StreamComposer

Contents

Package default Procedural Elements . . . 2
- config.php . . . 2
  - Define BASEDIR . . . 2
  - Define BASEURL . . . 2
  - Define DB_NAME . . . 2
  - Define DB_PASS . . . 2
  - Define DB_SERVER . . . 2
  - Define DB_USER . . . 2
  - Define PROJECTDIR . . . 2
  - Define PROJECTURL . . . 2
  - Define STREAMDIR . . . 2
  - Define STREAMURL . . . 2
- BrowseMyProjects.php . . . 3
- BrowseProjects.php . . . 4
- CLIController.php . . . 5
- EditProject.php . . . 6
- HTMLController.php . . . 7
- Login.php . . . 8
- Logout.php . . . 9
- NewProject.php . . . 10
- NewStream.php . . . 11
- ProjectCompiler.php . . . 12
- RegisterAccount.php . . . 13
- Survey.php . . . 14
- ViewProject.php . . . 15
- ViewStream.php . . . 16
- WelcomePage.php . . . 17
- index.php . . . 18
- Authentication.php . . . 19
- Database.php . . . 20
- ErrorHandler.php . . . 21
- Menu.php . . . 22
- Project.php . . . 23
- Stream.php . . . 24
- TaskList.php . . . 25
- UploadHandler.php . . . 26
- User.php . . . 27
- EditProjectForm.php . . . 28
- LoginPanel.php . . . 29
- MainTemplate.php . . . 30
- NewStreamForm.php . . . 31
- ProjectsBrowser.php . . . 32
- RegistrationForm.php . . . 33

RelatedItems.php . . . 34
StreamPlayer.php . . . 35
SurveyForm.php . . . 36
TaskListPanel.php . . . 37
UserPanel.php . . . 38
WelcomeScreen.php . . . 39
Package default Classes . . . 40
Class Authentication . . . 40
Method checkPassword . . . 40
Method getUserId . . . 40
Method isLoggedIn . . . 41
Method startSession . . . 41
Class BrowseMyProjects . . . 41
Constructor __construct . . . 42
Class BrowseProjects . . . 42
Constructor __construct . . . 42
Class CLIController . . . 43
Var $commands . . . 43
Var $output . . . 43
Method outputHandler . . . 43
Method showContent . . . 44
Class ContentTemplate . . . 44
Method html_main . . . 44
Class Database . . . 44
Constructor __construct . . . 45
Method affected_rows . . . 45
Method error . . . 45
Method escape_string . . . 46
Method fetch_assoc . . . 46
Method fetch_row . . . 46
Method free_result . . . 47
Method insert . . . 47
Method insert_id . . . 48
Method num_fields . . . 48
Method num_rows . . . 48
Method query . . . 49
Method quote . . . 49
Method select_db . . . 49
Method unescape_string . . . 50
Class DummyBox . . . 50
Constructor __construct . . . 50
Method html_content . . . 51
Class EditProject . . . 51
Constructor __construct . . . 51
Class EditProjectForm . . . 52
Method addStreamToGrid . . . 52
Method html_content . . . 52
Method setAvailableStreams . . . 52
Method setCompileUrl . . . 53

Method setSubmitUrl . . . 53
Class ErrorHandler . . . 53
Method triggerFatalError . . . 54
Class GlassBox . . . 54
Constructor __construct . . . 54
Method html_content . . . 55
Method html_main . . . 55
Class Guest . . . 55
Constructor __construct . . . 55
Class HTMLController . . . 56
Var $content . . . 56
Var $main . . . 56
Var $sidebar . . . 56
Constructor __construct . . . 56
Method showContent . . . 57
Class Login . . . 57
Constructor __construct . . . 58
Class LoginPanel . . . 58
Method html_content . . . 58
Class Logout . . . 58
Constructor __construct . . . 59
Class MainTemplate . . . 59
Constructor __construct . . . 59
Method html_main . . . 60
Method setMenu . . . 60
Class Member . . . 60
Constructor __construct . . . 60
Method createNew . . . 61
Method exists . . . 61
Class Menu . . . 62
Constructor __construct . . . 62
Method getMenuItems . . . 62
Class MyProjectsBrowser . . . 62
Method html_content . . . 63
Method setProjects . . . 63
Class NewProject . . . 63
Constructor __construct . . . 63
Class NewProjectForm . . . 64
Method html_content . . . 64
Method setErrorMessage . . . 64
Method setFormData . . . 64
Class NewStream . . . 65
Constructor __construct . . . 65
Class NewStreamForm . . . 66
Method html_content . . . 66
Method setErrorMessage . . . 66
Method setFormData . . . 66
Class NormalBox . . . 67
Constructor __construct . . . 67

Method html_content . . . 67
Method html_main . . . 67
Class Project . . . 68
Constructor __construct . . . 68
Method canEdit . . . 68
Method createNew . . . 68
Method deleteCurrentStreamFile . . . 69
Method existsTitle . . . 69
Method getAllProjects . . . 69
Method getAuthorId . . . 70
Method getAuthorName . . . 70
Method getDescription . . . 70
Method getEditLink . . . 70
Method getPermalink . . . 70
Method getProjectId . . . 71
Method getStreamFilename . . . 71
Method getStreams . . . 71
Method getStreamURI . . . 71
Method getTags . . . 71
Method getTitle . . . 72
Method getUserProjects . . . 72
Method projectExists . . . 72
Method setStreamFilename . . . 73
Method updateStreams . . . 73
Class ProjectCompiler . . . 73
Constructor __construct . . . 74
Method outputHandler . . . 74
Class ProjectsBrowser . . . 74
Method html_content . . . 75
Method setProjects . . . 75
Class RegisterAccount . . . 75
Constructor __construct . . . 75
Class RegistrationForm . . . 76
Method html_content . . . 76
Method setErrorMessage . . . 76
Method setFormData . . . 76
Class RelatedItems . . . 77
Constructor __construct . . . 77
Method addItem . . . 77
Method html_content . . . 78
Method setItems . . . 78
Class SidebarTemplate . . . 78
Method html_main . . . 78
Class Stream . . . 79
Constructor __construct . . . 79
Method createNew . . . 79
Method getAllStreams . . . 80
Method getAuthorId . . . 80
Method getAuthorName . . . 80

Method getDescription . . . 80
Method getFriendlyLength . . . 81
Method getLength . . . 81
Method getLengthString . . . 81
Method getPermalink . . . 81
Method getProjects . . . 82
Method getStreamFilename . . . 82
Method getStreamId . . . 82
Method getStreamURI . . . 82
Method getTags . . . 83
Method getTitle . . . 83
Method streamExists . . . 83
Class StreamPlayer . . . 83
Method html_content . . . 84
Method setEditUrl . . . 84
Method setStreamURL . . . 84
Class Survey . . . 85
Constructor __construct . . . 85
Class SurveyForm . . . 85
Method html_content . . . 85
Method setPhase . . . 86
Method setQuestion . . . 86
Method setQuestionID . . . 86
Class TaskList . . . 86
Constructor __construct . . . 86
Method flagTask . . . 87
Method getTasks . . . 87
Class TaskListPanel . . . 87
Method html_content . . . 88
Method setTasks . . . 88
Class Template . . . 88
Var $_subtemplates . . . 89
Method adopt . . . 89
Method html_main . . . 89
Class UploadHandler . . . 89
Method convertMp3ToOgg . . . 90
Method getStreamLength . . . 90
Method moveToStreamFolder . . . 90
Class User . . . 91
Var $email_address . . . 91
Var $id . . . 91
Var $is_admin . . . 91
Var $is_guest . . . 92
Var $is_logged . . . 92
Var $real_name . . . 92
Var $username . . . 92
Method getEmailAddress . . . 92
Method getRealName . . . 93
Method getUserId . . . 93

Method getUsername . . . 93
Method isAdmin . . . 93
Method isGuest . . . 93
Method isLoggedIn . . . 94
Class UserPanel . . . 94
Method html_content . . . 94
Class ViewProject . . . 95
Constructor __construct . . . 95
Class ViewStream . . . 95
Constructor __construct . . . 95
Class WelcomePage . . . 96
Constructor __construct . . . 96
Class WelcomeScreen . . . 96
Method html_content . . . 97
Appendices . . . 98
Appendix A - Class Trees . . . 99
default . . . 99

Package default Procedural Elements

config.php

- **Package** default

BASEDIR = dirname(__FILE__) *[line 12]*

config.php Contains general settings for the project.

$Revision: 88 $ $Author: aaron $ $Date: 2010-06-28 18:59:04 +0200 (Mon, 28 Jun 2010) $

BASEURL = 'http://'.(isset($_SERVER['HTTP_HOST'])?$_SERVER['HTTP_HOST']:'streamcomposer') *[line 13]*
DB_NAME = 'streamc' *[line 27]*
DB_PASS = 'CLLRcffZjwYpLC8B' *[line 26]*
DB_SERVER = 'localhost' *[line 24]*
DB_USER = 'streamc' *[line 25]*
PROJECTDIR = BASEDIR.'/static/projects' *[line 20]*
PROJECTURL = BASEURL.'/static/projects' *[line 21]*
STREAMDIR = BASEDIR.'/static/streams' *[line 16]*
STREAMURL = BASEURL.'/static/streams' *[line 17]*

BrowseMyProjects.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 11]*

BrowseMyProject.php Contains the BrowseMyProjects class.

$Revision: 133 $ $Author: aaron $ $Date: 2010-07-03 00:39:51 +0200 (Sat, 03 Jul 2010) $

BrowseProjects.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

BrowseProjects.php Contains the project browser controller.

$Revision: 133 $ $Author: aaron $ $Date: 2010-07-03 00:39:51 +0200 (Sat, 03 Jul 2010) $

CLIController.php

CLIController.php Contains the key CLI controller

$Revision: 133 $ $Author: aaron $ $Date: 2010-07-03 00:39:51 +0200 (Sat, 03 Jul 2010) $

- **Package** default

EditProject.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

EditProject.php Contains the edit project constructor.

$Revision: 133 $ $Author: aaron $ $Date: 2010-07-03 00:39:51 +0200 (Sat, 03 Jul 2010) $

HTMLController.php

HTMLController.php Contains the key HTML controller

$Revision: 133 $ $Author: aaron $ $Date: 2010-07-03 00:39:51 +0200 (Sat, 03 Jul 2010) $

- **Package** default

Login.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

Login.php Contains the controller for logging the user in.

$Revision: 14 $ $Author: aaron $ $Date: 2010-05-30 23:21:07 +0200 (Sun, 30 May 2010) $

Logout.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 13]*

Logout.php Contains the controller for logging the user out.

$Revision: 14 $ $Author: aaron $ $Date: 2010-05-30 23:21:07 +0200 (Sun, 30 May 2010) $

NewProject.php

- **Package** default

require_once **BASEDIR.'/models/Project.php'** *[line 13]*

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

CreateProject.php Contains the class CreateProject.

$Revision: 133 $ $Author: aaron $ $Date: 2010-07-03 00:39:51 +0200 (Sat, 03 Jul 2010) $

NewStream.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

NewStream.php Creates new streams from raw stream data.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

require_once **BASEDIR.'/controllers/CLIController.php'** *[line 12]*

CLIController.php Contains the key CLI controller

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

RegisterAccount.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

RegisterAccount.php Contains the RegisterAccount class.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

Survey.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

Survey.php Contains the survey class.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

ViewProject.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

ViewProject.php Contains the controller for project viewing.

$Revision: 14 $ $Author: aaron $ $Date: 2010-05-30 23:21:07 +0200 (Sun, 30 May 2010) $

ViewStream.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

ViewProject.php Contains the controller for project viewing.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

WelcomePage.php

- **Package** default

require_once **BASEDIR.'/controllers/HTMLController.php'** *[line 12]*

WelcomePage.php Contains the welcoming page constructor.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

index.php

- **Package** default

require **BASEDIR.'/controllers/'.$currentAction[0]** *[line 45]*

require_once **BASEDIR.'/models/User.php'** *[line 23]*

require_once **BASEDIR.'/models/Authentication.php'** *[line 22]*

require_once **BASEDIR.'/models/Database.php'** *[line 18]*

require_once **'config.php'** *[line 12]*

index.php Initiates key classes and determines which controller to use.

$Revision: 99 $ $Author: aaron $ $Date: 2010-06-28 23:09:14 +0200 (Mon, 28 Jun 2010) $

Authentication.php

Authentication.php Contains key class Authentication.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

Database.php

Database.php Contains key class Database.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

ErrorHandler.php

ErrorHandler.php Contains key class Error.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

Menu.php

Menu.php Contains the menu class.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

Project.php

Project.php Contains key class Project.

$Revision: 135 $ $Author: aaron $ $Date: 2010-07-03 01:20:33 +0200 (Sat, 03 Jul 2010) $

- **Package** default

Stream.php

Stream.php Contains key class Stream.

$Revision: 135 $ $Author: aaron $ $Date: 2010-07-03 01:20:33 +0200 (Sat, 03 Jul 2010) $

- **Package** default

TaskList.php

TaskList.php Contains the task list class.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

UploadHandler.php

UploadHandler.php Class used for processing file uploads.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

User.php

User.php Contains key class User, as well as the Guest and Member class that are derived from it.

$Revision: 134 $ $Author: aaron $ $Date: 2010-07-03 01:06:56 +0200 (Sat, 03 Jul 2010) $

- **Package** default

EditProjectForm.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

EditProjectForm.php Contains the edit project template.

$Revision: 107 $ $Author: aaron $ $Date: 2010-06-29 18:50:09 +0200 (Tue, 29 Jun 2010) $

LoginPanel.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

LoginPanel.php Contains the login panel template.

$Revision: 88 $ $Author: aaron $ $Date: 2010-06-28 18:59:04 +0200 (Mon, 28 Jun 2010) $

MainTemplate.php

- **Package** default

require_once **BASEDIR.'/templates/Template.php'** *[line 12]*

MainTemplate.php Defines the key templates.

$Revision: 89 $ $Author: aaron $ $Date: 2010-06-28 19:26:57 +0200 (Mon, 28 Jun 2010) $

NewStreamForm.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

NewStreamForm.php Contains the new stream form template.

$Revision: 76 $ $Author: aaron $ $Date: 2010-06-27 16:21:54 +0200 (Sun, 27 Jun 2010) $

ProjectsBrowser.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

ProjectsBrowser.php Contains the project browser template.

$Revision: 15 $ $Author: aaron $ $Date: 2010-05-30 23:53:43 +0200 (Sun, 30 May 2010) $

RegistrationForm.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

RegistrationForm.php Contains the registration form template.

$Revision: 88 $ $Author: aaron $ $Date: 2010-06-28 18:59:04 +0200 (Mon, 28 Jun 2010) $

RelatedItems.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

RelatedItems.php Contains panel used to display related items.

$Revision: 26 $ $Author: erik $ $Date: 2010-06-22 13:11:10 +0200 (Tue, 22 Jun 2010) $

StreamPlayer.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

StreamPlayer.php Contains a project.

$Revision: 15 $ $Author: aaron $ $Date: 2010-05-30 23:53:43 +0200 (Sun, 30 May 2010) $

SurveyForm.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

SurveyForm.php Contains the survey form.

$Revision: 103 $ $Author: aaron $ $Date: 2010-06-29 00:43:01 +0200 (Tue, 29 Jun 2010) $

TaskListPanel.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

TaskListPanel.php Contains the task list panel.

$Revision: 96 $ $Author: aaron $ $Date: 2010-06-28 21:47:53 +0200 (Mon, 28 Jun 2010) $

UserPanel.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

UserPanel.php Contains the User panel template.

$Revision: 30 $ $Author: raoul $ $Date: 2010-06-24 14:19:24 +0200 (do, 24 jun 2010) $

WelcomeScreen.php

- **Package** default

require_once **BASEDIR.'/templates/MainTemplate.php'** *[line 12]*

WelcomeScreen.php Contains the welcome screen template.

$Revision: 96 $ $Author: aaron $ $Date: 2010-06-28 21:47:53 +0200 (Mon, 28 Jun 2010) $

Package default Classes

Class Authentication

[line 15]

Authentication class, containing various static functions used for account verification and session management.

- **Package** default

void function Authentication::checkPassword($username, $password) *[line 33]*

Function Parameters:

- **$username**
- **$password**

Checks a password for a given username against the database.

- **Static**
- **Access** public

void function Authentication::getUserId($username) *[line 57]*

Function Parameters:

- **$username**

Finds the user id belonging to a certain username.

- **Static**
- **Access** public

void function Authentication::isLoggedIn() *[line 81]*

Verifies whether the user is currently logged in.

- **Static**
- **Access** public

void function Authentication::startSession() *[line 20]*

Initialises a new session, logging several browser-related information for security purposes.

- **Static**
- **Access** public

Class BrowseMyProjects

[line 18]

The controller class that allows users to browse his own projects.

- Requires the user to be logged in.
- Does not parse any user input.

- **Package** default

Constructor *void* function BrowseMyProjects::__construct() *[line 23]*

Constructor for the BrowseMyProjects class.

- **Access** public

Class BrowseProjects

[line 18]

The controller class that allows users to browse all compiled projects.

- Does not parse any user input.

- **Package** default

Constructor *void* function BrowseProjects::__construct() *[line 23]*

Constructor for the BrowseProjects class.

- **Access** public

Class CLIController

[line 14]

The abstract class that allows easy creation of non-html pages, e.g. CLI interfaces.

- **Package** default
- **Abstract Element**

CLIController::$commands

mixed = array() *[line 16]*

- **Access** protected

CLIController::$output

mixed = array() *[line 17]*

- **Access** protected

void function CLIController::outputHandler() *[line 33]*

Function to be called after command execution. Should be implemented by the including class.

- **Abstract Element**

void function CLIController::showContent() *[line 22]*

Main function; will be called by index.php after class instantiation.

- **Access** public

Class ContentTemplate

[line 95]

ContentTemplate

- **Package** default

void function ContentTemplate::html_main() *[line 97]*

- **Access** public

Class Database

[line 14]

The database model used to communicate with the MySQL server.

- **Package** default

Constructor *void* function Database::__construct($server, $user, $password, $name) *[line 26]*
Function Parameters:

- *server:* **$server** server to connect to.
- *user:* **$user** username to use for authentication.
- *password:* **$password** password to use for authentication.
- *name:* **$name** database to select.

Initialises a new database connection.

- **Access** public

void function Database::affected_rows([$connection = null]) *[line 118]*
Function Parameters:

- **$connection**

Returns the amount of rows affected by the previous query.

- **Access** public

void function Database::error([$connection = null]) *[line 102]*
Function Parameters:

- **$connection**

Returns the last MySQL error for a given connection.

- **Access** public

void function Database::escape_string($string) *[line 86]*
Function Parameters:

- **$string**

Escapes a string.

- **Access** public

void function Database::fetch_assoc($resource) *[line 46]*
Function Parameters:

- **$resource**

Fetches a row from a given recordset, using field names as keys.

- **Access** public

void function Database::fetch_row($resource) *[line 54]*
Function Parameters:

- **$resource**

Fetches a row from a given recordset, using numeric keys.

- **Access** public

void function Database::free_result($resource) *[line 62]*

Function Parameters:

- **$resource**

Destroys a given recordset.

- **Access** public

void function Database::insert([$method = 'replace'], $table, $columns, $data, $keys, [$disable_trans = false], [$connection = null]) *[line 332]*

Function Parameters:

- **$method**
- **$table**
- **$columns**
- **$data**
- **$keys**
- **$disable_trans**
- **$connection**

This function can be used to insert data into the database in a secure way.

- **Access** public

void function Database::insert_id([$connection = null]) *[line 126]*
Function Parameters:

- **$connection**

Returns the id of the row created by a previous query.

- **Access** public

void function Database::num_fields($resource) *[line 78]*
Function Parameters:

- **$resource**

Returns the amount of fields in a given recordset.

- **Access** public

void function Database::num_rows($resource) *[line 70]*
Function Parameters:

- **$resource**

Returns the amount of rows in a given recordset.

- **Access** public

void function Database::query($db_string, [$db_values = array()], [$connection = null]) *[line 234]*

Function Parameters:

- **$db_string**
- **$db_values**
- **$connection**

Escapes and quotes a string using values passed, and executes the query.

- **Access** public

void function Database::quote($db_string, [$db_values = array()], [$connection = null]) *[line 306]*

Function Parameters:

- **$db_string**
- **$db_values**
- **$connection**

Escapes and quotes a string just like db_query, but does not execute the query.

Useful for debugging purposes.

- **Access** public

void function Database::select_db($database, [$connection = null]) *[line 110]*

Function Parameters:

- **$database**
- **$connection**

Selects a database on a given connection.

- **Access** public

void function Database::unescape_string($string) *[line 94]*
Function Parameters:

- **$string**

Unescapes a string.

- **Access** public

Class DummyBox

[line 195]

NormalBox

- **Package** default

Constructor *void* function DummyBox::__construct([$title = ''], [$content = '']) *[line 197]*
Function Parameters:

- **$title**
- **$content**

- **Access** public

void function DummyBox::html_content() *[line 203]*

- **Access** public

Class EditProject

[line 20]

The class that users to edit their projects.

- Requires the user to be logged in and have permission to edit the project.
- Parses user input related to positioning, adding and removing streams.
- Passes a list of all streams to the template.

- **Package** default

Constructor *void* function EditProject::__construct() *[line 25]*

Constructor for the BrowseProjects class.

- **Access** public

Class EditProjectForm

[line 14]

NormalBox

- **Package** default

void function EditProjectForm::addStreamToGrid($data) *[line 118]*

Function Parameters:

- **$data**

- **Access** public

void function EditProjectForm::html_content() *[line 21]*

- **Access** protected

void function EditProjectForm::setAvailableStreams($streams) *[line 113]*

Function Parameters:

- **$streams**

- **Access** public

void function EditProjectForm::setCompileUrl($url) *[line 108]*

Function Parameters:

- **$url**

- **Access** public

void function EditProjectForm::setSubmitUrl($url) *[line 103]*

Function Parameters:

- **$url**

- **Access** public

Class ErrorHandler

[line 15]

ErrorHandling class, contains the static triggerFatalError function used to trigger fatal errors, halting all other processes.

- **Package** default

void function ErrorHandler::triggerFatalError($text) *[line 21]*
Function Parameters:

- **$text**

Triggers a fatal error, presenting the message in $text to the user.
!!! TODO: Implement this properly; this is _way_ too controller-ish now.

- **Static**
- **Access** public

Class GlassBox
[line 165]

GlassBox

- **Package** default
- **Abstract Element**

Constructor *void* function GlassBox::__construct([$title = '']) *[line 167]*
Function Parameters:

- **$title**

- **Access** public

void function GlassBox::html_content() *[line 191]*

- **Abstract Element**
- **Access** protected

void function GlassBox::html_main() *[line 172]*

- **Access** public

Class Guest

[line 185]

Guest model; sets typical guest settings to the common attributes.

- **Package** default

Constructor *void* function Guest::__construct() *[line 190]*

Constructor for the Guest class. Sets common attributes.

- **Access** public

Class HTMLController

[line 14]

The abstract class that allows easy creation of html pages.

- **Package** default

HTMLController::$content

mixed = [line 18]

- **Access** protected

HTMLController::$main

mixed = [line 16]

- **Access** protected

HTMLController::$sidebar

mixed = [line 17]

- **Access** protected

Constructor *void* function HTMLController::__construct([$title = 'StreamComposer']) *[line 26]*

Function Parameters:

- *title:* **$title** optional parameter to be used in the HTML header.

Constructor function.

- Initiates main template
- Initiates two widgets: LoginPanel or UserPanel and TaskList

- **Access** public

void function HTMLController::showContent() *[line 71]*

Main function; will be called by index.php after class instantiation.

- **Access** public

Class Login

[line 19]

The controller class that allows users to log in.

- Passes user input to the Authentication model.
- Redirects user after logging in.

- **Package** default

Constructor *void* function Login::__construct() *[line 24]*

Constructor for the Login class.

- **Access** public

Class LoginPanel

[line 14]

GlassBox

- **Package** default

void function LoginPanel::html_content() *[line 16]*

- **Access** protected

Class Logout

[line 20]

The controller class that allows users to log out.

- Clears the user's session.
- Redirects user after logging out.

- **Package** default

Constructor *void* function Logout::__construct() *[line 25]*

Constructor for the Logout class.

- **Access** public

Class MainTemplate

[line 17]

MainTemplate

- **Package** default

Constructor *void* function MainTemplate::__construct([$title = '']) *[line 22]*

Function Parameters:

- **$title**

- **Access** public

void function MainTemplate::html_main() *[line 27]*

- **Access** public

void function MainTemplate::setMenu($menu) *[line 86]*

Function Parameters:

- **$menu**

- **Access** public

Class Member

[line 84]

Member model; contains methods that only apply to (future) members.

- **Package** default

Constructor *void* function Member::__construct($id) *[line 89]*

Function Parameters:

- **$id**

Constructor for the Member class. Loads user data.

- **Access** public

void function Member::createNew([$data = array()]) *[line 121]*
Function Parameters:

- *data* **$data**

Creates a new member from the data provided.

- **Static**
- **Access** public

void function Member::exists($username) *[line 164]*
Function Parameters:

- *username* **$username** to check

Returns whether a username already exists.

- **Static**
- **Access** public

Class Menu

[line 14]

Menu class, contains the menu items to be displayed in the template.

- **Package** default

Constructor *void* function Menu::__construct() *[line 21]*

Constructor for the Menu class. Initialises the menu items and marks the active one as such.

- **Access** public

void function Menu::getMenuItems() *[line 91]*

Returns a copy of all menu items.

- **Access** public

Class MyProjectsBrowser

[line 12]

MyProjectsBrowser.php Contains the class MyProjectsBrowser

$Revision: 92 $ $Author: aaron $ $Date: 2010-06-28 20:11:46 +0200 (Mon, 28 Jun 2010) $

- **Package** default

void function MyProjectsBrowser::html_content() *[line 21]*

- **Access** protected

void function MyProjectsBrowser::setProjects($projects) *[line 16]*

Function Parameters:

- **$projects**

- **Access** public

Class NewProject

[line 21]

The controller class that allows users to create a new project.

- Requires the user to be logged in.
- Parses title and description from form.
- Redirects user to edit screen if successful.

- **Package** default

Constructor *void* function NewProject::__construct() *[line 26]*

Constructor for the NewProject class.

- **Access** public

Class NewProjectForm

[line 7]

ProjectCreationForm.php contains the ProjectCreationForm class.

- **Package** default

void function NewProjectForm::html_content() *[line 13]*

- **Access** protected

void function NewProjectForm::setErrorMessage($err_msg) *[line 41]*

Function Parameters:

- **$err_msg**

- **Access** public

void function NewProjectForm::setFormData([$data = array()]) *[line 35]*

Function Parameters:

- **$data**

    - **Access** public

Class NewStream

[line 20]

The controller class that allows users to create a new stream.

- Requires the user to be logged in.
- Parses title and description from form.
- Redirects user to stream player if successful.

    - **Package** default

Constructor *void* function NewStream::__construct() *[line 25]*

Constructor for the NewStream class.

    - **Access** public

Class NewStreamForm

[line 14]

NormalBox

- **Package** default

void function NewStreamForm::html_content() *[line 21]*

- **Access** protected

void function NewStreamForm::setErrorMessage($err_msg) *[line 53]*

Function Parameters:

- **$err_msg**

- **Access** public

void function NewStreamForm::setFormData([$data = array()]) *[line 47]*

Function Parameters:

- **$data**

- **Access** public

Class NormalBox

[line 133]

NormalBox

- **Package** default
- **Abstract Element**

Constructor *void* function NormalBox::__construct([$title = '']) *[line 135]*

Function Parameters:

- **$title**

- **Access** public

void function NormalBox::html_content() *[line 159]*

- **Abstract Element**
- **Access** protected

void function NormalBox::html_main() *[line 140]*

- **Access** public

Class Project

[line 14]

Project class: contains pretty much everything there is to know of and do with a project.

- **Package** default

Constructor *void* function Project::__construct($project_id, [$rowdata = array()]) *[line 30]*

Function Parameters:

- *project_id:* **$project_id** id of the project to initiate.
- *rowdata:* **$rowdata** optional rowdata to use instead of querying from the database.

Constructs a new project.

- **Access** public

void function Project::canEdit() *[line 332]*

Returns whether a user can edit this very project.

- **Access** public

void function Project::createNew([$data = array()]) *[line 67]*

Function Parameters:

- **$data**

Creates a new project from the given data.

- **Static**
- **Access** public

void function Project::deleteCurrentStreamFile() *[line 342]*

Deletes the current stream file.

- **Access** public

void function Project::existsTitle($title) *[line 97]*

Function Parameters:

- **$title**

Checks whether a project with a given title already exists.

- **Static**
- **Access** public

void function Project::getAllProjects() *[line 137]*

Returns a list of all compiled projects.

- **Static**
- **Access** public

void function Project::getAuthorId() *[line 260]*

Returns the author id.

- **Access** public

void function Project::getAuthorName() *[line 268]*

Returns the author's name.

- **Access** public

void function Project::getDescription() *[line 284]*

Returns the project description.

- **Access** public

void function Project::getEditLink() *[line 324]*

Returns an URL to the project editor.

- **Access** public

void function Project::getPermalink() *[line 316]*

Returns an URL to the stream player.

- **Access** public

void function Project::getProjectId() *[line 252]*

Returns the project id.

- **Access** public

void function Project::getStreamFilename() *[line 300]*

Returns the stream filename.

- **Access** public

void function Project::getStreams() *[line 187]*

Returns a list of all streams used by this project.

- **Access** public

void function Project::getStreamURI() *[line 308]*

Returns a permalink for the project's stream file.

- **Access** public

void function Project::getTags() *[line 292]*

Returns the tags linked to the project.

- **Access** public

void function Project::getTitle() *[line 276]*

Returns the project title.

- **Access** public

void function Project::getUserProjects($id_user) *[line 162]*

Function Parameters:

- **$id_user**

Returns a list of all projects created by a certain user.

- **Static**
- **Access** public

void function Project::projectExists($project_id) *[line 117]*

Function Parameters:

- **$project_id**

Checks whether a project with a given id exists.

- **Static**
- **Access** public

void function Project::setStreamFilename($filename) *[line 364]*
Function Parameters:

- **$filename**

Updates the database record to include a given filename as its stream.

- **Access** public

void function Project::updateStreams($list) *[line 223]*
Function Parameters:

- **$list**

Updates the streams used by this project with the list provided.

- **Access** public

Class ProjectCompiler

[line 20]

The controller class that allows users to compile a project in the background.

- Requires the user to be logged in.
- Compiles project via SOX commands.
- Outputs JSON data.

- **Package** default

Constructor *void* function ProjectCompiler::__construct() *[line 25]*

Constructor for the NewStream class.

- **Access** public

void function ProjectCompiler::outputHandler() *[line 90]*

Function called after command execution.

- **Access** public

Class ProjectsBrowser

[line 14]

NormalBox

- **Package** default

void function ProjectsBrowser::html_content() *[line 23]*

- **Access** protected

void function ProjectsBrowser::setProjects($projects) *[line 18]*

Function Parameters:

- **$projects**

- **Access** public

Class RegisterAccount

[line 20]

The controller class that allows users to register an account.

- Requires the user to be logged out.
- Parses submitted registration form data.
- Creates account and logs in if successful.

- **Package** default

Constructor *void* function RegisterAccount::__construct() *[line 25]*

Constructor for the RegisterAccount class.

- **Access** public

Class RegistrationForm

[line 14]

NormalBox

- **Package** default

void function RegistrationForm::html_content() *[line 21]*

- **Access** protected

void function RegistrationForm::setErrorMessage($err_msg) *[line 58]*

Function Parameters:

- **$err_msg**

- **Access** public

void function RegistrationForm::setFormData([$data = array()]) *[line 51]*

Function Parameters:

- **$data**

- **Access** public

Class RelatedItems

[line 14]

GlassBox

- **Package** default

Constructor *void* function RelatedItems::__construct([$title = 'Related items'], [$description = '']) *[line 19]*

Function Parameters:

- **$title**
- **$description**

- **Access** public

void function RelatedItems::addItem($item) *[line 33]*

Function Parameters:

- **$item**

- **Access** public

void function RelatedItems::html_content() *[line 38]*

- **Access** protected

void function RelatedItems::setItems($items) *[line 25]*

Function Parameters:

- **$items**

- **Access** public

Class SidebarTemplate

[line 114]

SidebarTemplate

- **Package** default

void function SidebarTemplate::html_main() *[line 116]*

- **Access** public

Class Stream

[line 14]

Stream class: contains pretty much everything there is to know of and do with a stream.

- **Package** default

Constructor *void* function Stream::__construct($stream_id, [$rowdata = array()]) *[line 31]*

Function Parameters:

- *stream_id:* **$stream_id** id of the stream to initiate.
- *rowdata:* **$rowdata** optional rowdata to use instead of querying from the database.

Constructs a new stream.

- **Access** public

void function Stream::createNew([$data = array()]) *[line 69]*

Function Parameters:

- **$data**

Creates a new stream from the given data.

- **Static**
- **Access** public

void function Stream::getAllStreams() *[line 125]*

Returns a list of all available streams.

- **Static**
- **Access** public

void function Stream::getAuthorId() *[line 190]*

Returns the author id.

- **Access** public

void function Stream::getAuthorName() *[line 198]*

Returns the author name.

- **Access** public

void function Stream::getDescription() *[line 214]*

Returns the stream description.

- **Access** public

void function Stream::getFriendlyLength($length) *[line 246]*

Function Parameters:

- **$length**

Returns a formatted stream length.

- **Static**
- **Access** public

void function Stream::getLength() *[line 238]*

Returns the stream's length as an integer.

- **Access** public

void function Stream::getLengthString($length) *[line 263]*

Function Parameters:

- **$length**

Returns a formatted stream length.

- **Static**
- **Access** public

void function Stream::getPermalink() *[line 280]*

Returns an URL to the stream player.

- **Access** public

void function Stream::getProjects() *[line 149]*

Returns a list of all projects using the current stream.

- **Access** public

void function Stream::getStreamFilename() *[line 230]*

Returns the stream filename.

- **Access** public

void function Stream::getStreamId() *[line 182]*

Returns the stream id.

- **Access** public

void function Stream::getStreamURI() *[line 272]*

Returns a permalink for the stream file.

- **Access** public

void function Stream::getTags() *[line 222]*

Returns the tags linked to this stream.

- **Access** public

void function Stream::getTitle() *[line 206]*

Returns the stream title.

- **Access** public

void function Stream::streamExists($stream_id) *[line 105]*

Function Parameters:

- **$stream_id**

Checks whether a stream with a given id exists.

- **Static**
- **Access** public

Class StreamPlayer

[line 14]

NormalBox

- **Package** default

void function StreamPlayer::html_content() *[line 31]*

- **Access** protected

void function StreamPlayer::setEditUrl($url) *[line 26]*
Function Parameters:

- **$url**

- **Access** public

void function StreamPlayer::setStreamURL($url, [$description = '']) *[line 20]*
Function Parameters:

- **$url**
- **$description**

- **Access** public

Class Survey

[line 19]

The controller class that allows users to fill out the survey.

- Requires the user to be logged in.
- Parses submitted survey form data.

- **Package** default

Constructor *void* function Survey::__construct() *[line 24]*

Constructor for the Survey class.

- **Access** public

Class SurveyForm

[line 14]

NormalBox

- **Package** default

void function SurveyForm::html_content() *[line 20]*

- **Access** protected

void function SurveyForm::setPhase($phase) *[line 75]*
Function Parameters:

- **$phase**

void function SurveyForm::setQuestion($question) *[line 80]*
Function Parameters:

- **$question**

void function SurveyForm::setQuestionID($id) *[line 85]*
Function Parameters:

- **$id**

Class TaskList

[line 14]

The TaskList model is used to load the tasks list and mark tasks as completed.

- **Package** default

Constructor *void* function TaskList::__construct($id_user) *[line 22]*
Function Parameters:

- **$id_user**

Constructor for the TaskList class. Loads tasks for a given user.

- **Access** public

void function TaskList::flagTask($task, [$force_id_user = null]) *[line 57]*

Function Parameters:

- **$task**
- **$force_id_user**

Flags a task as completed.

- **Access** public

void function TaskList::getTasks() *[line 80]*

Returns a copy of all tasks, both incomplete and complete.

- **Access** public

Class TaskListPanel

[line 14]

GlassBox

- **Package** default

void function TaskListPanel::html_content() *[line 18]*

- **Access** protected

void function TaskListPanel::setTasks($tasks) *[line 34]*

Function Parameters:

- **$tasks**

- **Access** public

Class Template

[line 11]

Template.php Contains key Template interface.

$Revision: 16 $ $Author: aaron $ $Date: 2010-05-31 01:23:19 +0200 (Mon, 31 May 2010) $

- **Package** default
- **Abstract Element**

Template::$_subtemplates

mixed = array() *[line 13]*

- **Access** protected

void function Template::adopt($template, [$position = 'end']) *[line 17]*

Function Parameters:

- *Template* **$template**
- **$position**

- **Access** public

void function Template::html_main() *[line 15]*

- **Abstract Element**
- **Access** public

Class UploadHandler

[line 14]

The UploadHandler model contains several static functions used when uploading streams.

- **Package** default

generated function UploadHandler::convertMp3ToOgg($filename) *[line 32]*

Function Parameters:

- **$filename**

Converts an MP3 file to an OGG file using the mpg321 and oggenc binaries.

- **Static**
- **Access** public

void function UploadHandler::getStreamLength($filename) *[line 19]*

Function Parameters:

- **$filename**

Calculates the length of an audio file by providing its filename.

- **Static**
- **Access** public

generated function UploadHandler::moveToStreamFolder($filename) *[line 55]*

Function Parameters:

- **$filename**

Moves a stream to a folder, generating a random filename in the process.

- **Static**

- **Access** public

Class User

[line 14]

User model; contains attributes and methods shared by both members and guests.

- **Package** default
- **Abstract Element**

User::$email_address

mixed = [line 21]

- **Access** protected

User::$id

mixed = [line 19]

- **Access** protected

User::$is_admin

mixed = [line 18]

- **Access** protected

User::$is_guest

mixed = [line 17]

- **Access** protected

User::$is_logged

mixed = [line 16]

- **Access** protected

User::$real_name

mixed = [line 22]

- **Access** protected

User::$username

mixed = [line 20]

- **Access** protected

void function User::getEmailAddress() *[line 51]*

Returns email address.

- **Access** public

void function User::getRealName() *[line 43]*

Returns real name.

- **Access** public

void function User::getUserId() *[line 27]*

Returns user id.

- **Access** public

void function User::getUsername() *[line 35]*

Returns username.

- **Access** public

void function User::isAdmin() *[line 75]*

Returns whether user is an administrator.

- **Access** public

void function User::isGuest() *[line 67]*

Returns whether user is a guest.

- **Access** public

void function User::isLoggedIn() *[line 59]*

Returns whether user is logged in.

- **Access** public

Class UserPanel

[line 14]

GlassBox

- **Package** default

void function UserPanel::html_content() *[line 16]*

- **Access** protected

Class ViewProject

[line 18]

The controller class that allows users to view a compiled project.

- Initialises the player and loads the project into it.

- **Package** default

Constructor *void* function ViewProject::__construct() *[line 23]*

Constructor for the ViewProject class.

- **Access** public

Class ViewStream

[line 18]

The controller class that allows users to view an individual stream.

- Initialises the player and loads the stream into it.

- **Package** default

Constructor *void* function ViewStream::__construct() *[line 23]*

Constructor for the ViewStream class.

- **Access** public

Class WelcomePage

[line 17]

Loads and displays a very simple welcoming page.

- **Package** default

Constructor *void* function WelcomePage::__construct() *[line 22]*

Constructor for the WelcomePage class.

- **Access** public

Class WelcomeScreen

[line 14]

NormalBox

- **Package** default

void function WelcomeScreen::html_content() *[line 16]*

- **Access** protected

Appendices

Appendix A - Class Trees

Package default

Authentication

- Authentication

CLIController

- CLIController
  - ProjectCompiler

Database

- Database

ErrorHandler

- ErrorHandler

HTMLController

- HTMLController
  - BrowseMyProjects
  - BrowseProjects
  - EditProject
  - Login
  - Logout
  - NewProject

- NewStream
- RegisterAccount
- Survey
- ViewProject
- ViewStream
- WelcomePage

Menu

- Menu

Project

- Project

Stream

- Stream

TaskList

- TaskList

Template

- Template
  - ContentTemplate
  - GlassBox
    - LoginPanel
    - RelatedItems
    - TaskListPanel
    - UserPanel

- MainTemplate
- NormalBox
  - DummyBox
  - EditProjectForm
  - MyProjectsBrowser
  - NewProjectForm
  - NewStreamForm
  - ProjectsBrowser
  - RegistrationForm
  - StreamPlayer
  - SurveyForm
  - WelcomeScreen

- SidebarTemplate

UploadHandler

- UploadHandler

User

- User
  - Guest
  - Member

Index

A

Authentication::isLoggedIn() . . . 41
Verifies whether the user is currently logged in.
Authentication::startSession() . . . 41
Initialises a new session, logging several browser-related information for security purposes.
Authentication::getUserId() . . . 40
Finds the user id belonging to a certain username.
Authentication::checkPassword() . . . 40
Checks a password for a given username against the database.
Authentication . . . 40
Authentication class, containing various static functions used for account verification and session management.
Authentication.php . . . 19
Authentication.php
Contains key class Authentication.

B

BrowseMyProjects . . . 41
The controller class that allows users to browse his own projects.
BrowseProjects . . . 42
The controller class that allows users to browse all compiled projects.
BrowseProjects.php . . . 4
BrowseMyProjects.php . . . 3
BASEURL . . . 2
BASEDIR . . . 2
config.php
Contains general settings for the project.

C

constructor NewStream::__construct() . . . 65
Constructor for the NewStream class.
constructor NormalBox::__construct() . . . 67
constructor Project::__construct() . . . 68
Constructs a new project.
constructor NewProject::__construct() . . . 63
Constructor for the NewProject class.
constructor Menu::__construct() . . . 62
Constructor for the Menu class. Initialises the menu items and marks the active one as such.
constructor MainTemplate::__construct() . . . 59
constructor Member::__construct() . . . 60
Constructor for the Member class. Loads user data.

constructor ProjectCompiler::__construct() . . . 74
Constructor for the NewStream class.
constructor RegisterAccount::__construct() . . . 75
Constructor for the RegisterAccount class.
constructor ViewProject::__construct() . . . 95
Constructor for the ViewProject class.
constructor ViewStream::__construct() . . . 95
Constructor for the ViewStream class.
constructor WelcomePage::__construct() . . . 96
Constructor for the WelcomePage class.
constructor TaskList::__construct() . . . 86
Constructor for the TaskList class. Loads tasks for a given user.
constructor Survey::__construct() . . . 85
Constructor for the Survey class.
constructor RelatedItems::__construct() . . . 77
constructor Stream::__construct() . . . 79
Constructs a new stream.
constructor Logout::__construct() . . . 59
Constructor for the Logout class.
constructor Login::__construct() . . . 58
Constructor for the Login class.
CLIController::$commands . . . 43
CLIController::$output . . . 43
CLIController::outputHandler() . . . 43
Function to be called after command execution. Should be implemented by the including class.
CLIController . . . 43
The abstract class that allows easy creation of non-html pages, e.g. CLI interfaces.
constructor BrowseProjects::__construct() . . . 42
Constructor for the BrowseProjects class.
CLIController.php . . . 5
CLIController.php
Contains the key CLI controller
constructor BrowseMyProjects::__construct() . . . 42
Constructor for the BrowseMyProjects class.
CLIController::showContent() . . . 44
Main function; will be called by index.php after class instantiation.
ContentTemplate . . . 44
ContentTemplate
constructor GlassBox::__construct() . . . 54
constructor Guest::__construct() . . . 55
Constructor for the Guest class. Sets common attributes.
constructor HTMLController::__construct() . . . 56
Constructor function.
constructor EditProject::__construct() . . . 51
Constructor for the BrowseProjects class.
constructor DummyBox::__construct() . . . 50
ContentTemplate::html_main() . . . 44
constructor Database::__construct() . . . 45
Initialises a new database connection.
config.php . . . 2

D

Database::num_rows() . . . 48
Returns the amount of rows in a given recordset.
Database::num_fields() . . . 48
Returns the amount of fields in a given recordset.
Database::insert_id() . . . 48
Returns the id of the row created by a previous query.
Database::insert() . . . 47
This function can be used to insert data into the database in a secure way.
Database::query() . . . 49
Escapes and quotes a string using values passed, and executes the query.
Database::quote() . . . 49
Escapes and quotes a string just like db_query, but does not execute the query.
DummyBox::html_content() . . . 51
DummyBox . . . 50
NormalBox
Database::unescape_string() . . . 50
Unescapes a string.
Database::select_db() . . . 49
Selects a database on a given connection.
Database::free_result() . . . 47
Destroys a given recordset.
Database::fetch_row() . . . 46
Fetches a row from a given recordset, using numeric keys.
Database.php . . . 20
Database.php
Contains key class Database.
DB_USER . . . 2
DB_SERVER . . . 2
DB_PASS . . . 2
Database . . . 44
The database model used to communicate with the MySQL server.
Database::affected_rows() . . . 45
Returns the amount of rows affected by the previous query.
Database::fetch_assoc() . . . 46
Fetches a row from a given recordset, using field names as keys.
Database::escape_string() . . . 46
Escapes a string.
Database::error() . . . 45
Returns the last MySQL error for a given connection.
DB_NAME . . . 2

E

EditProjectForm::setCompileUrl() . . . 53
EditProjectForm::setAvailableStreams() . . . 52
EditProjectForm::setSubmitUrl() . . . 53
ErrorHandler . . . 53
ErrorHandling class, contains the static triggerFatalError function used to trigger fatal errors, halting all other processes.
ErrorHandler::triggerFatalError() . . . 54

Triggers a fatal error, presenting the message in $text to the user.
EditProjectForm::html_content() . . . 52
EditProjectForm::addStreamToGrid() . . . 52
ErrorHandler.php . . . 21
ErrorHandler.php
Contains key class Error.
EditProjectForm.php . . . 28
EditProject . . . 51
The class that users to edit their projects.
EditProjectForm . . . 52
NormalBox
EditProject.php . . . 6

G

Guest . . . 55
Guest model; sets typical guest settings to the common attributes.
GlassBox::html_main() . . . 55
GlassBox::html_content() . . . 55
GlassBox . . . 54
GlassBox

H

HTMLController::$sidebar . . . 56
HTMLController::showContent() . . . 57
Main function; will be called by index.php after class instantiation.
HTMLController::$main . . . 56
HTMLController::$content . . . 56
HTMLController . . . 56
The abstract class that allows easy creation of html pages.
HTMLController.php . . . 7
HTMLController.php
Contains the key HTML controller

I

index.php . . . 18

L

LoginPanel::html_content() . . . 58
Logout . . . 58
The controller class that allows users to log out.
LoginPanel . . . 58
GlassBox
Login . . . 57
The controller class that allows users to log in.
Logout.php . . . 9

LoginPanel.php . . . 29
Login.php . . . 8

M

Menu::getMenuItems() . . . 62
Returns a copy of all menu items.
Menu . . . 62
Menu class, contains the menu items to be displayed in the template.
MyProjectsBrowser . . . 62
MyProjectsBrowser.php
Contains the class MyProjectsBrowser
MyProjectsBrowser::html_content() . . . 63
MyProjectsBrowser::setProjects() . . . 63
Member::exists() . . . 61
Returns whether a username already exists.
Member::createNew() . . . 61
Creates a new member from the data provided.
MainTemplate . . . 59
MainTemplate
MainTemplate.php . . . 30
MainTemplate::html_main() . . . 60
MainTemplate::setMenu() . . . 60
Member . . . 60
Member model; contains methods that only apply to (future) members.
Menu.php . . . 22
Menu.php
Contains the menu class.

N

NewStreamForm::setErrorMessage() . . . 66
NewStreamForm::html_content() . . . 66
NewStreamForm . . . 66
NormalBox
NewStreamForm::setFormData() . . . 66
NormalBox . . . 67
NormalBox
NormalBox::html_main() . . . 67
NormalBox::html_content() . . . 67
NewStream . . . 65
The controller class that allows users to create a new stream.
NewProjectForm::setFormData() . . . 64
NewStreamForm.php . . . 31
NewStream.php . . . 11
NewProject . . . 63
The controller class that allows users to create a new project.
NewProjectForm . . . 64
ProjectCreationForm.php
contains the ProjectCreationForm class.
NewProjectForm::setErrorMessage() . . . 64

NewProjectForm::html_content() . . . 64
NewProject.php . . . 10

P

Project::getTitle() . . . 72
Returns the project title.
Project::getUserProjects() . . . 72
Returns a list of all projects created by a certain user.
Project::getTags() . . . 71
Returns the tags linked to the project.
Project::getStreamURI() . . . 71
Returns a permalink for the project's stream file.
Project::getStreamFilename() . . . 71
Returns the stream filename.
Project::getStreams() . . . 71
Returns a list of all streams used by this project.
Project::projectExists() . . . 72
Checks whether a project with a given id exists.
Project::setStreamFilename() . . . 73
Updates the database record to include a given filename as its stream.
ProjectsBrowser::html_content() . . . 75
ProjectsBrowser::setProjects() . . . 75
ProjectsBrowser . . . 74
NormalBox
ProjectCompiler::outputHandler() . . . 74
Function called after command execution.
Project::updateStreams() . . . 73
Updates the streams used by this project with the list provided.
ProjectCompiler . . . 73
The controller class that allows users to compile a project in the background.
Project::getProjectId() . . . 71
Returns the project id.
Project::getPermalink() . . . 70
Returns an URL to the stream player.
Project . . . 68
Project class: contains pretty much everything there is to know of and do with a project.
Project::canEdit() . . . 68
Returns whether a user can edit this very project.
ProjectsBrowser.php . . . 32
Project.php . . . 23
Project.php
Contains key class Project.
PROJECTURL . . . 2
ProjectCompiler.php . . . 12
Project::createNew() . . . 68
Creates a new project from the given data.
Project::deleteCurrentStreamFile() . . . 69
Deletes the current stream file.
Project::getDescription() . . . 70
Returns the project description.
Project::getEditLink() . . . 70

Returns an URL to the project editor.
Project::getAuthorName() . . . 70
Returns the author's name.
Project::getAuthorId() . . . 70
Returns the author id.
Project::existsTitle() . . . 69
Checks whether a project with a given title already exists.
Project::getAllProjects() . . . 69
Returns a list of all compiled projects.
PROJECTDIR . . . 2

R

RelatedItems . . . 77
GlassBox
RegistrationForm::setFormData() . . . 76
RelatedItems::addItem() . . . 77
RelatedItems::html_content() . . . 78
RelatedItems::setItems() . . . 78
RegistrationForm::setErrorMessage() . . . 76
RegistrationForm::html_content() . . . 76
RegistrationForm.php . . . 33
RelatedItems.php . . . 34
RegisterAccount . . . 75
The controller class that allows users to register an account.
RegistrationForm . . . 76
NormalBox
RegisterAccount.php . . . 13

S

Stream::getTitle() . . . 83
Returns the stream title.
Stream::streamExists() . . . 83
Checks whether a stream with a given id exists.
StreamPlayer . . . 83
NormalBox
Stream::getTags() . . . 83
Returns the tags linked to this stream.
Stream::getStreamURI() . . . 82
Returns a permalink for the stream file.
Stream::getStreamFilename() . . . 82
Returns the stream filename.
Stream::getStreamId() . . . 82
Returns the stream id.
StreamPlayer::html_content() . . . 84
StreamPlayer::setEditUrl() . . . 84
SurveyForm::setPhase() . . . 86
SurveyForm::setQuestion() . . . 86
SurveyForm::setQuestionID() . . . 86
SurveyForm::html_content() . . . 85

SurveyForm . . . 85
NormalBox
StreamPlayer::setStreamURL() . . . 84
Survey . . . 85
The controller class that allows users to fill out the survey.
Stream::getProjects() . . . 82
Returns a list of all projects using the current stream.
Stream::getPermalink() . . . 81
Returns an URL to the stream player.
SurveyForm.php . . . 36
SidebarTemplate . . . 78
SidebarTemplate
SidebarTemplate::html_main() . . . 78
StreamPlayer.php . . . 35
Stream.php . . . 24
Stream.php
Contains key class Stream.
STREAMURL . . . 2
Survey.php . . . 14
Stream . . . 79
Stream class: contains pretty much everything there is to know of and do with a stream.
Stream::createNew() . . . 79
Creates a new stream from the given data.
Stream::getFriendlyLength() . . . 81
Returns a formatted stream length.
Stream::getLength() . . . 81
Returns the stream's length as an integer.
Stream::getLengthString() . . . 81
Returns a formatted stream length.
Stream::getDescription() . . . 80
Returns the stream description.
Stream::getAuthorName() . . . 80
Returns the author name.
Stream::getAllStreams() . . . 80
Returns a list of all available streams.
Stream::getAuthorId() . . . 80
Returns the author id.
STREAMDIR . . . 2

T

Template . . . 88
Template.php
Contains key Template interface.
TaskListPanel::setTasks() . . . 88
Template::$_subtemplates . . . 89
Template::adopt() . . . 89
Template::html_main() . . . 89
TaskListPanel::html_content() . . . 88
TaskListPanel . . . 87
GlassBox
TaskListPanel.php . . . 37

TaskList . . . 86
The TaskList model is used to load the tasks list and mark tasks as completed.
TaskList::flagTask() . . . 87
Flags a task as completed.
TaskList::getTasks() . . . 87
Returns a copy of all tasks, both incomplete and complete.
TaskList.php . . . 25
TaskList.php
Contains the task list class.

U

User::getRealName() . . . 93
Returns real name.
User::getUserId() . . . 93
Returns user id.
User::getEmailAddress() . . . 92
Returns email address.
User::$username . . . 92
User::$real_name . . . 92
User::getUsername() . . . 93
Returns username.
User::isAdmin() . . . 93
Returns whether user is an administrator.
UserPanel::html_content() . . . 94
UserPanel . . . 94
GlassBox
User::isLoggedIn() . . . 94
Returns whether user is logged in.
User::isGuest() . . . 93
Returns whether user is a guest.
User::$is_logged . . . 92
User::$is_guest . . . 92
UploadHandler::convertMp3ToOgg() . . . 90
Converts an MP3 file to an OGG file using the mpg321 and oggenc binaries.
UploadHandler . . . 89
The UploadHandler model contains several static functions used when uploading streams.
UserPanel.php . . . 38
User.php . . . 27
User.php
Contains key class User, as well as the Guest and Member class that are derived from it.
UploadHandler::getStreamLength() . . . 90
Calculates the length of an audio file by providing its filename.
UploadHandler::moveToStreamFolder() . . . 90
Moves a stream to a folder, generating a random filename in the process.
User::$is_admin . . . 91
User::$id . . . 91
User::$email_address . . . 91
User . . . 91
User model; contains attributes and methods shared by both members and guests.
UploadHandler.php . . . 26
UploadHandler.php

Class used for processing file uploads.

V

ViewStream 95
The controller class that allows users to view an individual stream.
ViewProject 95
The controller class that allows users to view a compiled project.
ViewStream.php 16
ViewProject.php 15

W

WelcomeScreen::html_content() 97
WelcomeScreen 96
NormalBox
WelcomePage 96
Loads and displays a very simple welcoming page.
WelcomeScreen.php 39
WelcomePage.php 17