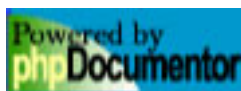


Caramel CMS Documentation



Contents

Package admin Procedural Elements	2
ajaxCommand.php	2
Define BASEDIR	2
Define TEMPLATEDIR	2
index.php	3
Package default Procedural Elements	5
index.php	5
Define BASEDIR	5
Define TEMPLATEDIR	5
Package inc Procedural Elements	7
BackendController.php	7
FrontendController.php	8
Package inc Classes	9
Class BackendController	9
Class Constant VERSION	9
Class Constant VERSION_DATE	9
Constructor BackendController	9
Method ajaxAction	9
Method backendOutputAction	10
Method getBaseUrl	10
Method getGlobalConfig	10
Method getMeta	10
Method getParametersBefore	11
Method getParametersBehind	11
Method getSession	11
Method headTagAction	11
Method logoutAction	12
Method sessionAction	12
Method versionInformationAction	12
Class FrontendController	12
Class Constant VERSION	13
Class Constant VERSION_DATE	13
Constructor FrontendController	13
Method frontendOutputAction	13
Method getBaseUrl	13
Method getDisplay	13
Method getFooter	14
Method getJsAndCss	14
Method getLanguage	14
Method getLanguageSelector	14
Method getNavigationLinks	14
Method getParametersBefore	15

Method <code>getParametersBehind</code>	15
Method <code>getSocialbar</code>	15
Method <code>headTagAction</code>	15
Method <code>languageCodeAction</code>	16
Method <code>languageRedirectAction</code>	16
Method <code>versionInformationAction</code>	16
Class <code>ConfigurationModel</code>	16
Method <code>getAdminAction</code>	17
Method <code>getAdminConfigStringAction</code>	17
Method <code>getConfigStringAction</code>	17
Method <code>getConfigurationModel</code>	18
Method <code>getGlobalsAction</code>	18
Method <code>reloadAdminConfigFile</code>	18
Method <code>reloadConfigFile</code>	18
Method <code>setAdminAction</code>	19
Method <code>setAdminConfigString</code>	19
Method <code>setConfigString</code>	19
Method <code>setGlobalsAction</code>	20
Class <code>DatabaseModel</code>	20
Method <code>getAllLanguagesAction</code>	21
Method <code>getAllMetaTagsAction</code>	21
Method <code>getAllPageNamesAction</code>	21
Method <code>getDatabaseModel</code>	21
Method <code>getWebsiteContentAction</code>	22
Method <code>getWebsiteNavigationAction</code>	22
Method <code>getWebsiteTitleAction</code>	22
Class <code>CaramelException</code>	23
Constructor <code>construct</code>	23
Method <code>getDetails</code>	24
Class <code>TemplateView</code>	24
Constructor <code>TemplateView</code>	24
Method <code>addCssJs</code>	25
Method <code>assign</code>	25
Method <code>render</code>	25
Method <code>returnTemplate</code>	25
Method <code>setTemplateFile</code>	26
Package <code>template</code> Procedural Elements	28
index.tpl-alt.php	28
index.tpl.php	29
Appendices	30
Appendix A - Class Trees	31
admin	31
default	31
inc	31
template	32

Package admin Procedural Elements

ajaxCommand.php

- **Package** admin

```
BASEDIR = substr($_SERVER["SCRIPT_FILENAME"],0,strrpos($_SERVER["SCRIPT_FILENAME"],"/admin/"))  
[line 10]
```

- **Var** BASEDIR Constant which represents the basedir of the server

```
require_once BASEDIR.'/inc/controller/BackendController.php' [line 16]
```

```
TEMPLATEDIR =  
"../".substr(dirname($_SERVER["SCRIPT_NAME"]),0,strrpos(dirname($_SERVER["SCRIPT_NAME"]),"/admin/"))."t  
emplate" [line 14]
```

- **Var** TEMPLATEDIR Constant which represents the template directory of Caramel CMS installation

index.php

- **Package** admin

require_once **BASDIR.'/inc/controller/BackendController.php'** [*line 28*]

Package default Procedural Elements

index.php

- **Package** default

```
BASEDIR = substr($_SERVER["SCRIPT_FILENAME"],0, strrpos($_SERVER["SCRIPT_FILENAME"],"/")) [line 21]
```

- **Var** BASEDIR Constant which represents the basedir of the server

```
require_once BASEDIR.'/inc/controller/FrontendController.php' [line 27]
```

```
TEMPLATEDIR = dirname($_SERVER["SCRIPT_NAME"])."/template" [line 25]
```

- **Var** TEMPLATEDIR Constant which represents the template directory of Caramel CMS installation

Package inc Procedural Elements

BackendController.php

- **Package** inc
- **Sub-Package** controller

require_once **BASEDIR.'/inc/view/TemplateView.php'** [*line 15*]

require_once **BASEDIR.'/inc/model/ConfigurationModel.php'** [*line 14*]

require_once **BASEDIR.'/inc/model/DatabaseModel.php'** [*line 13*]

require_once **BASEDIR.'/inc/utility/CaramelException.php'** [*line 12*]

Imports

FrontendController.php

- **Package** inc
- **Sub-Package** controller

require_once **BASEDIR.'/inc/view/TemplateView.php'** [*line 14*]

require_once **BASEDIR.'/inc/model/ConfigurationModel.php'** [*line 13*]

require_once **BASEDIR.'/inc/model/DatabaseModel.php'** [*line 12*]

require_once **BASEDIR.'/inc/utility/CaramelException.php'** [*line 11*]

Imports

Package inc Classes

Class BackendController

[line 29]

BackendController class

- **Package** inc
- **Sub-Package** controller
- **Author** Felix Rupp < kontakt@felixrupp.com >
- **Version** \$Id\$
- **Copyright** Copyright (c) 2011, Felix Rupp, Nicole Reinhardt
- **License** [MIT-License](#)

BackendController::VERSION

= "0.1.5" *[line 37]*

BackendController::VERSION_DATE

= "2012-01-14" *[line 38]*

Constructor *void* function BackendController::BackendController() *[line 44]*

Constructor

- **Access** public

Content function BackendController::ajaxAction(\$actionName) *[line 286]*

Function Parameters:

- *String* **\$actionName** Name of the ajax action to perform

Method to make ajax requests

- **Deprecated**
- **Access** public

void function BackendController::backendOutputAction() [*line 67*]

This method assigns needed content to our template engine and renders the template.

- **Access** public

string function BackendController::getBaseUrl() [*line 465*]

Print out base url in index.php

- **Access** protected

void function BackendController::getGlobalConfig() [*line 495*]

- **Access** protected

string function BackendController::getMeta() [*line 482*]

Print out meta-tags in index.php

- **Access** protected

string function BackendController::getParametersBefore() [*line 386*]

Get parameters of GET-query before ampersand

- **Access** protected

string function BackendController::getParametersBehind() [*line 418*]

Get parameters of GET-query behind ampersand

- **Access** protected

TRUE function BackendController::getSession() [*line 369*]

Check if session is active or not

- **Access** protected

string function BackendController::headTagAction() [*line 329*]

Print out head-tag in index.php

- **Access** public

void function BackendController::logoutAction() [*line 350*]

This action logs the user off

- **Access** public

void function BackendController::sessionAction() [*line 300*]

Method to initialize login session

- **Access** public

string function BackendController::versionInformationAction() [*line 314*]

Print out version-information in index.php

- **Access** public

Class FrontendController

[*line 28*]

FrontendController class

- **Package** inc
- **Sub-Package** controller
- **Author** Felix Rupp < kontakt@felixrupp.com >
- **Version** \$Id\$
- **Copyright** Copyright (c) 2011, Felix Rupp, Nicole Reinhardt
- **License** [MIT-License](#)

FrontendController::VERSION

= "0.1.5" [*line 36*]

FrontendController::VERSION_DATE

= "2012-01-14" [*line 37*]

Constructor *void* function FrontendController::FrontendController() [*line 43*]

Constructor

- **Access** public

void function FrontendController::frontendOutputAction() [*line 65*]

This method assigns needed content to our template engine and renders the template.

- **Access** public

string function FrontendController::getBaseUrl() [*line 522*]

Print out base url in index.php

- **Access** protected

string function FrontendController::getDisplay() [*line 425*]

Get display from GET-query

- **Access** protected

string function FrontendController::getFooter() [*line 377*]

Print out footer in footer of index.php

- **Access** protected

string function FrontendController::getJsAndCss() [*line 542*]

Return minified JS and CSS files

- **Deprecated**
- **Access** protected

string function FrontendController::getLanguage() [*line 406*]

Extract language from GET-query

- **Access** protected

string function FrontendController::getLanguageSelector() [*line 332*]

Print out language selector in index.php

- **Access** protected

Array function FrontendController::getNavigationLinks(\$navigationArray) [*line 197*]

Function Parameters:

- `array $navigationArray` Array with detailed navigation information

This method build our navigation links from given information **Note: Navigation is restricted to one sublevel**

- **Access** protected

string function FrontendController::getParametersBefore() [*line 443*]

Get parameters of GET-query before ampersand

- **Access** protected

string function FrontendController::getParametersBehind() [*line 474*]

Get parameters of GET-query behind ampersand

- **Access** protected

string function FrontendController::getSocialbar() [*line 557*]

Print out facebook like button in index.php

- **Access** protected

string function FrontendController::headTagAction() [*line 160*]

Print out head-tag in index.php

- **Access** public

string function FrontendController::languageCodeAction() [*line 148*]
Print out current language code in index.php

- **Access** public

void function FrontendController::languageRedirectAction() [*line 92*]
Redirects the user to the language set in browser

- **Access** public

string function FrontendController::versionInformationAction() [*line 134*]
Print out version-information in index.php

- **Access** public

Class ConfigurationModel

[*line 20*]

ConfigurationModel class

- **Package** inc
- **Sub-Package** model
- **Author** Felix Rupp < kontakt@felixrupp.com>
- **Version** \$Id\$
- **Copyright** Copyright (c) 2011, Felix Rupp, Nicole Reinhardt
- **License** [MIT-License](#)

Array function ConfigurationModel::getAdminAction() [*line 147*]

This method returns an ordered array with all admin settings

- **Access** public

string function ConfigurationModel::getAdminConfigStringAction(\$key) [*line 259*]

Function Parameters:

- *string* **\$key** Key to lookup in config-file

Get String from key of admin configfile

- **Access** public

string function ConfigurationModel::getConfigStringAction(\$key) [*line 209*]

Function Parameters:

- *string* **\$key** Key to lookup in config-file

Get String from key of config-file

- **Throws** CaramelException
- **Access** public

ConfigurationModel function ConfigurationModel::getConfigurationModel() [*line 63*]

Singleton-create-method

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

Array function ConfigurationModel::getGlobalsAction() [*line 89*]

This method returns an ordered array with all global settings

- **Access** public

void function ConfigurationModel::reloadAdminConfigFile() [*line 328*]

This method updates the contents of the local adminConfigFile in our singleton.

- **Throws** CaramelException
- **Access** protected

void function ConfigurationModel::reloadConfigFile() [*line 309*]

This method updates the contents of the local configFile in our singleton.

- **Throws** CaramelException
- **Access** protected

void function ConfigurationModel::setAdminAction(\$admin, \$globals) [line 170]

Function Parameters:

- *array* **\$globals** Array that contains all admin settings
throws CaramelException
- **\$admin**

Method to save admin config

- **Access** public

void function ConfigurationModel::setAdminConfigString(\$key, \$newValue) [line 283]

Function Parameters:

- *string* **\$key** Key of the setting to change
- *string* **\$newValue** New value for the setting

Set String from key of admin config-file

- **Throws** CaramelException
- **Access** protected

void function ConfigurationModel::setConfigString(\$key, \$newValue) [line 233]

Function Parameters:

- *string* **\$key** Key of the setting to change
- *string* **\$newValue** New value for the setting

Set String from key of config-file

- **Throws** CaramelException
- **Access** protected

void function ConfigurationModel::setGlobalsAction(\$globals) [line 119]

Function Parameters:

- **array \$globals** Array that contains all global settings
throws CaramelException

Method to save global config

- **Access** public

Class DatabaseModel

[line 20]

DatabaseModel class

- **Package** inc
- **Sub-Package** model
- **Author** Felix Rupp < kontakt@felixrupp.com >

- **Version** \$Id\$
- **Copyright** Copyright (c) 2011, Felix Rupp, Nicole Reinhardt
- **License** [MIT-License](#)

Array function DatabaseModel::getAllLanguagesAction() [*line 86*]

Method to get all used languages out of database

- **Throws** CaramelException
- **Access** public

string function DatabaseModel::getAllMetaTagsAction(\$lang, \$pageName) [*line 118*]

Function Parameters:

- *string* **\$lang** Current language used
- *string* **\$pageName** Name of the current page

Method to return all meta information for the given page/language

- **Throws** CaramelException
- **Access** public

Array function DatabaseModel::getAllPageNamesAction() [*line 155*]

Method to return only all page paths

- **Access** public

DatabaseModel function DatabaseModel::getDatabaseModel() [*line 60*]

Singleton-create-method

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

string function DatabaseModel::getWebsiteContentAction(\$lang, \$pageName) [*line 214*]

Function Parameters:

- *string* **\$lang** Current language used
- *string* **\$pageName** Name of the current page

Method to return the content of a page

- **Throws** CaramelException
- **Access** public

Array function DatabaseModel::getWebsiteNavigationAction(\$lang) [*line 239*]

Function Parameters:

- *string* **\$lang** Current language

Method to return array with localized navigation **Note:** Navigation is restricted to one sublevel

- **Access** public

string function DatabaseModel::getWebsiteTitleAction(\$lang, \$pageName) [*line 188*]

Function Parameters:

- *string* **\$lang** Current language used

- *string* **\$pageName** Name of the current page

Method to return the needed website title

- **Throws** CaramelException
- **Access** public

Class CaramelException

[line 20]

CaramelException class

- **Package** inc
- **Sub-Package** utility
- **Author** Felix Rupp < kontakt@felixrupp.com >
- **Version** \$Id\$
- **Copyright** Copyright (c) 2011, Felix Rupp, Nicole Reinhardt
- **License** [MIT-License](#)

Constructor *void* function CaramelException::__construct(\$caramelCode) *[line 38]*

Function Parameters:

- *int* **\$caramelCode** Errorcode

Constructor for CaramelException

- **Access** public

void function CaramelException::getDetails() [*line 48*]

- **Access** public

Class TemplateView

[*line 20*]

TemplateView class

- **Package** inc
- **Sub-Package** view
- **Author** Felix Rupp < kontakt@felixrupp.com >
- **Version** \$Id\$
- **Copyright** Copyright (c) 2011, Felix Rupp, Nicole Reinhardt
- **License** [MIT-License](#)

Constructor *void* function TemplateView::TemplateView(\$activeTemplate) [*line 45*]

Function Parameters:

- *String* **\$activeTemplate** Path to active template directory

Constructor of TemplateView

- **Access** public

String function `TemplateView::addCssJs()` [*line 77*]

Method to add correct JS and CSS files from template

- **Access** public

void function `TemplateView::assign($key, $value)` [*line 64*]

Function Parameters:

- *String* **\$key** Array-key
- *String* **\$value** Array-value for key

Method to assign all template values to our template

- **Access** public

void function `TemplateView::render()` [*line 107*]

Function which renders the template

- **Access** public

Content function `TemplateView::returnTemplate()` [*line 128*]

Method to return template content as string

- **Access** public

`void function TemplateView::setTemplateFile($templateFile) [line 94]`

Function Parameters:

- *String* **\$templateFile**

Method to set different template filename

- **Access** public

Package template Procedural Elements

index.tpl-alt.php

- **Package** template
- **Sub-Package** Backend

include ["editglobals.tpl.php"](#)*[line 91]*

include ["edittemplates.tpl.php"](#)*[line 88]*

include ["editusers.tpl.php"](#)*[line 85]*

include ["editpages.tpl.php"](#)*[line 82]*

include ["newpage.tpl.php"](#)*[line 79]*

Index template

index.tpl.php

- **Package** template
- **Sub-Package** Backend

include ["editglobals.tpl.php"](#)*[line 93]*

include ["edittemplates.tpl.php"](#)*[line 90]*

include ["editusers.tpl.php"](#)*[line 87]*

include ["editpages.tpl.php"](#)*[line 84]*

include ["newpage.tpl.php"](#)*[line 81]*

Index template

Appendices

Appendix A - Class Trees

Package admin

Package default

Package inc

BackendController

- [BackendController](#)

CaramelException

- Exception
 - [CaramelException](#)

ConfigurationModel

- [ConfigurationModel](#)

DatabaseModel

- [DatabaseModel](#)

FrontendController

- [FrontendController](#)

TemplateView

- [TemplateView](#)

Package template

Index

A

ajaxCommand.php	2
---	---

B

BackendController::getParametersBehind()	11
<i>Get parameters of GET-query behind ampersand</i>	
BackendController::getParametersBefore()	11
<i>Get parameters of GET-query before ampersand</i>	
BackendController::getMeta()	10
<i>Print out meta-tags in index.php</i>	
BackendController::getSession()	11
<i>Check if session is active or not</i>	
BackendController::headTagAction()	11
<i>Print out head-tag in index.php</i>	
BackendController::versionInformationAction()	12
<i>Print out version-information in index.php</i>	
BackendController::sessionAction()	12
<i>Method to initialize login session</i>	
BackendController::logoutAction()	12
<i>This action logs the user off</i>	
BackendController::getGlobalConfig()	10
BackendController::getBaseUrl()	10
<i>Print out base url in index.php</i>	
BackendController	9
<i>BackendController class</i>	
BackendController.php	7
BASEDIR	5
BackendController::VERSION	9
BackendController::VERSION_DATE	9
BackendController::backendOutputAction()	10
<i>This method assigns needed content to our template engine and renders the template.</i>	
BackendController::ajaxAction()	9
<i>Method to make ajax requests</i>	
BASEDIR	2

C

ConfigurationModel::setConfigString()	19
<i>Set String from key of config-file</i>	
ConfigurationModel::setAdminConfigString()	19
<i>Set String from key of admin config-file</i>	
ConfigurationModel::setAdminAction()	19

<i>Method to save admin config</i>	20
ConfigurationModel::setGlobalsAction()	20
<i>Method to save global config</i>	
CaramelException	23
<i>CaramelException class</i>	
constructor TemplateView::TemplateView()	24
<i>Constructor of TemplateView</i>	
CaramelException::getDetails()	24
constructor CaramelException::construct()	23
<i>Constructor for CaramelException</i>	
ConfigurationModel::reloadConfigFile()	18
<i>This method updates the contents of the local configFile in our singleton.</i>	
ConfigurationModel::reloadAdminConfigFile()	18
<i>This method updates the contents of the local adminConfigFile in our singleton.</i>	
ConfigurationModel::getAdminAction()	17
<i>This method returns an ordered array with all admin settings</i>	
ConfigurationModel	16
<i>ConfigurationModel class</i>	
constructor FrontendController::FrontendController()	13
<i>Constructor</i>	
ConfigurationModel::getAdminConfigStringAction()	17
<i>Get String from key of admin configfile</i>	
ConfigurationModel::getConfigStringAction()	17
<i>Get String from key of config-file</i>	
ConfigurationModel::getGlobalsAction()	18
<i>This method returns an ordered array with all global settings</i>	
ConfigurationModel::getConfigurationModel()	18
<i>Singleton-create-method</i>	
constructor BackendController::BackendController()	9
<i>Constructor</i>	

D

DatabaseModel::getWebsiteContentAction()	22
<i>Method to return the content of a page</i>	
DatabaseModel::getWebsiteNavigationAction()	22
<i>Method to return array with localized navigation</i>	
<i>Note: Navigation is restricted to one sublevel</i>	
DatabaseModel::getWebsiteTitleAction()	22
<i>Method to return the needed website title</i>	
DatabaseModel::getDatabaseModel()	21
<i>Singleton-create-method</i>	
DatabaseModel::getAllPageNamesAction()	21
<i>Method to return only all page paths</i>	
DatabaseModel::getAllLanguagesAction()	21
<i>Method to get all used languages out of database</i>	
DatabaseModel::getAllMetaTagsAction()	21
<i>Method to return all meta information for the given page/language</i>	
DatabaseModel	20
<i>DatabaseModel class</i>	

F

FrontendController::getParametersBehind()	15
<i>Get parameters of GET-query behind ampersand</i>	
FrontendController::getParametersBefore()	15
<i>Get parameters of GET-query before ampersand</i>	
FrontendController::getNavigationLinks()	14
<i>This method build our navigation links from given information</i>	
<i>Note: Navigation is restricted to one sublevel</i>	
FrontendController::getSocialbar()	15
<i>Print out facebook like button in index.php</i>	
FrontendController::headTagAction()	15
<i>Print out head-tag in index.php</i>	
FrontendController::versionInformationAction()	16
<i>Print out version-information in index.php</i>	
FrontendController::languageRedirectAction()	16
<i>Redirects the user to the language set in browser</i>	
FrontendController::languageCodeAction()	16
<i>Print out current language code in index.php</i>	
FrontendController::getLanguageSelector()	14
<i>Print out language selector in index.php</i>	
FrontendController::getLanguage()	14
<i>Extract language from GET-query</i>	
FrontendController::VERSION_DATE	13
FrontendController::VERSION	13
FrontendController	12
<i>FrontendController class</i>	
FrontendController::frontendOutputAction()	13
<i>This method assigns needed content to our template engine and renders the template.</i>	
FrontendController::getBaseUrl()	13
<i>Print out base url in index.php</i>	
FrontendController::getJsAndCss()	14
<i>Return minified JS and CSS files</i>	
FrontendController::getFooter()	14
<i>Print out footer in footer of index.php</i>	
FrontendController::getDisplay()	13
<i>Get display from GET-query</i>	
FrontendController.php	8

I

index.tpl.php	29
index.tpl-alt.php	28
index.php	5
index.php	3

T

TemplateView::render()	25
<i>Function which renders the template</i>	
TemplateView::returnTemplate()	25

<i>Method to return template content as string</i>	
TemplateView::setTemplateFile()	26
<i>Method to set different template filename</i>	
TemplateView::assign()	25
<i>Method to assign all template values to our template</i>	
TemplateView::addCssJs()	25
<i>Method to add correct JS and CSS files from template</i>	
TEMPLATEDIR	5
TemplateView	24
<i>TemplateView class</i>	
TEMPLATEDIR	2