

DBA02

Erzeugt von Doxygen 1.8.5

Don Aug 29 2013 08:47:03

Inhaltsverzeichnis

1	Hierarchie-Verzeichnis	1
1.1	Klassenhierarchie	1
2	Klassen-Verzeichnis	3
2.1	Auflistung der Klassen	3
3	Klassen-Dokumentation	5
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	FrontController 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	run	10
3.6	Controller/Index Klassenreferenz	10

3.6.1	Ausführliche Beschreibung	10
3.6.2	Dokumentation der Elementfunktionen	11
3.6.2.1	Index_Action	11
3.7	CustomException\InvalidUserPassException Klassenreferenz	11
3.7.1	Ausführliche Beschreibung	11
3.8	Redirect Klassenreferenz	11
3.8.1	Ausführliche Beschreibung	11
3.8.2	Dokumentation der Elementfunktionen	12
3.8.2.1	Header	12
3.8.2.2	toController	13
3.9	Session Klassenreferenz	13
3.9.1	Ausführliche Beschreibung	14
3.9.2	Dokumentation der Elementfunktionen	14
3.9.2.1	__get	14
3.9.2.2	__set	14
3.9.2.3	authUser	14
3.9.2.4	destroy	15
3.9.2.5	isUserAuthed	15
3.9.2.6	isUserAuthedCheck	15
3.10	Controller\Admin\Stats Klassenreferenz	15
3.10.1	Ausführliche Beschreibung	16
3.10.2	Beschreibung der Konstruktoren und Destruktoren	16
3.10.2.1	__construct	16
3.10.3	Dokumentation der Elementfunktionen	16
3.10.3.1	Index_Action	16
3.11	Model\Survey Klassenreferenz	16
3.11.1	Ausführliche Beschreibung	17
3.11.2	Dokumentation der Elementfunktionen	17
3.11.2.1	addSurvey	17
3.11.2.2	deleteSurvey	17
3.11.2.3	getStats	18
3.11.2.4	getSurveyItemCount	18
3.11.2.5	getSurveyItems	18
3.11.2.6	getSurveyName	19
3.11.2.7	getSurveyResult	19
3.11.2.8	getSurveys	20
3.11.2.9	getTableCount	20
3.11.2.10	saveNewAnswers	20
3.12	Controller\Admin\Survey Klassenreferenz	21
3.12.1	Ausführliche Beschreibung	21

3.12.2	Beschreibung der Konstruktoren und Destruktoren	21
3.12.2.1	__construct	21
3.12.3	Dokumentation der Elementfunktionen	22
3.12.3.1	Add_POST_Action	22
3.12.3.2	Delete_Action	22
3.12.3.3	Index_Action	22
3.13	Controller\Survey Klassenreferenz	23
3.13.1	Ausführliche Beschreibung	23
3.13.2	Beschreibung der Konstruktoren und Destruktoren	23
3.13.2.1	__construct	23
3.13.3	Dokumentation der Elementfunktionen	24
3.13.3.1	Index_Action	24
3.13.3.2	Save_POST_Action	24
3.13.3.3	Show_Action	24
3.13.3.4	showSurveyResult	25
3.14	Controller\Admin\User Klassenreferenz	25
3.14.1	Ausführliche Beschreibung	25
3.14.2	Beschreibung der Konstruktoren und Destruktoren	25
3.14.2.1	__construct	25
3.14.3	Dokumentation der Elementfunktionen	26
3.14.3.1	Add_POST_Action	26
3.14.3.2	Delete_Action	26
3.14.3.3	Index_Action	26
3.15	Model\User Klassenreferenz	27
3.15.1	Ausführliche Beschreibung	27
3.15.2	Dokumentation der Elementfunktionen	27
3.15.2.1	addUser	27
3.15.2.2	checkCredentials	28
3.15.2.3	deleteUser	29
3.15.2.4	getUsers	29
3.15.2.5	updateLastLogin	30
3.16	CustomException\UserNotAuthedException Klassenreferenz	31
3.16.1	Ausführliche Beschreibung	31
3.17	Controller\UserSession Klassenreferenz	31
3.17.1	Ausführliche Beschreibung	31
3.17.2	Dokumentation der Elementfunktionen	32
3.17.2.1	Login_POST_Action	32
3.17.2.2	Logout_Action	32
3.18	View Klassenreferenz	32
3.18.1	Ausführliche Beschreibung	33

3.18.2	Dokumentation der Elementfunktionen	33
3.18.2.1	assign	33
3.18.2.2	display	33
3.18.2.3	getTemplateOnFilesystem	33
3.18.2.4	setTemplate	34
 Index		 35

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	21
Controller\Admin\User	25
Model\Base	6
Model\Survey	16
Model\User	27
Config	6
Exception	
CustomException\InvalidUserPassException	11
CustomException\UserNotAuthedException	31
FrontController	9
Controller/Index	10
PDO	
Database	8
Redirect	11
Session	13
Controller\Survey	23
Controller\UserSession	31
View	32

Kapitel 2

Klassen-Verzeichnis

2.1 Auflistung der Klassen

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

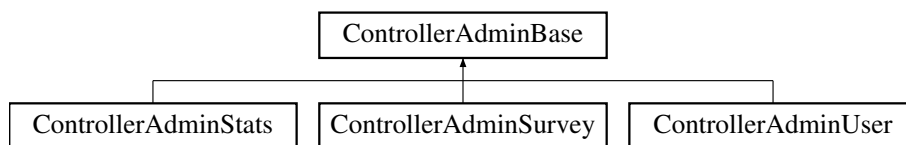
Controller\Admin\Base	5
Model\Base	6
Config	6
Database	8
FrontController	9
Controller/Index	10
CustomException\InvalidUserPassException	11
Redirect	11
Session	13
Controller\Admin\Stats	15
Model\Survey	16
Controller\Admin\Survey	21
Controller\Survey	23
Controller\Admin\User	25
Model\User	27
CustomException\UserNotAuthedException	31
Controller\UserSession	31
View	32

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 Konstrukt generell ueberpruefen ob sich der [User](#) eingeloggt hat.

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

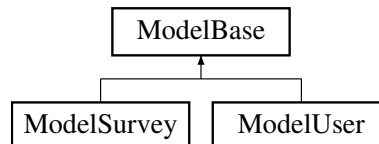
```
00022         {
00023
00024             \Session::isUserAuthedCheck();
00025
00026             $this->view = new \View();
00027
00028         }
```

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	\$key	Schlüssel
--------	--------------	-----------

Rückgabe

String Wert

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

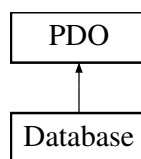
```
00051         {
00052
00053             if (array_key_exists($key, $this->config)) {
00054                 return $this->config[$key];
00055             } else {
00056                 throw new Exception("Config Schluessel '$key' nicht verfuegbar");
00057             }
00058         }
00059     }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

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

```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Database.php](#)

3.5 FrontController Klassenreferenz

Öffentliche Methoden

- [__construct](#) ()
- [run](#) ()

Private Attribute

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

3.5.1 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](#).

3.5.2 Beschreibung der Konstruktoren und Destruktoren

3.5.2.1 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         }

```

3.5.3 Dokumentation der Elementfunktionen

3.5.3.1 FrontController::run ()

[FrontController](#) ausuehren

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](#)

3.6 Controller/Index Klassenreferenz

Öffentliche Methoden

- [Index_Action](#) ()

3.6.1 Ausführliche Beschreibung

[Index](#) Controller

Wird default immer Aufgerufen

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

3.6.2 Dokumentation der Elementfunktionen

3.6.2.1 Controller\Index::Index_Action ()

Default [Index](#) Get Action

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

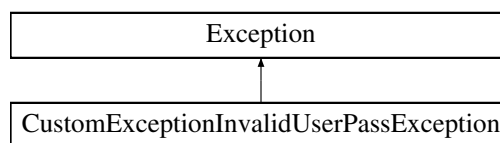
```
00018                                     {
00019
00020             $view = new \View();
00021             $view->setTemplate('index');
00022             $view->display();
00023
00024     }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Index.php](#)

3.7 CustomException\InvalidUserPassException Klassenreferenz

Klassendiagramm für CustomException\InvalidUserPassException:



3.7.1 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](#)

3.8 Redirect Klassenreferenz

Öffentliche, statische Methoden

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

3.8.1 Ausführliche Beschreibung

[Redirect](#) Klasse

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

3.8.2 Dokumentation der Elementfunktionen

3.8.2.1 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     }
```

3.8.2.2 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](#)

3.9 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

3.9.1 Ausführliche Beschreibung

[Session](#) Klasse als Singleton Pattern

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

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

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

3.9.2 Dokumentation der Elementfunktionen

3.9.2.1 Session::__get (\$key)

Getter Methode

Parameter

String	\$key	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 }
```

3.9.2.2 Session::__set (\$key, \$value)

Setter Methode

Parameter

String	\$key	Schlüssel
String	\$value	Wert

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

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

3.9.2.3 static Session::authUser () [static]

Benutzer Authentifizieren

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

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

3.9.2.4 static Session::destroy () [static]

[Session](#) löschen

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

```

00114                                     {
00115
00116             session_destroy();
00117
00118     }

```

3.9.2.5 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
00092             return false;
00093
00094     }

```

3.9.2.6 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\UserNotAuthenticatedException("
00106             Sie haben keine Berechtigung für diese Seite");
00107
00108     }

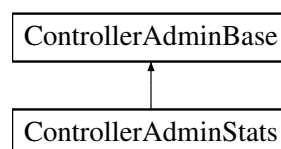
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Session.php](#)

3.10 Controller\Admin\Stats Klassenreferenz

Klassendiagramm für Controller\Admin\Stats:



Öffentliche Methoden

- [__construct\(\)](#)
- [Index_Action\(\)](#)

Weitere Geerbte Elemente

3.10.1 Ausführliche Beschreibung

Admin Statistik Controller

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

3.10.2 Beschreibung der Konstruktoren und Destruktoren

3.10.2.1 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         }
```

3.10.3 Dokumentation der Elementfunktionen

3.10.3.1 Controller\Admin\Stats::Index_Action ()

Default [Index](#) Get Action

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

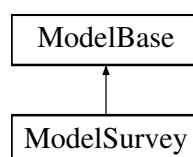
```
00031             {
00032
00033             $this->view->setTemplate('admin_stats');
00034             $this->view->assign('stats', $this->model->getStats());
00035             $this->view->display();
00036
00037         }
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Stats.php](#)

3.11 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

3.11.1 Ausführliche Beschreibung

Umfrage Datenmodell

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

3.11.2 Dokumentation der Elementfunktionen

3.11.2.1 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
00170             $stmt = $this->dbh->prepare("INSERT INTO Survey SET Name = :name");
00171             $stmt->bindParam(':name', $name);
00172
00173             $stmt->execute();
00174
00175             $surveyID = $this->dbh->lastInsertId();
00176
00177             $stmt = $this->dbh->prepare("INSERT INTO SurveyItems SET Name = :name, SurveyID = :id");
00178
00179             $stmt->bindParam(':id', $surveyID);
00180
00181             foreach ($answerArr as $answer) {
00182
00183                 $stmt->bindParam(':name', $answer);
00184                 $stmt->execute();
00185
00186             }
00187
00188         }

```

3.11.2.2 Model\Survey::deleteSurvey (\$id)

Umfrage löschen

Parameter

Integer	<i>\$id</i>	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     }
```

3.11.2.3 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     }
```

3.11.2.4 Model\Survey::getSurveyItemCount (\$survey)

Anzahl der Abgegeben Umfragewerte

Parameter

Integer	<i>\$survey</i>	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)");
00078         $stmt->bindParam(':id', $survey);
00079         $stmt->execute();
00080         $res = $stmt->fetchObject();
00081
00082         return $res->cnt;
00083
00084     }
```

3.11.2.5 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     }
```

3.11.2.6 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     }
```

3.11.2.7 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     }

```

3.11.2.8 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     }

```

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

Anzahl der Tupel zurückgeben

Parameter

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

Rückgabe

Integer Anzahl der Tupel

Definiert in Zeile 199 der Datei [Model/Survey.php](#).

```

00199                                     {
00200
00201
00202             switch ($table) {
00203
00204                 case "User":
00205                 case "Survey":
00206                 case "SurveyItems":
00207                 case "SurveyAnswer":
00208                 $stmt = $this->dbh->query("SELECT COUNT(*) AS cnt FROM $table");
00209                 return $stmt->fetchObject()->cnt;
00210                 break;
00211             default:
00212                 throw new \Exception("Illegalen Tabellenname $table");
00213
00214             }
00215
00216     }

```

3.11.2.10 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](#).

```

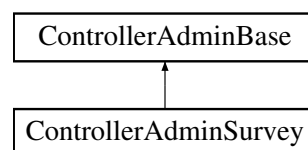
00128                                     {
00129
00130             $stmt = $this->dbh->prepare("INSERT INTO SurveyAnswer SET SurveyItemID = :id");
00131
00132             foreach ($answerArr as $answer) {
00133
00134                 $stmt->bindParam(':id', $answer);
00135                 $stmt->execute();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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Model/Survey.php](#)

3.12 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                 } 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 && $_GET['action'] == 'Save') {
00067
00068                 $answerArr = $_POST['answer'];
00069
00070                 $this->model->saveNewAnswers($answerArr);
00071
00072                 $this->showSurveyResult();
00073
00074             } else {
00075
00076                 //Exception werfen
00077
00078             }
00079
00080
00081     }
```

3.13.3.3 Controller\Survey::Show_Action ()

Umfrage Ergebnisse anzeigen

Direkt nur fuer Administratoren moeglich

Definiert in Zeile [90](#) der Datei [Controller/Survey.php](#).

```

00090                                     {
00091
00092             if ($this->survey == 0) throw new \Exception("Illegaler Aufruf");
00093
00094             \Session::isUserAuthedCheck();
00095
00096             $this->showSurveyResult();
00097
00098     }
```

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

Auswertung anzeigen

Definiert in Zeile 105 der Datei [Controller/Survey.php](#).

```

00105                                     {
00106
00107         $this->view->setTemplate('survey_result');
00108         $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00109         $this->view->assign('survey_cnt', $this->model->getSurveyItemCount($this->survey));
00110         $this->view->assign('survey_result', $this->model->getSurveyResult($this->survey));
00111         $this->view->display();
00112
00113     }

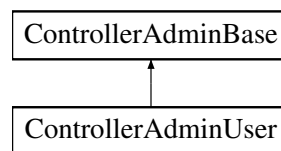
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Controller/Survey.php

3.14 Controller\Admin\User Klassenreferenz

Klassendiagramm für Controller\Admin\User:



Öffentliche Methoden

- [__construct \(\)](#)
- [Index_Action \(\)](#)
- [Add_POST_Action \(\)](#)
- [Delete_Action \(\)](#)

Weitere Geerbte Elemente

3.14.1 Ausführliche Beschreibung

Admin Benutzer Controller

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

3.14.2 Beschreibung der Konstruktoren und Destruktoren

3.14.2.1 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     }

```

3.14.3 Dokumentation der Elementfunktionen

3.14.3.1 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("Illegalen Aufruf");
00059
00060             }
00061
00062         }
00063     }

```

3.14.3.2 Controller\Admin\User::Delete_Action ()

Benutzer loeschen

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

```

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

```

3.14.3.3 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             }

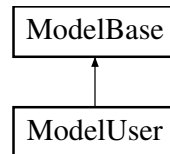
```


Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Controller/Admin/User.php

3.15 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 83 der Datei [Model/User.php](#).

```
00083                                     {
00084
00085         $stmt = $this->dbh->prepare("INSERT INTO User SET Name = :name, Passwort = :pass");
00086         $stmt->bindParam(':name', $name);
00087         $stmt->bindParam(':pass', $pass);
00088
00089         $stmt->execute();
00090
00091     }
```

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 = $stmt->fetchObject();
00031
00032             if ($res->Passwort == $pass) {
00033
00034                 $this->updateLastLogIn($user);
00035                 return true;
00036             } else {
00037
00038                 return false;
00039             }
00040         }
00041     }
00042 }
00043

```

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

Benutzer löschen

Parameter

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

Definiert in Zeile 66 der Datei [Model/User.php](#).

```

00066                                     {
00067
00068             $stmt = $this->dbh->prepare("DELETE FROM User WHERE ID = :id");
00069             $stmt->bindParam(':id', $id);
00070
00071             $stmt->execute();
00072
00073         }

```

3.15.2.4 Model\User::getUsers ()

Benutzer holen

Rückgabe

Array

Definiert in Zeile 52 der Datei [Model/User.php](#).

```

00052                                     {
00053
00054             $stmt = $this->dbh->query("SELECT ID, Name, LastLogIn FROM User");
00055             return $stmt->fetchAll();
00056
00057         }

```

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 101 der Datei [Model/User.php](#).

```

00101             {
00102
00103             $stmt = $this->dbh->prepare("UPDATE User SET LastLogIn = :lastlogin WHERE Name = :name");
00104             $stmt->bindParam(':name', $user);
00105             $stmt->bindValue(':lastlogin', date("Y-m-d H:i:s"));
00106
00107             $stmt->execute();
00108
00109         }

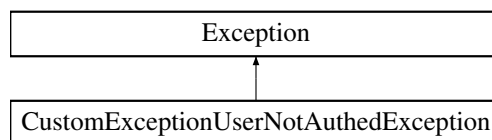
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Model/User.php](#)

3.16 CustomException\UserNotAuthedException Klassenreferenz

Klassendiagramm für CustomException\UserNotAuthedException:



3.16.1 Ausführliche Beschreibung

Diese Exception wird geworfen wenn der Benutzer nicht eingeloggt ist.

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [UserNotAuthedException.php](#)

3.17 Controller\UserSession Klassenreferenz

Öffentliche Methoden

- [Login_POST_Action](#) ()
- [Logout_Action](#) ()

3.17.1 Ausführliche Beschreibung

Login Controller

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

3.17.2 Dokumentation der Elementfunktionen

3.17.2.1 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     }
```

3.17.2.2 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

3.18 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()`

3.18.1 Ausführliche Beschreibung

[View](#) Klasse

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

3.18.2 Dokumentation der Elementfunktionen

3.18.2.1 `View::assign ($key, $value)`

Template Variable zuweisen

Parameter

String	<code>\$key</code>	Schluessel
String	<code>\$value</code>	Wert

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

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

3.18.2.2 `View::display ()`

Template anzeigen

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

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

3.18.2.3 `View::getTemplateOnFilesystem ($template)` [private]

Anhand des Templatennamen 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     }
```

3.18.2.4 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, [21](#)
 - Controller::Admin::User, [25](#)
 - Controller::Survey, [23](#)
 - Database, [9](#)
 - FrontController, [9](#)
 - Model::Base, [6](#)
- __get
 - Config, [7](#)
 - Session, [14](#)
- __set
 - Session, [14](#)
- Add_POST_Action
 - Controller::Admin::Survey, [22](#)
 - Controller::Admin::User, [26](#)
- addSurvey
 - Model::Survey, [17](#)
- addUser
 - Model::User, [27](#)
- assign
 - View, [33](#)
- 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, [21](#)
 - Add_POST_Action, [22](#)
 - Delete_Action, [22](#)
 - Index_Action, [22](#)
- Controller::Admin::User
 - __construct, [25](#)
 - Add_POST_Action, [26](#)
 - Delete_Action, [26](#)
 - Index_Action, [26](#)
- Controller::Index
 - Index_Action, [11](#)
- Controller::Survey
 - __construct, [23](#)
 - Index_Action, [24](#)
 - Save_POST_Action, [24](#)
 - Show_Action, [24](#)
 - showSurveyResult, [24](#)
- Controller::UserSession
 - Login_POST_Action, [32](#)
 - Logout_Action, [32](#)
- Controller\Admin\Base, [5](#)
- Controller\Admin\Stats, [15](#)
- Controller\Admin\Survey, [21](#)
- Controller\Admin\User, [25](#)
- Controller\Index, [10](#)
- Controller\Survey, [23](#)
- Controller\UserSession, [31](#)
- CustomException\InvalidUserPassException, [11](#)
- CustomException\UserNotAuthedException, [31](#)
- Database, [8](#)
 - __construct, [9](#)
- Delete_Action
 - Controller::Admin::Survey, [22](#)
 - Controller::Admin::User, [26](#)
- deleteSurvey
 - Model::Survey, [17](#)
- deleteUser
 - Model::User, [29](#)
- destroy
 - Session, [14](#)
- display
 - View, [33](#)
- FrontController, [9](#)
 - __construct, [9](#)
 - run, [10](#)
- getStats
 - Model::Survey, [18](#)
- getSurveyItemCount
 - Model::Survey, [18](#)
- getSurveyItems
 - Model::Survey, [18](#)
- getSurveyName
 - Model::Survey, [19](#)
- getSurveyResult
 - Model::Survey, [19](#)
- getSurveys
 - Model::Survey, [20](#)
- getTableCount

- Model::Survey, 20
- getTemplateOnFilesystem
 - View, 33
- getUsers
 - Model::User, 29
- Header
 - Redirect, 12
- Index_Action
 - Controller::Admin::Stats, 16
 - Controller::Admin::Survey, 22
 - Controller::Admin::User, 26
 - Controller::Index, 11
 - Controller::Survey, 24
- isUserAuthenticated
 - Session, 15
- isUserAuthenticatedCheck
 - Session, 15
- Login_POST_Action
 - Controller::UserSession, 32
- Logout_Action
 - Controller::UserSession, 32
- Model::Base
 - __construct, 6
- Model::Survey
 - addSurvey, 17
 - deleteSurvey, 17
 - getStats, 18
 - getSurveyItemCount, 18
 - getSurveyItems, 18
 - getSurveyName, 19
 - getSurveyResult, 19
 - getSurveys, 20
 - getTableCount, 20
 - saveNewAnswers, 20
- Model::User
 - addUser, 27
 - checkCredentials, 27
 - deleteUser, 29
 - getUsers, 29
 - updateLastLogIn, 29
- Model\Base, 6
- Model\Survey, 16
- Model\User, 27
- Redirect, 11
 - Header, 12
 - toController, 13
- run
 - FrontController, 10
- Save_POST_Action
 - Controller::Survey, 24
- saveNewAnswers
 - Model::Survey, 20
- Session, 13
 - __get, 14
 - __set, 14
 - authUser, 14
 - destroy, 14
 - isUserAuthenticated, 15
 - isUserAuthenticatedCheck, 15
 - setTemplate
 - View, 34
 - Show_Action
 - Controller::Survey, 24
 - showSurveyResult
 - Controller::Survey, 24
 - toController
 - Redirect, 13
 - updateLastLogIn
 - Model::User, 29
 - View, 32
 - assign, 33
 - display, 33
 - getTemplateOnFilesystem, 33
 - setTemplate, 34