>	<b>1</b>
1.1	Klassenhierarchie . . . . .	1
<b>2</b>	<b>Klassen-Verzeichnis</b>	<b>3</b>
2.1	Auflistung der Klassen . . . . .	3
<b>3</b>	<b>Klassen-Dokumentation</b>	<b>5</b>
3.1	Controller\Admin\Base Klassenreferenz . . . . .	5
3.1.1	Ausführliche Beschreibung . . . . .	5
3.1.2	Beschreibung der Konstruktoren und Destrukturen . . . . .	5
3.1.2.1	__construct . . . . .	5
3.2	Model\Base Klassenreferenz . . . . .	6
3.2.1	Ausführliche Beschreibung . . . . .	6
3.2.2	Beschreibung der Konstruktoren und Destrukturen . . . . .	6
3.2.2.1	__construct . . . . .	6
3.3	Config Klassenreferenz . . . . .	6
3.3.1	Ausführliche Beschreibung . . . . .	7
3.3.2	Beschreibung der Konstruktoren und Destrukturen . . . . .	7
3.3.2.1	__construct . . . . .	7
3.3.3	Dokumentation der Elementfunktionen . . . . .	7
3.3.3.1	__get . . . . .	7
3.4	Database Klassenreferenz . . . . .	8
3.4.1	Ausführliche Beschreibung . . . . .	8
3.4.2	Beschreibung der Konstruktoren und Destrukturen . . . . .	9
3.4.2.1	__construct . . . . .	9
3.5	Controller\Database Klassenreferenz . . . . .	9
3.5.1	Ausführliche Beschreibung . . . . .	9
3.5.2	Beschreibung der Konstruktoren und Destrukturen . . . . .	9
3.5.2.1	__construct . . . . .	9
3.5.3	Dokumentation der Elementfunktionen . . . . .	10
3.5.3.1	CreateTable_Action . . . . .	10
3.5.3.2	Diagramm_Action . . . . .	10

3.6	FrontController Klassenreferenz	10
3.6.1	Ausführliche Beschreibung	10
3.6.2	Beschreibung der Konstruktoren und Destruktoren	11
3.6.2.1	__construct	11
3.6.3	Dokumentation der Elementfunktionen	11
3.6.3.1	run	11
3.7	Controller\Index Klassenreferenz	11
3.7.1	Ausführliche Beschreibung	11
3.7.2	Dokumentation der Elementfunktionen	12
3.7.2.1	Index_Action	12
3.8	CustomException\InvalidUserPassException Klassenreferenz	12
3.8.1	Ausführliche Beschreibung	12
3.9	Redirect Klassenreferenz	12
3.9.1	Ausführliche Beschreibung	12
3.9.2	Dokumentation der Elementfunktionen	13
3.9.2.1	Header	13
3.9.2.2	toController	13
3.10	Session Klassenreferenz	13
3.10.1	Ausführliche Beschreibung	14
3.10.2	Dokumentation der Elementfunktionen	14
3.10.2.1	__get	14
3.10.2.2	__set	14
3.10.2.3	authUser	14
3.10.2.4	destroy	15
3.10.2.5	isUserAuthed	15
3.10.2.6	isUserAuthedCheck	15
3.11	Controller\Admin\Stats Klassenreferenz	15
3.11.1	Ausführliche Beschreibung	16
3.11.2	Beschreibung der Konstruktoren und Destruktoren	16
3.11.2.1	__construct	16
3.11.3	Dokumentation der Elementfunktionen	16
3.11.3.1	Index_Action	16
3.12	Controller\Admin\Survey Klassenreferenz	17
3.12.1	Ausführliche Beschreibung	17
3.12.2	Beschreibung der Konstruktoren und Destruktoren	17
3.12.2.1	__construct	17
3.12.3	Dokumentation der Elementfunktionen	17
3.12.3.1	Add_POST_Action	17
3.12.3.2	Delete_Action	18
3.12.3.3	Index_Action	18

3.13 Controller\Survey Klassenreferenz	18
3.13.1 Ausführliche Beschreibung	19
3.13.2 Beschreibung der Konstruktoren und Destruktoren	19
3.13.2.1 __construct	19
3.13.3 Dokumentation der Elementfunktionen	19
3.13.3.1 Index_Action	19
3.13.3.2 Save_POST_Action	20
3.13.3.3 Show_Action	20
3.13.3.4 showSurveyResult	20
3.14 Model\Survey Klassenreferenz	20
3.14.1 Ausführliche Beschreibung	21
3.14.2 Dokumentation der Elementfunktionen	21
3.14.2.1 addSurvey	21
3.14.2.2 deleteSurvey	22
3.14.2.3 getStats	22
3.14.2.4 getSurveyItemCount	22
3.14.2.5 getSurveyItems	23
3.14.2.6 getSurveyName	23
3.14.2.7 getSurveyResult	23
3.14.2.8 getSurveys	24
3.14.2.9 getTableCount	24
3.14.2.10 saveNewAnswers	25
3.15 Model\User Klassenreferenz	26
3.15.1 Ausführliche Beschreibung	26
3.15.2 Dokumentation der Elementfunktionen	26
3.15.2.1 addUser	26
3.15.2.2 checkCredentials	27
3.15.2.3 deleteUser	27
3.15.2.4 getUsers	28
3.15.2.5 updateLastLogin	28
3.16 Controller\Admin\User Klassenreferenz	28
3.16.1 Ausführliche Beschreibung	29
3.16.2 Beschreibung der Konstruktoren und Destruktoren	29
3.16.2.1 __construct	29
3.16.3 Dokumentation der Elementfunktionen	29
3.16.3.1 Add_POST_Action	29
3.16.3.2 Delete_Action	29
3.16.3.3 Index_Action	30
3.17 CustomException\UserNotAuthedException Klassenreferenz	30
3.17.1 Ausführliche Beschreibung	30

3.18 Controller\UserSession Klassenreferenz . . . . .	31
3.18.1 Ausführliche Beschreibung . . . . .	31
3.18.2 Dokumentation der Elementfunktionen . . . . .	31
3.18.2.1 Login_POST_Action . . . . .	31
3.18.2.2 Logout_Action . . . . .	31
3.19 View Klassenreferenz . . . . .	31
3.19.1 Ausführliche Beschreibung . . . . .	32
3.19.2 Dokumentation der Elementfunktionen . . . . .	32
3.19.2.1 assign . . . . .	32
3.19.2.2 display . . . . .	32
3.19.2.3 getTemplateOnFilesystem . . . . .	33
3.19.2.4 setTemplate . . . . .	34
<b>Index</b>	<b>35</b>

# Kapitel 1

## Hierarchie-Verzeichnis

### 1.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

Controller\Admin\Base . . . . .	5
Controller\Admin\Stats . . . . .	15
Controller\Admin\Survey . . . . .	17
Controller\Admin\User . . . . .	28
Model\Base . . . . .	6
Model\Survey . . . . .	20
Model\User . . . . .	26
Config . . . . .	6
Controller\Database . . . . .	9
Exception	
CustomException\InvalidUserPassException . . . . .	12
CustomException\UserNotAuthedException . . . . .	30
FrontController . . . . .	10
Controller/Index . . . . .	11
PDO	
Database . . . . .	8
Redirect . . . . .	12
Session . . . . .	13
Controller\Survey . . . . .	18
Controller\UserSession . . . . .	31
View . . . . .	31





## Kapitel 2

# Klassen-Verzeichnis

### 2.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

Controller\Admin\Base	5
Model\Base	6
Config	6
Database	8
Controller\Database	9
FrontController	10
Controller/Index	11
CustomException\InvalidUserPassException	12
Redirect	12
Session	13
Controller\Admin\Stats	15
Controller\Admin\Survey	17
Controller\Survey	18
Model\Survey	20
Model\User	26
Controller\Admin\User	28
CustomException\UserNotAuthenticatedException	30
Controller\UserSession	31
View	31

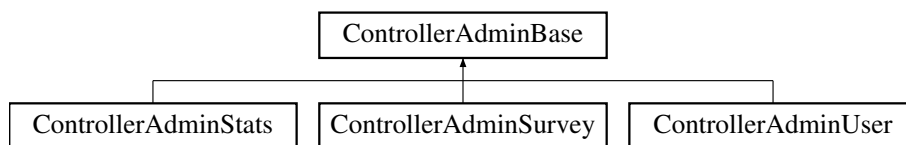


## Kapitel 3

# Klassen-Dokumentation

### 3.1 Controller\Admin\Base Klassenreferenz

Klassendiagramm für Controller\Admin\Base:



#### Öffentliche Methoden

- `__construct()`

#### Geschützte Attribute

- `$model`
- `$view`

#### 3.1.1 Ausführliche Beschreibung

Admin `Base` Controller

Definiert in Zeile 10 der Datei `Controller/Admin/Base.php`.

#### 3.1.2 Beschreibung der Konstruktoren und Destruktoren

##### 3.1.2.1 Controller\Admin\Base::\_\_construct ( )

Im Konstruktor generell ueberpruefen ob sich der `User` eingeloggt hat.

Definiert in Zeile 22 der Datei `Controller/Admin/Base.php`.

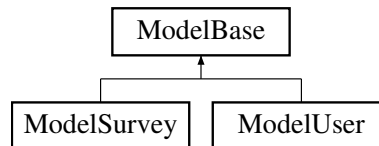
```
00022         {
00023
00024             \Session::isUserAuthedCheck();
00025
00026             $this->view = new \View();
00027
00028         }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Controller/Admin/Base.php

## 3.2 Model\Base Klassenreferenz

Klassendiagramm für Model\Base:



### Öffentliche Methoden

- [\\_\\_construct](#) ()

### Geschützte Attribute

- **\$dbh**

#### 3.2.1 Ausführliche Beschreibung

Basis Datenmodell

Definiert in Zeile 9 der Datei [Model/Base.php](#).

#### 3.2.2 Beschreibung der Konstruktoren und Destruktoren

##### 3.2.2.1 Model\Base::\_\_construct ( )

Konstruktor

Datenbank Verbindung herstellen.

Definiert in Zeile 21 der Datei [Model/Base.php](#).

```
00021                                     {
00022
00023             $this->dbh = new \Database();
00024
00025     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/Base.php

## 3.3 Config Klassenreferenz

### Öffentliche Methoden

- [\\_\\_get](#) (\$key)

## Öffentliche, statische Methoden

- static **getInstance** ()

## Private Methoden

- **\_\_construct** ()
- **\_\_clone** ()

## Private Attribute

- **\$config** = array()

## Statische, private Attribute

- static **\$instance** = null

### 3.3.1 Ausführliche Beschreibung

[Config](#) Klasse als Singleton Pattern

```
$conf = ::getInstance(); $value = $conf->key;
```

Die Werte stehen in der Ini Datei und koennen direkt abgerufen werden.

Definiert in Zeile [13](#) der Datei [Config.php](#).

### 3.3.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.3.2.1 Config::\_\_construct ( ) [private]

Konstruktor

Wird nur einmal aufgerufen wegen Singleton Pattern

Konfigurationsdatei einlesen

Definiert in Zeile [36](#) der Datei [Config.php](#).

```
00036                                     {
00037         $this->config = parse_ini_file(CONFIGFILE);
00038     }
```

### 3.3.3 Dokumentation der Elementfunktionen

#### 3.3.3.1 Config::\_\_get ( \$key )

Getter Methode

Parameter

String	<b>\$key</b>	Schlüssel
--------	--------------	-----------

## Rückgabe

String Wert

Definiert in Zeile 51 der Datei [Config.php](#).

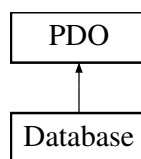
```
00051         {
00052
00053             if (array_key_exists($key, $this->config)) {
00054                 return $this->config[$key];
00055             } else {
00056                 throw new Exception("Config Schluessel '$key' nicht verfuegbar");
00057             }
00058         }
00059     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Config.php](#)

## 3.4 Database Klassenreferenz

Klassendiagramm für Database:



### Öffentliche Methoden

- [\\_\\_construct\(\)](#)

### Private Methoden

- [beMoreVerbose\(\)](#)

### Private Attribute

- [\\$dsn](#)
- [\\$user](#)
- [\\$pass](#)

### 3.4.1 Ausführliche Beschreibung

[Database](#) Klasse erweitert PHP PDO

`$db = new Database\(\);`

Definiert in Zeile 9 der Datei [Database.php](#).

### 3.4.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.4.2.1 Database::\_\_construct ( )

Konstruktor.

Konfiguration laden und der Elternklasse uebergeben.

Definiert in Zeile [22](#) der Datei [Database.php](#).

```

00022                                     {
00023                                     $conf = \Config::getInstance();
00024                                     $this->dsn = $conf->database_type . ":host=" . $conf->database_host . ";port=" . $conf->
00025                                     database_port . ";dbname=" . $conf->database_name;
00026                                     $this->user = $conf->database_user;
00027                                     $this->pass = $conf->database_pass;
00028                                     parent::__construct($this->dsn, $this->user, $this->pass);
00029                                     if ($conf->database_verbose == 1) $this->beMoreVerbose();
00030                                     }
00031
00032
00033
00034
00035
00036
00037

```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Database.php](#)

## 3.5 Controller\Database Klassenreferenz

### Öffentliche Methoden

- [\\_\\_construct \(\)](#)
- [Diagramm\\_Action \(\)](#)
- [CreateTable\\_Action \(\)](#)

### Private Attribute

- [\\$view](#)

### 3.5.1 Ausführliche Beschreibung

Datenbankinformationen Controller

Definiert in Zeile [11](#) der Datei [Controller/Database.php](#).

### 3.5.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.5.2.1 Controller\Database::\_\_construct ( )

Konstruktor

[View](#) Initialisieren

Definiert in Zeile [22](#) der Datei [Controller/Database.php](#).

```

00022                                     {
00023
00024         \Session::isUserAuthedCheck();
00025
00026         $this->view = new \View();
00027
00028     }

```

### 3.5.3 Dokumentation der Elementfunktionen

#### 3.5.3.1 Controller\Database::CreateTable\_Action ( )

SQL Statements anzeigen

Definiert in Zeile 47 der Datei [Controller/Database.php](#).

```

00047                                     {
00048
00049         $this->view->setTemplate('database_create_table');
00050         $this->view->display();
00051
00052     }

```

#### 3.5.3.2 Controller\Database::Diagramm\_Action ( )

UML Diagramm anzeigen

Definiert in Zeile 35 der Datei [Controller/Database.php](#).

```

00035                                     {
00036
00037         $this->view->setTemplate('database_diagramm');
00038         $this->view->display();
00039
00040     }

```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Controller/Database.php](#)

## 3.6 FrontController Klassenreferenz

### Öffentliche Methoden

- [\\_\\_construct](#) ()
- [run](#) ()

### Private Attribute

- **\$controller**
- **\$action**

#### 3.6.1 Ausführliche Beschreibung

[FrontController](#) Klasse

Get Request: ?controller=test&action=machwas

Ergebniss: Automatisches Einbinden der Controllerklasse "test" und Aufruf der Methode machwas\_Action()

Beim gleichen Request als Post wird die Methode machwas\_Action\_POST() aufgerufen

Definiert in Zeile 18 der Datei [FrontController.php](#).



### 3.6.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.6.2.1 FrontController::\_\_construct ( )

Konstruktor

`$c = new FrontCrontroller() $c->run();`

Definiert in Zeile 31 der Datei [FrontController.php](#).

```

00031         {
00032
00033             if (isset($_GET['controller'])) $controller = $_GET['controller'];
00034
00035             // Workaround magic_quotes() . Ab PHP 5.4 DEPRICATED
00036             if (get_magic_quotes_gpc()) {
00037                 $controller = str_replace("\\\\", "\\", $controller);
00038             }
00039
00040             if (isset($_GET['action'])) $action = $_GET['action'];
00041
00042
00043             $this->controller = !empty($controller) ? CONTROLLER_NAMESPACE . $controller :
CONTROLLER_NAMESPACE . "Index";
00044
00045             $this->action = !empty($action) ? $action : "Index";
00046             if ($_SERVER['REQUEST_METHOD'] == 'POST') {
00047                 $this->action .= "_POST";
00048             }
00049             $this->action .= "_Action";
00050
00051         }

```

### 3.6.3 Dokumentation der Elementfunktionen

#### 3.6.3.1 FrontController::run ( )

[FrontController](#) ausführen

Definiert in Zeile 58 der Datei [FrontController.php](#).

```

00058         {
00059
00060             $controller = new $this->controller();
00061
00062             if(in_array($this->action, get_class_methods($controller))) {
00063                 $controller->{$this->action}();
00064             } else {
00065                 throw new Exception("'" . $this->controller . "' hat keine Aktion '" . $this->action . "'");
00066             }
00067
00068
00069         }

```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [FrontController.php](#)

## 3.7 Controller/Index Klassenreferenz

### Öffentliche Methoden

- [Index\\_Action](#) ( )

#### 3.7.1 Ausführliche Beschreibung

[Index](#) Controller

Wird default immer Aufgerufen

Definiert in Zeile 11 der Datei [Index.php](#).

### 3.7.2 Dokumentation der Elementfunktionen

#### 3.7.2.1 Controller\Index::Index\_Action ( )

Default [Index](#) Get Action

Definiert in Zeile 18 der Datei [Index.php](#).

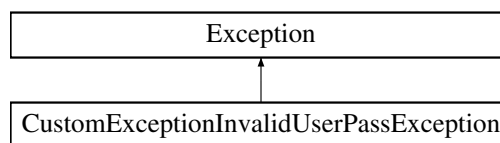
```
00018                                     {
00019
00020         $view = new \View();
00021         $view->setTemplate('index');
00022         $view->display();
00023
00024     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Index.php](#)

## 3.8 CustomException\InvalidUserPassException Klassenreferenz

Klassendiagramm für CustomException\InvalidUserPassException:



#### 3.8.1 Ausführliche Beschreibung

Diese Exception wird geworfen wenn der Benutzer nicht vorhanden oder das Passwort falsch ist.

Definiert in Zeile 10 der Datei [InvalidUserPassException.php](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [InvalidUserPassException.php](#)

## 3.9 Redirect Klassenreferenz

### Öffentliche, statische Methoden

- static [Header](#) (\$url)
- static [toController](#) (\$controller)

#### 3.9.1 Ausführliche Beschreibung

[Redirect](#) Klasse

Definiert in Zeile 7 der Datei [Redirect.php](#).

### 3.9.2 Dokumentation der Elementfunktionen

#### 3.9.2.1 static Redirect::Header ( \$url ) [static]

Header [Redirect](#) durchfuehren

Parameter

String	<i>\$url</i>	URL
--------	--------------	-----

Definiert in Zeile 16 der Datei [Redirect.php](#).

```
00016                                     {
00017
00018             header("Location: $url");
00019     }
```

#### 3.9.2.2 static Redirect::toController ( \$controller ) [static]

Controller [Redirect](#) durchfuehren

Parameter

String	<i>\$controller</i>	Controllername
--------	---------------------	----------------

Definiert in Zeile 28 der Datei [Redirect.php](#).

```
00028                                     {
00029
00030             header("Location: index.php?controller=$controller");
00031     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Redirect.php](#)

## 3.10 Session Klassenreferenz

### Öffentliche Methoden

- [\\_\\_get](#) (\$key)
- [\\_\\_set](#) (\$key, \$value)

### Öffentliche, statische Methoden

- static [getInstance](#) ()
- static [authUser](#) ()
- static [isUserAuthed](#) ()
- static [isUserAuthedCheck](#) ()
- static [destroy](#) ()

### Private Methoden

- [\\_\\_clone](#) ()

### Private Attribute

- **\$session** = array()

## Statische, private Attribute

- static **\$instance** = null

### 3.10.1 Ausführliche Beschreibung

[Session](#) Klasse als Singleton Pattern

```
$session = ::getInstance(); $session->key = "value";
```

```
::destroy();
```

Definiert in Zeile [17](#) der Datei [Session.php](#).

### 3.10.2 Dokumentation der Elementfunktionen

#### 3.10.2.1 Session::\_\_get ( \$key )

Getter Methode

Parameter

String	<i>\$key</i>	Schlüssel
--------	--------------	-----------

Rückgabe

String Wert

Definiert in Zeile [46](#) der Datei [Session.php](#).

```
00046                                     {
00047
00048             if (array_key_exists($key, $_SESSION)) {
00049                 return $_SESSION;
00050             } else {
00051                 throw new Exception("Session Schluessel '$key' nicht verfuegbar");
00052             }
00053
00054 }
```

#### 3.10.2.2 Session::\_\_set ( \$key, \$value )

Setter Methode

Parameter

String	<i>\$key</i>	Schlüssel
String	<i>\$value</i>	Wert

Definiert in Zeile [64](#) der Datei [Session.php](#).

```
00064                                     {
00065
00066             $_SESSION[$key] = $value;
00067
00068 }
```

#### 3.10.2.3 static Session::authUser ( ) [static]

Benutzer Authentifizieren

Definiert in Zeile [75](#) der Datei [Session.php](#).

```

00075                                     {
00076
00077             $_SESSION['__AUTH'] = 1;
00078
00079     }

```

#### 3.10.2.4 static Session::destroy ( ) [static]

Session löschen

Definiert in Zeile 114 der Datei [Session.php](#).

```

00114                                     {
00115
00116             session_destroy();
00117
00118     }

```

#### 3.10.2.5 static Session::isUserAuthenticated ( ) [static]

Überprüfen ob Benutzer Authentifiziert ist

Rückgabe

Boolean

Definiert in Zeile 88 der Datei [Session.php](#).

```

00088                                     {
00089
00090             if (array_key_exists('__AUTH', $_SESSION) && $_SESSION['__AUTH'] == 1) return true;
00091             return false;
00092
00093
00094     }

```

#### 3.10.2.6 static Session::isUserAuthenticatedCheck ( ) [static]

Überprüfen ob Benutzer Authentifiziert ist

Ansonsten Exception werfen

Definiert in Zeile 103 der Datei [Session.php](#).

```

00103                                     {
00104
00105             if (!self::isUserAuthenticated()) throw new Ex\UserNotAuthenticated("
00106             Sie haben keine Berechtigung für diese Seite");
00107
00108     }

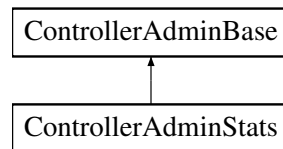
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Session.php

## 3.11 Controller\Admin\Stats Klassenreferenz

Klassendiagramm für Controller\Admin\Stats:



## Öffentliche Methoden

- [\\_\\_construct\(\)](#)
- [Index\\_Action\(\)](#)

## Weitere Geerbte Elemente

### 3.11.1 Ausführliche Beschreibung

Admin Statistik Controller

Definiert in Zeile [10](#) der Datei [Stats.php](#).

### 3.11.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.11.2.1 Controller\Admin\Stats::\_\_construct( )

Modell initialisieren

Definiert in Zeile [18](#) der Datei [Stats.php](#).

```
00018                                     {
00019
00020             $this->model = new \Model\Survey();
00021
00022             parent::__construct();
00023
00024     }
```

### 3.11.3 Dokumentation der Elementfunktionen

#### 3.11.3.1 Controller\Admin\Stats::Index\_Action( )

Default [Index](#) Get Action

Definiert in Zeile [31](#) der Datei [Stats.php](#).

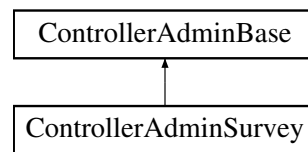
```
00031                                     {
00032
00033             $this->view->setTemplate('admin_stats');
00034             $this->view->assign('stats', $this->model->getStats());
00035             $this->view->display();
00036
00037     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Stats.php](#)

## 3.12 Controller\Admin\Survey Klassenreferenz

Klassendiagramm für Controller\Admin\Survey:



### Öffentliche Methoden

- [\\_\\_construct\(\)](#)
- [Index\\_Action\(\)](#)
- [Add\\_POST\\_Action\(\)](#)
- [Delete\\_Action\(\)](#)

### Weitere Geerbte Elemente

#### 3.12.1 Ausführliche Beschreibung

Admin Umfrage Controller

Definiert in Zeile 10 der Datei [Controller/Admin/Survey.php](#).

#### 3.12.2 Beschreibung der Konstruktoren und Destruktoren

##### 3.12.2.1 Controller\Admin\Survey::\_\_construct ( )

Modell initialisieren

Definiert in Zeile 17 der Datei [Controller/Admin/Survey.php](#).

```

00017                                     {
00018
00019             $this->model = new \Model\Survey();
00020
00021             parent::__construct();
00022
00023     }
  
```

#### 3.12.3 Dokumentation der Elementfunktionen

##### 3.12.3.1 Controller\Admin\Survey::Add\_POST\_Action ( )

neue Umfragen hinzufuegen

Definiert in Zeile 43 der Datei [Controller/Admin/Survey.php](#).

```

00043                                     {
00044
00045             if (isset($_POST['name']) && isset($_POST['answer'])) {
00046
00047                 $name = $_POST['name'];
00048                 $answer_arr = $_POST['answer'];
00049
00050                 $this->model->addSurvey($name, $answer_arr);
00051
00052                 $this->Index_Action();
00053
00054             }
00055     }
  
```

```

00054             } else {
00055
00056                 throw new \Exception("Illegaler Aufruf");
00057             }
00058         }
00059     }
00060 }

```

### 3.12.3.2 Controller\Admin\Survey::Delete\_Action ( )

Umfragen loeschen

Definiert in Zeile 67 der Datei [Controller/Admin/Survey.php](#).

```

00067             {
00068
00069                 if (isset($_GET['survey'])) {
00070
00071                     $id = intval($_GET['survey']);
00072
00073                     if ($id > 0) {
00074
00075                         $this->model->deleteSurvey($id);
00076
00077                         $this->Index_Action();
00078
00079                     } else {
00080
00081                         throw new \Exception("Illegaler Aufruf");
00082
00083                     }
00084
00085                 } else {
00086
00087                     throw new \Exception("Illegaler Aufruf");
00088
00089                 }
00090             }
00091         }
00092     }
00093 }

```

### 3.12.3.3 Controller\Admin\Survey::Index\_Action ( )

Default [Index](#) Get Action

Definiert in Zeile 30 der Datei [Controller/Admin/Survey.php](#).

```

00030             {
00031
00032                 $this->view->setTemplate('admin_surveys');
00033                 $this->view->assign('surveys', $this->model->getSurveys());
00034                 $this->view->display();
00035             }
00036         }

```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Controller/Admin/Survey.php](#)

## 3.13 Controller\Survey Klassenreferenz

### Öffentliche Methoden

- [\\_\\_construct](#) ()
- [Index\\_Action](#) ()
- [Save\\_POST\\_Action](#) ()
- [Show\\_Action](#) ()



## Private Methoden

- [showSurveyResult\(\)](#)

## Private Attribute

- `$view`
- `$model`
- `$survey`

### 3.13.1 Ausführliche Beschreibung

Umfrage Controller

Definiert in Zeile 10 der Datei [Controller/Survey.php](#).

### 3.13.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.13.2.1 Controller\Survey::\_\_construct()

Konstruktor

Definiert in Zeile 21 der Datei [Controller/Survey.php](#).

```
00021         {
00022
00023             $this->view = new \View();
00024             $this->model = new \Model\Survey();
00025             if (isset($_GET['survey'])) $this->survey = intval($_GET['survey']);
00026         }
00027     }
```

### 3.13.3 Dokumentation der Elementfunktionen

#### 3.13.3.1 Controller\Survey::index\_Action()

Default [Index](#) Get Action

Definiert in Zeile 34 der Datei [Controller/Survey.php](#).

```
00034         {
00035
00036
00037
00038             if ($this->survey == 0) {
00039
00040                 $this->view->setTemplate('surveys');
00041                 $this->view->assign('surveys', $this->model->getSurveys());
00042                 $this->view->display();
00043
00044             } else {
00045
00046                 $this->view->setTemplate('survey');
00047                 $this->view->assign('survey_id', $this->survey);
00048                 $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00049                 $this->view->assign('survey_items', $this->model->getSurveyItems($this->survey));
00050                 $this->view->display();
00051
00052             }
00053         }
00054     }
00055 }
```

### 3.13.3.2 Controller\Survey::Save\_POST\_Action ( )

Umfrage Ergebnisse speichern

Definiert in Zeile 64 der Datei [Controller/Survey.php](#).

```

00064                                     {
00065
00066             if ($this->survey > 0 && isset($_POST['answer'])) {
00067
00068                 $answerArr = $_POST['answer'];
00069
00070                 $this->model->saveNewAnswers($answerArr);
00071
00072                 $this->showSurveyResult();
00073
00074             } else {
00075
00076                 // Wird nichts ausgewaelt einfach das Ergebnis anzeigen.
00077                 $this->showSurveyResult();
00078
00079             }
00080
00081         }
00082     }

```

### 3.13.3.3 Controller\Survey::Show\_Action ( )

Umfrage Ergebnisse anzeigen

Direkt nur fuer Administratoren moeglich

Definiert in Zeile 91 der Datei [Controller/Survey.php](#).

```

00091                                     {
00092
00093             if ($this->survey == 0) throw new \Exception("Illegaler Aufruf");
00094
00095             \Session::isUserAuthedCheck();
00096
00097             $this->showSurveyResult();
00098
00099         }

```

### 3.13.3.4 Controller\Survey::showSurveyResult ( ) [private]

Auswertung anzeigen

Definiert in Zeile 106 der Datei [Controller/Survey.php](#).

```

00106                                     {
00107
00108             $this->view->setTemplate('survey_result');
00109             $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00110             $this->view->assign('survey_cnt', $this->model->getSurveyItemCount($this->survey));
00111             $this->view->assign('survey_result', $this->model->getSurveyResult($this->survey));
00112             $this->view->display();
00113
00114         }

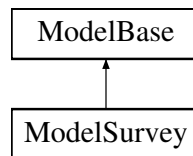
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Controller/Survey.php

## 3.14 Model\Survey Klassenreferenz

Klassendiagramm für Model\Survey:



## Öffentliche Methoden

- [getSurveys](#) ()
- [getSurveyName](#) (\$survey)
- [getSurveyItems](#) (\$survey)
- [getSurveyItemCount](#) (\$survey)
- [getSurveyResult](#) (\$survey)
- [getStats](#) ()
- [saveNewAnswers](#) (\$answerArr)
- [deleteSurvey](#) (\$id)
- [addSurvey](#) (\$name, \$answerArr)

## Private Methoden

- [getTableCount](#) (\$table)

## Weitere Geerbte Elemente

### 3.14.1 Ausführliche Beschreibung

Umfrage Datenmodell

Definiert in Zeile 9 der Datei [Model/Survey.php](#).

### 3.14.2 Dokumentation der Elementfunktionen

#### 3.14.2.1 Model\Survey::addSurvey ( \$name, \$answerArr )

neue Umfrage hinzufügen

Parameter

String	<i>\$name</i>	Name der Umfrage
Array	<i>\$answerArr</i>	Array mit den möglichen Antworten

Definiert in Zeile 167 der Datei [Model/Survey.php](#).

```

00167                                     {
00168
00169         if (empty($name)) return;
00170
00171         $stmt = $this->dbh->prepare("INSERT INTO Survey SET Name = :name");
00172         $stmt->bindParam(':name', $name);
00173
00174         $stmt->execute();
00175
00176         $surveyID = $this->dbh->lastInsertId();
00177
00178         $stmt = $this->dbh->prepare("INSERT INTO SurveyItems SET Name = :name, SurveyID = :id");
00179         $stmt->bindParam(':id', $surveyID);
00180
00181         foreach ($answerArr as $answer) {
00182             if (!empty($answer)) {
00183
00184

```

```

00185             $stmt->bindParam(':name', $answer);
00186             $stmt->execute();
00187         }
00188     }
00189 }
00190
00191 }

```

### 3.14.2.2 Model\Survey::deleteSurvey ( \$id )

Umfrage löschen

Parameter

Integer	\$id	UmfragenID
---------	------	------------

Definiert in Zeile 149 der Datei [Model/Survey.php](#).

```

00149         {
00150
00151             $stmt = $this->dbh->prepare("DELETE FROM Survey WHERE ID = :id");
00152             $stmt->bindParam(':id', $id);
00153
00154             $stmt->execute();
00155
00156         }

```

### 3.14.2.3 Model\Survey::getStats ( )

Statistiken holen

Definiert in Zeile 109 der Datei [Model/Survey.php](#).

```

00109         {
00110
00111             $stat = array();
00112
00113             $stat['user_cnt'] = $this->getTableCount("User");
00114             $stat['survey_cnt'] = $this->getTableCount("Survey");
00115             $stat['survey_items_cnt'] = $this->getTableCount("SurveyItems");
00116             $stat['answer_cnt'] = $this->getTableCount("SurveyAnswer");
00117
00118             return $stat;
00119         }

```

### 3.14.2.4 Model\Survey::getSurveyItemCount ( \$survey )

Anzahl der Abgegeben Umfragewerte

Parameter

Integer	\$survey	Umfrageld
---------	----------	-----------

Rückgabe

Integer Anzahl der Ergebnisse zur Umfrage

Definiert in Zeile 75 der Datei [Model/Survey.php](#).

```

00075         {
00076
00077             $stmt = $this->dbh->prepare("SELECT COUNT(*) AS cnt FROM SurveyAnswer WHERE SurveyItemID IN
00078 (SELECT ID FROM SurveyItems WHERE SurveyID = :id)");
00079             $stmt->bindParam(':id', $survey);
00080             $stmt->execute();
00081             $res = $stmt->fetchObject();
00082
00083             return $res->cnt;
00084         }

```

### 3.14.2.5 Model\Survey::getSurveyItems ( \$survey )

Umfrage Punkte holen

Parameter

Integer	<i>\$survey</i>	Umfrageld
---------	-----------------	-----------

Rückgabe

Array

Definiert in Zeile 54 der Datei [Model/Survey.php](#).

```
00054                                     {
00055
00056     $stmt = $this->dbh->prepare("SELECT ID, Name FROM SurveyItems WHERE SurveyID = :id ORDER BY
    Name");
00057     $stmt->bindParam(':id', $survey);
00058     $stmt->execute();
00059
00060     return $stmt->fetchAll();
00061
00062
00063
00064 }
```

### 3.14.2.6 Model\Survey::getSurveyName ( \$survey )

Umfragenamen holen

Parameter

Integer	<i>\$survey</i>	Umfrageld
---------	-----------------	-----------

Rückgabe

String Name der Umfrage

Definiert in Zeile 34 der Datei [Model/Survey.php](#).

```
00034                                     {
00035
00036     $stmt = $this->dbh->prepare("SELECT Name FROM Survey WHERE ID = :id");
00037     $stmt->bindParam(':id', $survey);
00038     $stmt->execute();
00039     $res = $stmt->fetchObject();
00040
00041     return $res->Name;
00042
00043 }
```

### 3.14.2.7 Model\Survey::getSurveyResult ( \$survey )

Umfrage Ergebnisse holen

Parameter

Integer	<i>\$survey</i>	Umfrageld
---------	-----------------	-----------

**Rückgabe**

Array

Definiert in Zeile 94 der Datei [Model/Survey.php](#).

```

00094                                     {
00095
00096             $stmt = $this->dbh->prepare("SELECT i.Name, COUNT(a.ID) as cnt FROM SurveyItems i LEFT JOIN
SurveyAnswer a ON i.ID = a. SurveyItemID WHERE i.SurveyID = :id GROUP BY Name ORDER BY i.Name");
00097             $stmt->bindParam(':id', $survey);
00098             $stmt->execute();
00099
00100             return $stmt->fetchAll();
00101
00102     }

```

**3.14.2.8 Model\Survey::getSurveys ( )**

Umfragen holen

**Rückgabe**

Array

Definiert in Zeile 18 der Datei [Model/Survey.php](#).

```

00018                                     {
00019
00020             $stmt = $this->dbh->query("SELECT * FROM Survey");
00021             return $stmt->fetchAll();
00022
00023     }

```

**3.14.2.9 Model\Survey::getTableCount( \$table ) [private]**

Anzahl der Tupel zurückgeben

**Parameter**

String	<i>\$table</i>	Tabellenname
--------	----------------	--------------

**Rückgabe**

Integer Anzahl der Tupel

Definiert in Zeile 202 der Datei [Model/Survey.php](#).

```

00202                                     {
00203
00204
00205             switch ($table) {
00206
00207                 case "User":
00208                 case "Survey":
00209                 case "SurveyItems":
00210                 case "SurveyAnswer":
00211                 $stmt = $this->dbh->query("SELECT COUNT(*) AS cnt FROM $table");
00212                 return $stmt->fetchObject()->cnt;
00213                 break;
00214             default:
00215                 throw new \Exception("Illegaler Tabellenname $table");
00216
00217             }
00218
00219     }

```

**3.14.2.10 Model\Survey::saveNewAnswers ( *\$answerArr* )**

Antworten Speichern

**Parameter**

Integer	<i>\$answerArr</i>	Array mit den Antwort IDs
---------	--------------------	---------------------------

Definiert in Zeile 128 der Datei [Model/Survey.php](#).

```

00128                                     {
00129
00130             $stmt = $this->dbh->prepare("INSERT INTO SurveyAnswer SET SurveyItemID = :id");
00131
00132             foreach ($answerArr as $answer) {
00133
00134                 $stmt->bindParam(':id', $answer);
00135                 $stmt->execute();
00136
00137             }
00138
00139     }

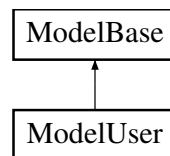
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Model/Survey.php](#)

### 3.15 Model\User Klassenreferenz

Klassendiagramm für Model\User:



#### Öffentliche Methoden

- [checkCredentials](#) (\$user, \$pass)
- [getUsers](#) ()
- [deleteUser](#) (\$id)
- [addUser](#) (\$name, \$pass)

#### Private Methoden

- [updateLastLogin](#) (\$user)

#### Weitere Geerbte Elemente

##### 3.15.1 Ausführliche Beschreibung

Benutzer Datenmodell

Definiert in Zeile 9 der Datei [Model/User.php](#).

##### 3.15.2 Dokumentation der Elementfunktionen

###### 3.15.2.1 Model\User::addUser ( \$name, \$pass )

Benutzer hinzufügen



**Parameter**

String	<i>\$name</i>	Benutzername
String	<i>\$pass</i>	Passwort

Definiert in Zeile 85 der Datei [Model/User.php](#).

```

00085                                     {
00086
00087         $stmt = $this->dbh->prepare("INSERT INTO User SET Name = :name, Passwort = :pass");
00088         $stmt->bindParam(':name', $name);
00089         $stmt->bindParam(':pass', $pass);
00090
00091         $stmt->execute();
00092
00093     }
```

**3.15.2.2 Model\User::checkCredentials ( \$user, \$pass )**

Berechtigung ueberprüfen

**Parameter**

String	<i>\$user</i>	Benutzername
String	<i>\$pass</i>	Passwort

**Rückgabe**

Boolean

Definiert in Zeile 22 der Datei [Model/User.php](#).

```

00022                                     {
00023
00024         if (empty($user) || empty($pass)) return false;
00025
00026         $stmt = $this->dbh->prepare("SELECT Passwort FROM User WHERE Name = :name");
00027         $stmt->bindParam(':name', $user);
00028
00029         $stmt->execute();
00030         $res = $stmt->fetchObject();
00031
00032         if (!is_object($res)) return false;
00033
00034         if ($res->Passwort == $pass) {
00035
00036             $this->updateLastLogIn($user);
00037             return true;
00038
00039         } else {
00040
00041             return false;
00042
00043         }
00044
00045     }
```

**3.15.2.3 Model\User::deleteUser ( \$id )**

Benutzer löschen

**Parameter**


---

Integer	<i>\$id</i>	BenutzerID
---------	-------------	------------

Definiert in Zeile 68 der Datei [Model/User.php](#).

```

00068                                     {
00069
00070             $stmt = $this->dbh->prepare("DELETE FROM User WHERE ID = :id");
00071             $stmt->bindParam(':id', $id);
00072
00073             $stmt->execute();
00074
00075         }

```

### 3.15.2.4 Model\User::getUsers ( )

Benutzer holen

Rückgabe

Array

Definiert in Zeile 54 der Datei [Model/User.php](#).

```

00054                                     {
00055
00056             $stmt = $this->dbh->query("SELECT ID, Name, LastLogIn FROM User");
00057             return $stmt->fetchAll();
00058
00059         }

```

### 3.15.2.5 Model\User::updateLastLogIn ( \$user ) [private]

Zeitstempel des letzten Logins vom [User](#) Updaten

Parameter

String	<i>\$user</i>	Benutzername
--------	---------------	--------------

Definiert in Zeile 103 der Datei [Model/User.php](#).

```

00103                                     {
00104
00105             $stmt = $this->dbh->prepare("UPDATE User SET LastLogIn = :lastlogin WHERE Name = :name");
00106             $stmt->bindParam(':name', $user);
00107             $stmt->bindValue(':lastlogin', date("Y-m-d H:i:s"));
00108
00109             $stmt->execute();
00110
00111         }

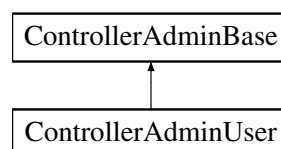
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Model/User.php](#)

## 3.16 Controller\Admin\User Klassenreferenz

Klassendiagramm für Controller\Admin\User:



## Öffentliche Methoden

- [\\_\\_construct\(\)](#)
- [Index\\_Action\(\)](#)
- [Add\\_POST\\_Action\(\)](#)
- [Delete\\_Action\(\)](#)

## Weitere Geerbte Elemente

### 3.16.1 Ausführliche Beschreibung

Admin Benutzer Controller

Definiert in Zeile 10 der Datei [Controller/Admin/User.php](#).

### 3.16.2 Beschreibung der Konstruktoren und Destruktoren

#### 3.16.2.1 Controller\Admin\User::\_\_construct()

Modell initialisieren

Definiert in Zeile 18 der Datei [Controller/Admin/User.php](#).

```
00018                                     {
00019
00020             $this->model = new \Model\User();
00021
00022             parent::__construct();
00023
00024 }
```

### 3.16.3 Dokumentation der Elementfunktionen

#### 3.16.3.1 Controller\Admin\User::Add\_POST\_Action()

Benutzer hinzufuegen

Definiert in Zeile 45 der Datei [Controller/Admin/User.php](#).

```
00045                                     {
00046
00047             if (isset($_POST['name']) && isset($_POST['pass'])) {
00048
00049                 $name = $_POST['name'];
00050                 $pass = $_POST['pass'];
00051
00052                 $this->model->addUser($name, $pass);
00053
00054                 $this->Index_Action();
00055
00056             } else {
00057
00058                 throw new \Exception("Illegaler Aufruf");
00059
00060             }
00061
00062
00063 }
```

#### 3.16.3.2 Controller\Admin\User::Delete\_Action()

Benutzer loeschen

Definiert in Zeile 70 der Datei [Controller/Admin/User.php](#).

```

00070             {
00071
00072
00073         if (isset($_GET['user'])) {
00074
00075             $id = intval($_GET['user']);
00076
00077             if ($id > 0) {
00078
00079                 $this->model->deleteUser($id);
00080
00081                 $this->Index_Action();
00082
00083             } else {
00084
00085                 throw new \Exception("Illegaler Aufruf");
00086
00087             }
00088
00089         } else {
00090
00091             throw new \Exception("Illegaler Aufruf");
00092
00093         }
00094     }
00095 }

```

### 3.16.3.3 Controller\Admin\User::Index\_Action ( )

Default [Index](#) Get Action

Definiert in Zeile [31](#) der Datei [Controller/Admin/User.php](#).

```

00031             {
00032
00033         $this->view->setTemplate('admin_user');
00034         $this->view->assign('users', $this->model->getUsers());
00035
00036         $this->view->display();
00037     }
00038 }

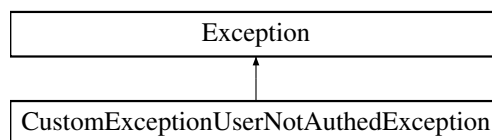
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Controller/Admin/User.php](#)

## 3.17 CustomException\UserNotAuthenticatedException Klassenreferenz

Klassendiagramm für CustomException\UserNotAuthenticatedException:



### 3.17.1 Ausführliche Beschreibung

Diese Exception wird geworfen wenn der Benutzer nicht eingeloggt ist.

Definiert in Zeile [10](#) der Datei [UserNotAuthenticatedException.php](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [UserNotAuthenticatedException.php](#)

## 3.18 Controller\UserSession Klassenreferenz

### Öffentliche Methoden

- [Login\\_POST\\_Action](#) ()
- [Logout\\_Action](#) ()

### 3.18.1 Ausführliche Beschreibung

Login Controller

Definiert in Zeile 12 der Datei [UserSession.php](#).

### 3.18.2 Dokumentation der Elementfunktionen

#### 3.18.2.1 Controller\UserSession::Login\_POST\_Action ( )

Verarbeitung der Logindaten

Definiert in Zeile 20 der Datei [UserSession.php](#).

```

00020                                     {
00021
00022         $user = $_POST['user'];
00023         $pass = $_POST['pass'];
00024
00025         $model = new \Model\User();
00026
00027         if ($model->checkCredentials($user, $pass)) {
00028
00029             \Session::authUser();
00030             \Redirect::toController("Index");
00031
00032         } else {
00033
00034             throw new Ex\InvalidUserPassException();
00035
00036         }
00037
00038
00039     }
```

#### 3.18.2.2 Controller\UserSession::Logout\_Action ( )

Logout Action

Definiert in Zeile 46 der Datei [UserSession.php](#).

```

00046                                     {
00047
00048         \Session::destroy();
00049         \Redirect::toController("Index");
00050
00051     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [UserSession.php](#)

## 3.19 View Klassenreferenz

### Öffentliche Methoden

- [assign](#) (\$key, \$value)

- [setTemplate](#) (\$template)
- [display](#) ()

### Private Methoden

- [getTemplateOnFilesystem](#) (\$template)

### Private Attribute

- **\$templatePath** = 'templates'
- **\$template** = 'index'
- **\$layouttemplate** = '\_layout'
- **\$includetemplate**
- **\$view** = array()

## 3.19.1 Ausführliche Beschreibung

[View](#) Klasse

Definiert in Zeile 7 der Datei [View.php](#).

## 3.19.2 Dokumentation der Elementfunktionen

### 3.19.2.1 View::assign( \$key, \$value )

Template Variable zuweisen

Parameter

String	<i>\$key</i>	Schlüssel
String	<i>\$value</i>	Wert

Definiert in Zeile 29 der Datei [View.php](#).

```
00029
00030
00031
    {
        $this->view[$key] = $value;
    }
```

### 3.19.2.2 View::display ( )

Template anzeigen

Definiert in Zeile 49 der Datei [View.php](#).

```
00049
00050
00051
    {
        $this->includetemplate = $this->getTemplateOnFilesystem($this->
template);
00052
00053
00054
        // Überprüfen ob das angeforderte Template Existiert
        if (file_exists($this->includetemplate)) {
00055
00056
            // Layouttemplate einbinden
            include $this->getTemplateOnFilesystem($this->layouttemplate
);
00057
00058
        } else {
00059
00060
            throw new Exception("Template '{$this->template}' nicht gefunden");
00061
00062
        }
00063
00064
    }
```

### 3.19.2.3 View::getTemplateOnFilesystem ( *\$template* ) [private]

Anhand des Templatenamen den vollen Pfad inkl Datei zurueckgeben

**Parameter**

String	<i>\$template</i>	Templatename
--------	-------------------	--------------

**Rückgabe**

String Pfad inkl Dateiname zum Template

Definiert in Zeile [75](#) der Datei [View.php](#).

```
00075                                     {
00076
00077         return $this->templatePath . DIRECTORY_SEPARATOR . $template . ".tpl.php";
00078
00079     }
```

**3.19.2.4 View::setTemplate ( *\$template* )**

Template setzen

**Parameter**

String	<i>\$template</i>	Templatename
--------	-------------------	--------------

Definiert in Zeile [40](#) der Datei [View.php](#).

```
00040                                     {
00041         $this->template = $template;
00042     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- View.php



# Index

- \_\_construct
  - Config, [7](#)
  - Controller::Admin::Base, [5](#)
  - Controller::Admin::Stats, [16](#)
  - Controller::Admin::Survey, [17](#)
  - Controller::Admin::User, [29](#)
  - Controller::Database, [9](#)
  - Controller::Survey, [19](#)
  - Database, [9](#)
  - FrontController, [11](#)
  - Model::Base, [6](#)
- \_\_get
  - Config, [7](#)
  - Session, [14](#)
- \_\_set
  - Session, [14](#)
- Add\_POST\_Action
  - Controller::Admin::Survey, [17](#)
  - Controller::Admin::User, [29](#)
- addSurvey
  - Model::Survey, [21](#)
- addUser
  - Model::User, [26](#)
- assign
  - View, [32](#)
- authUser
  - Session, [14](#)
- checkCredentials
  - Model::User, [27](#)
- Config, [6](#)
  - \_\_construct, [7](#)
  - \_\_get, [7](#)
- Controller::Admin::Base
  - \_\_construct, [5](#)
- Controller::Admin::Stats
  - \_\_construct, [16](#)
  - Index\_Action, [16](#)
- Controller::Admin::Survey
  - \_\_construct, [17](#)
  - Add\_POST\_Action, [17](#)
  - Delete\_Action, [18](#)
  - Index\_Action, [18](#)
- Controller::Admin::User
  - \_\_construct, [29](#)
  - Add\_POST\_Action, [29](#)
  - Delete\_Action, [29](#)
  - Index\_Action, [30](#)
- Controller::Database
  - \_\_construct, [9](#)
  - CreateTable\_Action, [10](#)
  - Diagramm\_Action, [10](#)
- Controller::Index
  - Index\_Action, [12](#)
- Controller::Survey
  - \_\_construct, [19](#)
  - Index\_Action, [19](#)
  - Save\_POST\_Action, [19](#)
  - Show\_Action, [20](#)
  - showSurveyResult, [20](#)
- Controller::UserSession
  - Login\_POST\_Action, [31](#)
  - Logout\_Action, [31](#)
- Controller\Admin\Base, [5](#)
- Controller\Admin\Stats, [15](#)
- Controller\Admin\Survey, [17](#)
- Controller\Admin\User, [28](#)
- Controller\Database, [9](#)
- Controller\Index, [11](#)
- Controller\Survey, [18](#)
- Controller\UserSession, [31](#)
- CreateTable\_Action
  - Controller::Database, [10](#)
- CustomException\InvalidUserPassException, [12](#)
- CustomException\UserNotAuthenticatedException, [30](#)
- Database, [8](#)
  - \_\_construct, [9](#)
- Delete\_Action
  - Controller::Admin::Survey, [18](#)
  - Controller::Admin::User, [29](#)
- deleteSurvey
  - Model::Survey, [22](#)
- deleteUser
  - Model::User, [27](#)
- destroy
  - Session, [15](#)
- Diagramm\_Action
  - Controller::Database, [10](#)
- display
  - View, [32](#)
- FrontController, [10](#)
  - \_\_construct, [11](#)
  - run, [11](#)
- getStats
  - Model::Survey, [22](#)
- getSurveyItemCount

- Model::Survey, 22
- getSurveyItems
  - Model::Survey, 22
- getSurveyName
  - Model::Survey, 23
- getSurveyResult
  - Model::Survey, 23
- getSurveys
  - Model::Survey, 24
- getTableCount
  - Model::Survey, 24
- getTemplateOnFilesystem
  - View, 32
- getUsers
  - Model::User, 28
- Header
  - Redirect, 13
- Index\_Action
  - Controller::Admin::Stats, 16
  - Controller::Admin::Survey, 18
  - Controller::Admin::User, 30
  - Controller::Index, 12
  - Controller::Survey, 19
- isUserAuthenticated
  - Session, 15
- isUserAuthenticatedCheck
  - Session, 15
- Login\_POST\_Action
  - Controller::UserSession, 31
- Logout\_Action
  - Controller::UserSession, 31
- Model::Base
  - \_\_construct, 6
- Model::Survey
  - addSurvey, 21
  - deleteSurvey, 22
  - getStats, 22
  - getSurveyItemCount, 22
  - getSurveyItems, 22
  - getSurveyName, 23
  - getSurveyResult, 23
  - getSurveys, 24
  - getTableCount, 24
  - saveNewAnswers, 24
- Model::User
  - addUser, 26
  - checkCredentials, 27
  - deleteUser, 27
  - getUsers, 28
  - updateLastLogIn, 28
- Model\Base, 6
- Model\Survey, 20
- Model\User, 26
- Redirect, 12
- Header, 13
  - toController, 13
- run
  - FrontController, 11
- Save\_POST\_Action
  - Controller::Survey, 19
- saveNewAnswers
  - Model::Survey, 24
- Session, 13
  - \_\_get, 14
  - \_\_set, 14
  - authUser, 14
  - destroy, 15
  - isUserAuthenticated, 15
  - isUserAuthenticatedCheck, 15
- setTemplate
  - View, 34
- Show\_Action
  - Controller::Survey, 20
- showSurveyResult
  - Controller::Survey, 20
- toController
  - Redirect, 13
- updateLastLogIn
  - Model::User, 28
- View, 31
  - assign, 32
  - display, 32
  - getTemplateOnFilesystem, 32
  - setTemplate, 34