symfony API documentation

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Class sfFinder

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Class sfViewCacheManager

symfony API

action

- sfAction: sfAction executes all the logic for the current request.
- sfActions: sfActions executes all the logic for the current request.
- sfActionStack: sfActionStack keeps a list of all requested actions and provides accessor methods for retrieving individual entries.
- sfActionStackEntry: sfActionStackEntry represents information relating to a single sfAction request during a single HTTP request.
- sfComponent: sfComponent.
- sfComponents: sfComponents.

addon

- sfData: This class defines the interface for interacting with data, as well as default implementations.
- sfEzComponentsBridge: This class makes easy to use ez components classes within symfony
- sfMail: sfMail class.
- sfPager: sfPager class.
- sfPropelBehavior:
- sfPropelData: This class is the Propel implementation of sfData. It interacts with the data source and loads data.
- sfPropelDatabaseSchema:
- sfPropelPager: sfPropelPager class.
- sfZendFrameworkBridge: This class makes easy to use Zend Framework classes within symfony

cache

- sfCache: sfCache is an abstract class for all cache classes in symfony.
- sfFileCache: Cache class that stores content in files.
- sfFunctionCache: This class can be used to cache the result and output of functions/methods.
- sfProcessCache: sfProcessCache stores content in memory if you run a PHP accelerator.
- sfSQLiteCache: Cache class that stores content in a sqlite database.

config

- sfAutoloadConfigHandler:
- sfCacheConfigHandler: sfCacheConfigHandler allows you to configure cache.
- sfCompileConfigHandler: sfCompileConfigHandler gathers multiple files and puts them into a single file.
- sfConfig: sfConfig stores all configuration information for a symfony application.
- sfConfigCache: sfConfigCache allows you to customize the format of a configuration file to make it easy-to-use, yet still provide a PHP formatted result for direct inclusion into your modules.
- sfConfigHandler: sfConfigHandler allows a developer to create a custom formatted configuration file pertaining to any information they like and still have it auto-generate PHP code.

- sfDatabaseConfigHandler: sfDatabaseConfigHandler allows you to setup database connections in a configuration file that will be created for you automatically upon first request.
- sfDefineEnvironmentConfigHandler:
- sfFactoryConfigHandler: sfFactoryConfigHandler allows you to specify which factory implementation the system will use.
- sfFilterConfigHandler: sfFilterConfigHandler allows you to register filters with the system.
- sfGeneratorConfigHandler: sfGeneratorConfigHandler.
- sfLoggingConfigHandler: sfLoggingConfigHandler allows you to configure logging and register loggers with the system.
- sfPhpConfigHandler: sfPhpConfigHandler allows you to override php.ini configuration at runtime.
- sfRootConfigHandler: sfRootConfigHandler allows you to specify configuration handlers for the application or on a module level.
- sfRoutingConfigHandler:
- sfSecurityConfigHandler: sfSecurityConfigHandler allows you to configure action security.
- sfSimpleYamlConfigHandler: sfSimpleYamlConfigHandler allows you to load simple configuration files formatted as YAML.
- sfValidatorConfigHandler: sfValidatorConfigHandler allows you to register validators with the system.
- sfViewConfigHandler: sfViewConfigHandler allows you to configure views.
- sfYamlConfigHandler: sfYamlConfigHandler is a base class for YAML (.yml) configuration handlers. This class provides a central location for parsing YAML files and detecting required categories.

controller

- sfConsoleController:
- sfController: sfController directs application flow.
- sfFrontWebController: sfFrontWebController allows you to centralize your entry point in your web application, but at the same time allow for any module and action combination to be requested.
- sfRouting: sfRouting class controls the creation of URLs and parses URLs. It maps an array of parameters to URLs definition.
- sfWebController: sfWebController provides web specific methods to sfController such as, url redirection.

countable

- Countable:

database

- sfDatabase: sfDatabase is a base abstraction class that allows you to setup any type of database connection via a configuration file.
- sfDatabaseManager: sfDatabaseManager allows you to setup your database connectivity before the request is handled. This eliminates the need for a filter to manage database connections.
- sfMySQLDatabase: sfMySQLDatabase provides connectivity for the MySQL brand database.
- sfPDODatabase: sfPDODatabase provides connectivity for the PDO database abstraction layer.

- sfPostgreSQLDatabase: sfPostgreSQLDatabase provides connectivity for the PostgreSQL brand database.
- sfPropelDatabase: A symfony database driver for Propel, derived from the native Creole driver.
- sfPropelDataRetriever: sfGenerator is the abstract base class for all generators.

debug

- sfDebug: sfDebug provides some method to help debugging a symfony application.
- sfWebDebug: sfWebDebug creates debug information for easy debugging in the browser.

exception

- sfActionException: sfActionException is thrown when an error occurs in an action.
- sfAutoloadException: sfAutoloadException is thrown when a class that has been required cannot be loaded.
- sfCacheException: sfCacheException is thrown when ConfigCache fails to execute properly.
- sfConfigurationException: sfConfigurationException is thrown when the framework finds an error in a configuration setting.
- sfContextException: sfContextException is thrown when an instance of the context has not been created.
- sfControllerException: sfControllerException is thrown when a requested Controller implementation doesn&apos;t exist.
- sfDatabaseException: sfDatabaseException is thrown when a database related error occurs.
- sfError404Exception: sfError404Exception is thrown when a 404 error occurs in an action.
- sfException: sfException is the base class for all symfony related exceptions and provides an additional method for printing up a detailed view of an exception.
- sfFactoryException: sfFactoryException is thrown when an error occurs while attempting to create a new factory implementation instance.
- sfFileException: sfFileException is thrown when an error occurs while moving an uploaded file.
- sfFilterException: sfFilterException is thrown when an error occurs while attempting to initialize or execute a filter.
- sfForwardException: sfForwardException is thrown when an error occurs while attempting to forward the request.
- sfInitializationException: sfInitializationException is thrown when an initialization procedure fails.
- sfParseException: sfParseException is thrown when a parsing procedure fails to complete successfully.
- sfRenderException: sfRenderException is thrown when a view&apos;s pre-render check fails.
- sfSecurityException: sfSecurityException is thrown when a security related error occurs.
- sfStopException: sfStopException is thrown when you want to stop action flow.
- sfStorageException: sfStorageException is thrown when a requested sfStorage implementation doesn&apos;t exist or data cannot be read from or written to the storage.
- sfValidatorException: sfValidatorException is thrown when an error occurs in a validator.
- sfViewException: sfViewException is thrown when an error occurs in a view.

filter

- sfBasicSecurityFilter: sfBasicSecurityFilter checks security by calling the getCredential() method of the action. Once the credential has been acquired, sfBasicSecurityFilter verifies the user has the same credential by calling the hasCredential() method of SecurityUser.
- sfCacheFilter: sfCacheFilter deals with page caching and action caching.
- sfCommonFilter: sfCommonFilter automatically adds javascripts and stylesheets information in the sfResponse content.
- sfExecutionFilter: sfExecutionFilter is the last filter registered for each filter chain. This filter does all action and view execution.
- sfFillInFormFilter: sfFillInFormFilter fills in forms.
- sfFilter: sfFilter provides a way for you to intercept incoming requests or outgoing responses.
- sfFilterChain: sfFilterChain manages registered filters for a specific context.
- sfFlashFilter: sfFlashFilter removes flash attributes from the session.
- sfRenderingFilter: sfRenderingFilter is the last filter registered for each filter chain. This filter does the rendering.
- sfSecurityFilter: sfSecurityFilter provides a base class that classifies a filter as one that handles security.
- sfWebDebugFilter:

generator

- sfAdminColumn: Propel admin generator column.
- sfAdminGenerator: Admin generator.
- sfCrudGenerator: CRUD generator.
- sfGenerator: sfGenerator is the abstract base class for all generators.
- sfGeneratorManager: sfGeneratorManager helps generate classes, views and templates for scaffolding, admin interface, ...
- sfPropelAdminGenerator: Propel Admin generator.
- sfPropelCrudGenerator: Propel CRUD generator.

helper

- AssetHelper: AssetHelper.
- CacheHelper: CacheHelper.
- DateFormHelper: DateFormHelper.
- DateHelper: DateHelper.
- EscapingHelper: The functions are primarily used by the output escaping component.
- FormHelper: FormHelper.
- HelperHelper: HelperHelper.
- I18NHelper: I18NHelper.
- JavascriptHelper: JavascriptHelper.
- NumberHelper: NumberHelper.
- ObjectAdminHelper: ObjectHelper for admin generator.
- ObjectHelper: ObjectHelper.
- PartialHelper: PartialHelper.
- sfRichTextEditor: sfRichTextEditor is an abstract class for rich text editor classes.

- sfRichTextEditorFCK: sfRichTextEditorFCK implements the FCK rich text editor.
- sfRichTextEditorTinyMCE: sfRichTextEditorTinyMCE implements the TinyMCE rich text editor.
- TagHelper: TagHelper defines some base helpers to construct html tags.
- TextHelper:
- UrlHelper: UrlHelper.
- ValidationHelper: ValidationHelper.

i18n

- sfI18N:

log

- sfFileLogger:
- sfLogger: sfLogger manages all logging in symfony projects.
- sfLogManager: Log manager
- sfWebDebugLogger:

request

- sfConsoleRequest:
- sfRequest: sfRequest provides methods for manipulating client request information such as attributes, errors and parameters. It is also possible to manipulate the request method originally sent by the user.
- sfWebRequest: sfWebRequest class.

response

- sfConsoleResponse: sfConsoleResponse provides methods for manipulating client response in cli environment.
- sfResponse: sfResponse provides methods for manipulating client response information such as headers, cookies and content.
- sfWebResponse: sfWebResponse class.

sfcore

- sfCore: core symfony class.

storage

- sfMySQLSessionStorage: Provides support for session storage using a MySQL brand database.
- sfPDOSessionStorage: Provides support for session storage using a PDO database abstraction layer.
- sfPostgreSQLSessionStorage: Provides support for session storage using a PostgreSQL brand database.
- sfSessionStorage: sfSessionStorage allows you to store persistent symfony data in the user session.

- sfSessionTestStorage: sfSessionTestStorage is a fake sfSessionStorage implementation to allow easy testing.
- sfStorage: sfStorage allows you to customize the way symfony stores its persistent data.

test

- sfTestBrowser: sfTestBrowser simulates a fake browser which can test a symfony application.

user

- sfBasicSecurityUser: sfBasicSecurityUser will handle any type of data as a credential.
- sfSecurityUser: sfSecurityUser interface provides advanced security manipulation methods.
- sfUser: sfUser wraps a client session and provides accessor methods for user attributes. It also makes storing and retrieving multiple page form data rather easy by allowing user attributes to be stored in namespaces, which help organize data.

util

- sfBrowser: sfBrowser simulates a fake browser which can surf a symfony application.
- sfContext: sfContext provides information about the current application context, such as the module and action names and the module directory. References to the current controller, request, and user implementation instances are also provided.
- sfDomCssSelector: sfDomCssSelector allows to navigate a DOM with CSS selector.
- sfFakeRenderingFilter: sfFilter provides a way for you to intercept incoming requests or outgoing responses.
- sfFillInForm:
- sfFinder: Allow to build rules to find files and directories.
- sfGlobToRegex: Match globbing patterns against text.
- sfInflector:
- sfLoader: sfLoader is a class which contains the logic to look for files/classes in symfony.
- sfMixer: sfMixer implements mixins and hooks.
- sfNumberCompare: Numeric comparisons.
- sfParameterHolder: sfParameterHolder provides a base class for managing parameters.
- sfPropelManyToMany:
- sfTimer: sfTimer class allows to time some PHP code.
- sfTimerManager: sfTimerManager is a container for sfTimer objects.
- sfToolkit: sfToolkit provides basic utility methods.
- sfYaml: sfYaml class.

validator

- sfCallbackValidator: sfCallbackValidator allows you to use a custom callback function or method to validate the input. The function should return true on valid and false on invalid and should be callable using is_callable().
- sfCompareValidator: sfCompareValidator checks the equality of two different request parameters.
- sfDateValidator: sfDateValidator verifies a parameter is of a date format.

- sfEmailValidator: sfEmailValidator verifies a parameter contains a value that qualifies as an email address.
- sfFileValidator: sfFileValidator allows you to apply constraints to file upload.
- sfHtmlValidator:
- sfNumberValidator: sfNumberValidator verifies a parameter is a number and allows you to apply size constraints.
- sfPropelUniqueValidator: sfPropelUniqueValidator validates that the uniqueness of a column.
- sfRegexValidator: sfRegexValidator allows you to match a value against a regular expression pattern.
- sfStringValidator: sfStringValidator allows you to apply string-related constraints to a parameter.
- sfUrlValidator: sfUrlValidator verifies a parameter contains a value that qualifies as a valid URL.
- sfValidator: sfValidator allows you to apply constraints to user entered parameters.
- sfValidatorManager: sfValidatorManager provides management for request parameters and their associated validators.

view

- sfMailView:
- sfOutputEscaper: Abstract class that provides an interface for escaping of output.
- sfOutputEscaperArrayDecorator: Output escaping decorator class for arrays.
- sfOutputEscaperGetterDecorator: Abstract output escaping decorator class for "getter" objects.
- sfOutputEscaperIteratorDecorator: Output escaping iterator decorator.
- sfOutputEscaperObjectDecorator: Output escaping object decorator that intercepts all method calls and escapes their return values.
- sfPartialView:
- sfPHPView:
- sfView: A view represents the presentation layer of an action. Output can be customized by supplying attributes, which a template can manipulate and display.
- sfViewCacheManager: Cache class to cache the HTML results for actions and templates.

[more]

Class sfAction

sfAction executes all the logic for the current request.

Inheritence

sfAction < sfComponent

Direct sub-classes

- sfActions: sfActions executes all the logic for the current request.

Method summary

	forward($module, $action)	Forwards current action to a new one (without browser redirection).
	forward404($message = &apos;&apos;)	Forwards current action to the default 404 error action.
	forward404If($condition, $message = &apos;&apos;)	Forwards current action to the default 404 error action if the specified condition is true.
	forward404Unless($condition, $message = &apos;&apos;)	Forwards current action to the default 404 error action unless the specified condition is true.
	forwardIf($condition, $module, $action)	If the condition is true, forwards current action to a new one (without browser redirection).
	forwardUnless($condition, $module, $action)	Unless the condition is true, forwards current action to a new one (without browser redirection).
mixed	getCredential()	Gets credentials the user must have to access this action.
string	getDefaultView()	Retrieves the default view to be executed when a given request is not served by this action.
mixed	getLayout()	Gets the name of the alternate layout for this sfAction.
int	getRequestMethods()	Retrieves the request methods on which this action will process validation and execution.

string	getSecurityConfiguration()	Returns the security configuration for this module.
string	getTemplate()	Gets the name of the alternate template for this sfAction.
string	handleError()	Executes any post-validation error application logic.
bool	initialize((sfContext) $context)	Initializes this action.
bool	isSecure()	Indicates that this action requires security.
	postExecute()	Execute an application defined process immediately after execution of this sfAction object.
	preExecute()	Executes an application defined process prior to execution of this sfAction object.
	redirect($url, $statusCode = 302)	Redirects current request to a new URL.
	redirect404()	Redirects current action to the default 404 error action (with browser redirection).
	redirectIf($condition, $url)	Redirects current request to a new URL, only if specified condition is true.
	redirectUnless($condition, $url)	Redirects current request to a new URL, unless specified condition is true.
sfView::NONE	renderText($text)	Appends the given text to the response content and bypasses the built-in view system.
	setLayout($name)	Sets an alternate layout for this sfAction.
	setSecurityConfiguration($security)	Overrides the current security configuration for this module.
	setTemplate($name)	Sets an alternate template for this sfAction.
	setViewClass($class)	Changes the default view class used for rendering the template associated with the current action.
bool	validate()	Validates manually files and parameters.

Method details

- ### forward ($module, $action)

$module	A module name
$action	An action name

Forwards current action to a new one (without browser redirection).

This method stops the action. So, no code is executed after a call to this method.

throws sfStopException

- ### forward404 ($message = &apos;&apos;)

$message	Message of the generated exception

Forwards current action to the default 404 error action.

throws sfError404Exception

- ### forward404If ($condition, $message = &apos;&apos;)

$condition	A condition that evaluates to true or false
$message	Message of the generated exception

Forwards current action to the default 404 error action if the specified condition is true.

throws sfError404Exception

- ### forward404Unless ($condition, $message = &apos;&apos;)

$condition	A condition that evaluates to true or false
$message	Message of the generated exception

Forwards current action to the default 404 error action unless the specified condition is true.

throws sfError404Exception

- ### forwardIf ($condition, $module, $action)

$condition	A condition that evaluates to true or false

$module	A module name
$action	An action name

If the condition is true, forwards current action to a new one (without browser redirection).

This method stops the action. So, no code is executed after a call to this method.

throws sfStopException

-
forwardUnless ($condition, $module, $action)

$condition	A condition that evaluates to true or false
$module	A module name
$action	An action name

Unless the condition is true, forwards current action to a new one (without browser redirection).

This method stops the action. So, no code is executed after a call to this method.

throws sfStopException

-
(mixed) getCredential ()

Gets credentials the user must have to access this action.

returns An array or a string describing the credentials the user must have to access this action

-
(string) getDefaultView ()

Retrieves the default view to be executed when a given request is not served by this action.

returns A string containing the view name associated with this action

-
(mixed) getLayout ()

Gets the name of the alternate layout for this sfAction.

WARNING: It only returns the layout you set with the setLayout() method, and does not return the layout that you configured in your view.yml.

returns Layout name. Returns null if no layout has been set within the action

-
(int) getRequestMethods ()

Retrieves the request methods on which this action will process validation and execution.

returns

One of the following values:

  - ♦ sfRequest::GET

    - sfRequest::POST
    - sfRequest::PUT
    - sfRequest::DELETE
    - sfRequest::HEAD
    - sfRequest::NONE

(string) getSecurityConfiguration ()

Returns the security configuration for this module.

returns Current security configuration as an array

(string) getTemplate ()

Gets the name of the alternate template for this sfAction.

WARNING: It only returns the template you set with the setTemplate() method, and does not return the template that you configured in your view.yml.

See &apos;Naming Conventions&apos; in the &apos;Symfony View&apos; documentation.

returns Template name. Returns null if no template has been set within the action

(string) handleError ()

Executes any post-validation error application logic.

returns A string containing the view name associated with this action

(bool) initialize ((sfContext) $context)

$context	The current application context.

Initializes this action.

returns true, if initialization completes successfully, otherwise false

(bool) isSecure ()

Indicates that this action requires security.

returns true, if this action requires security, otherwise false.

postExecute ()

Execute an application defined process immediately after execution of this sfAction object.

By default, this method is empty.

- ## preExecute ()

  Executes an application defined process prior to execution of this sfAction object.

  By default, this method is empty.

- ## redirect ($url, $statusCode = 302)

$url	Url
$statusCode	Status code (default to 302)

  Redirects current request to a new URL.

  2 URL formats are accepted :

  - a full URL: http://www.google.com/
  - an internal URL (url_for() format): module/action

  This method stops the action. So, no code is executed after a call to this method.

  throws sfStopException

- ## redirect404 ()

  Redirects current action to the default 404 error action (with browser redirection).

  This method stops the current code flow.

- ## redirectIf ($condition, $url)

$condition	A condition that evaluates to true or false
$url	url

  Redirects current request to a new URL, only if specified condition is true.

  This method stops the action. So, no code is executed after a call to this method.

  throws sfStopException

- ## redirectUnless ($condition, $url)

$condition	A condition that evaluates to true or false
$url	Url

  Redirects current request to a new URL, unless specified condition is true.

  This method stops the action. So, no code is executed after a call to this method.

  throws sfStopException

- ## (sfView::NONE) renderText ($text)

$text	Text to append to the response

  Appends the given text to the response content and bypasses the built-in view system.

  This method must be called as with a return:

  ```
  renderText('some text')]]>
  ```

- ## setLayout ($name)

  $name Layout name or false to de-activate the layout

  Sets an alternate layout for this sfAction.

  To de-activate the layout, set the layout name to false.

  To revert the layout to the one configured in the view.yml, set the template name to null.

- ## setSecurityConfiguration ($security)

  $security The new security configuration

  Overrides the current security configuration for this module.

- ## setTemplate ($name)

  $name Template name

  Sets an alternate template for this sfAction.

  See &apos;Naming Conventions&apos; in the &apos;Symfony View&apos; documentation.

- ## setViewClass ($class)

  $class View class name

  Changes the default view class used for rendering the template associated with the current action.

- ## (bool) validate ()

  Validates manually files and parameters.

  returns true, if validation completes successfully, otherwise false.

[more]

Class sfActions

sfActions executes all the logic for the current request.

Inheritence

sfActions < sfComponent < sfAction

Method summary

string	execute()	Dispatches to the action defined by the &apos;action&apos; parameter of the sfRequest object.

Methods inherited from sfAction

forward, forward404, forward404If, forward404Unless, forwardIf, forwardUnless, getCredential, getDefaultView, getLayout, getRequestMethods, getSecurityConfiguration, getTemplate, handleError, initialize, isSecure, postExecute, preExecute, redirect, redirect404, redirectIf, redirectUnless, renderText, setLayout, setSecurityConfiguration, setTemplate, setViewClass, validate

Methods inherited from sfComponent

__call, __get, __isset, __set, __unset, debugMessage, execute, getActionName, getContext, getController, getFlash, getLogger, getModuleName, getPresentationFor, getRequest, getRequestParameter, getResponse, getUser, getVar, getVarHolder, hasFlash, hasRequestParameter, initialize, logMessage, sendEmail, setFlash, setVar

Method details

- ### (string) execute ()

  Dispatches to the action defined by the &apos;action&apos; parameter of the sfRequest object.

  This method try to execute the executeXXX() method of the current object where XXX is the defined action name.

  returns A string containing the view name associated with this action

  throws sfInitializationException

[more]

Class sfActionStack

sfActionStack keeps a list of all requested actions and provides accessor methods for retrieving individual entries.

Method summary

	addEntry($moduleName, $actionName, (sfAction) $actionInstance)	Adds an entry to the action stack.
	getEntry($index)	Retrieves the entry at a specific index.
mixed	getFirstEntry()	Retrieves the first entry.
mixed	getLastEntry()	Retrieves the last entry.
int	getSize()	Retrieves the size of this stack.
	popEntry(0)	Removes the entry at a specific index.

Method details

- ### addEntry ($moduleName, $actionName, (sfAction) $actionInstance)

$moduleName	A module name
$actionName	An action name
$actionInstance	An sfAction implementation instance

Adds an entry to the action stack.

returns sfActionStackEntry instance

- ### getEntry ($index)

$index	An entry index

Retrieves the entry at a specific index.

returns An action stack entry implementation.

- ### (mixed) getFirstEntry ()

Retrieves the first entry.

returns An action stack entry implementation or null if there is no sfAction instance in the stack

- ### (mixed) getLastEntry ()
  Retrieves the last entry.

  returns An action stack entry implementation or null if there is no sfAction instance in the stack
- ### (int) getSize ()
  Retrieves the size of this stack.

  returns The size of this stack.
- ### popEntry (0)

0	An entry index

  Removes the entry at a specific index.

  returns An action stack entry implementation.

[more]

Class sfActionStackEntry

sfActionStackEntry represents information relating to a single sfAction request during a single HTTP request.

Method summary

	__construct($moduleName, $actionName, (sfAction) $actionInstance)	Class constructor.
	getActionInstance()	Retrieves this entry&apos;s action instance.
string	getActionName()	Retrieves this entry&apos;s action name.
string	getModuleName()	Retrieves this entry&apos;s module name.
string&	getPresentation()	Retrieves this entry&apos;s rendered view presentation.
	getViewInstance()	Retrieves this entry&apos;s view instance.
	setPresentation([[[amp]]]$presentation)	Sets the rendered presentation for this action.
	setViewInstance((sfView) $viewInstance)	Sets this entry&apos;s view instance.

Method details

- ### __construct ($moduleName, $actionName, (sfAction) $actionInstance)

$moduleName	A module name
$actionName	An action name
$actionInstance	An sfAction implementation instance

  Class constructor.

- ### getActionInstance ()

  Retrieves this entry&apos;s action instance.

  returns An sfAction implementation instance

- ### (string) getActionName ()

  Retrieves this entry&apos;s action name.

  returns An action name

- ### (string) getModuleName ()

  Retrieves this entry&apos;s module name.

  returns A module name
- ### (string&) getPresentation ()

  Retrieves this entry&apos;s rendered view presentation.

  This will only exist if the view has processed and the render mode is set to sfView::RENDER_VAR.

  returns Rendered view presentation
- ### getViewInstance ()

  Retrieves this entry&apos;s view instance.

  returns A sfView implementation instance.
- ### setPresentation ([[[amp]]]$presentation)

[[[amp]]]$presentation	A rendered presentation.

  Sets the rendered presentation for this action.
- ### setViewInstance ((sfView) $viewInstance)

$viewInstance	A sfView implementation instance.

  Sets this entry&apos;s view instance.

[more]

Class sfComponent

sfComponent.

Direct sub-classes

- sfComponents: sfComponents.
- sfAction: sfAction executes all the logic for the current request.

Method summary

mixed	__call($method, $arguments)	Calls methods defined via the sfMixer class.
mixed&	__get($key)	Gets a variable for the template.
boolean	__isset($name)	Returns true if a variable for the template is set.
boolean	__set($key, $value)	Sets a variable for the template.
	__unset($name)	Removes a variable for the template.
	debugMessage($message)	Displays a message as a short message in the sfWebDebug toolbar.
mixed	execute()	Execute any application/business logic for this component.
string	getActionName()	Gets the action name associated with this component.
	getContext()	Retrieves the current application context.
	getController()	Retrieves the current sfController object.
mixed	getFlash($name)	Gets a flash variable.
	getLogger()	Retrieves the current logger instance.
string	getModuleName()	Gets the module name associated with this component.
string	getPresentationFor($module, $action, $viewName = null)	Returns the rendered view presentation of a given module/action.
	getRequest()	Retrieves the current sfRequest object.
string	getRequestParameter($name, $default = null)	Returns the value of a request parameter.
	getResponse()	Retrieves the current sfResponse object.
	getUser()	Retrieves the current sfUser object.
mixed	getVar($name)	Gets a variable set for the template.

	getVarHolder()	Gets the sfParameterHolder object that stores the template variables.
boolean	hasFlash($name)	Returns true if a flash variable of the specified name exists.
boolean	hasRequestParameter($name)	Returns true if a request parameter exists.
boolean	initialize((sfContext) $context)	Initializes this component.
	logMessage($message, $priority = &apos;info&apos;)	Logs a message using the sfLogger object.
string	sendEmail($module, $action)	Sends and email from the current action.
	setFlash($name, $value, $persist = true)	Sets a flash variable that will be passed to the very next action.
	setVar($name, $value)	Sets a variable for the template.

Method details

- ### (mixed) __call ($method, $arguments)

$method	The method name
$arguments	The method arguments

Calls methods defined via the sfMixer class.

returns The returned value of the called method

- ### (mixed&) __get ($key)

$key	The variable name

Gets a variable for the template.

This is a shortcut for:

```
getVar('name')]]>
```

returns The variable value

- ### (boolean) __isset ($name)

$name	The variable name

Returns true if a variable for the template is set.

This is a shortcut for:

```
getVarHolder()->has('name')]]>
```

returns true if the variable is set

- ## (boolean) __set ($key, $value)

$key	The variable name
$value	The variable value

Sets a variable for the template.

This is a shortcut for:

```
setVar('name', 'value')]]>
```

returns always true

- ## __unset ($name)

$name	The variable Name

Removes a variable for the template.

This is just really a shortcut for:

```
getVarHolder()->remove('name')]]>
```

- ## debugMessage ($message)

$message	The message text

Displays a message as a short message in the sfWebDebug toolbar.

- ## (mixed) execute ()

Execute any application/business logic for this component.

In a typical database-driven application, execute() handles application logic itself and then proceeds to create a model instance. Once the model instance is initialized it handles all business logic for the action.

A model should represent an entity in your application. This could be a user account, a shopping cart, or even a something as simple as a single product.

returns A string containing the view name associated with this action

- ## (string) getActionName ()

Gets the action name associated with this component.

returns An action name

- ## getContext ()

  Retrieves the current application context.

  returns The current sfContext instance

- ## getController ()

  Retrieves the current sfController object.

  This is a proxy method equivalent to:

  ```
  getContext()->getController()]]>
  ```

  returns The current sfController implementation instance

- ## (mixed) getFlash ($name)

$name	The name of the flash variable

  Gets a flash variable.

  returns The value of the flash variable

- ## getLogger ()

  Retrieves the current logger instance.

  returns The current sfLogger instance

- ## (string) getModuleName ()

  Gets the module name associated with this component.

  returns A module name

- ## (string) getPresentationFor ($module, $action, $viewName = null)

$module	A module name
$action	An action name
$viewName	A View class name

  Returns the rendered view presentation of a given module/action.

  This is a shortcut for

  ```
  getController()->getPresentationFor($module, $action, $viewName)]]>
  ```

  returns The generated content

- ## getRequest ()

  Retrieves the current sfRequest object.

  This is a proxy method equivalent to:

  ```
  getContext()->getRequest()]]>
  ```

  returns The current sfRequest implementation instance

- ## (string) getRequestParameter ($name, $default = null)

  $name The parameter name

  Returns the value of a request parameter.

  This is a proxy method equivalent to:

  ```
  getRequest()->getParameterHolder()->get($name)]]>
  ```

  returns The request parameter value

- ## getResponse ()

  Retrieves the current sfResponse object.

  This is a proxy method equivalent to:

  ```
  getContext()->getResponse()]]>
  ```

  returns The current sfResponse implementation instance

- ## getUser ()

  Retrieves the current sfUser object.

  This is a proxy method equivalent to:

  ```
  getContext()->getController()]]>
  ```

  returns The current sfUser implementation instance

- ## (mixed) getVar ($name)

  $name The variable name

  Gets a variable set for the template.

  returns The variable value

- ## getVarHolder ()

  Gets the sfParameterHolder object that stores the template variables.

  returns The variable holder.

- ## (boolean) hasFlash ($name)

  $name The name of the flash variable

  Returns true if a flash variable of the specified name exists.

  returns true if the variable exists, false otherwise

- ## (boolean) hasRequestParameter ($name)

  $name The parameter name

  Returns true if a request parameter exists.

  This is a proxy method equivalent to:

  ```
  getRequest()->getParameterHolder()->has($name)]]>
  ```

  returns true if the request parameter exists, false otherwise

- ## (boolean) initialize ((sfContext) $context)

$context	The current application context

  Initializes this component.

  returns true, if initialization completes successfully, otherwise false

- ## logMessage ($message, $priority = &apos;info&apos;)

$message	String or object containing the message to log
$priority	The priority of the message (available priorities: emerg, alert, crit, err, warning, notice, info, debug)

  Logs a message using the sfLogger object.

- ## (string) sendEmail ($module, $action)

$module	A module name
$action	An action name

  Sends and email from the current action.

This methods calls a module/action with the sfMailView class.

This is a shortcut for

```
getController()->sendEmail($module, $action)]]>
```

returns The generated mail content

- ## setFlash ($name, $value, $persist = true)

$name	The name of the flash variable
$value	The value of the flash variable
$persist	true if the flash have to persist for the following request (true by default)

  Sets a flash variable that will be passed to the very next action.

- ## setVar ($name, $value)

$name	The variable name
$value	The variable value

  Sets a variable for the template.

[more]

Class sfComponents

sfComponents.

Inheritence

sfComponents < sfComponent

Method summary

execute()

Methods inherited from sfComponent

__call, __get, __isset, __set, __unset, debugMessage, execute, getActionName, getContext, getController, getFlash, getLogger, getModuleName, getPresentationFor, getRequest, getRequestParameter, getResponse, getUser, getVar, getVarHolder, hasFlash, hasRequestParameter, initialize, logMessage, sendEmail, setFlash, setVar

Method details

- ### execute ()

[more]

Class sfData

This class defines the interface for interacting with data, as well as default implementations.

Direct sub-classes

- sfPropelData: This class is the Propel implementation of sfData. It interacts with the data source and loads data.

Method summary

	doLoadData($fixture_files)	Manages reading all of the fixture data files and loading them into the data source
	doLoadDataFromFile($fixture_file)	Loads data for the database from a YAML file
boolean	getDeleteCurrentData()	Gets the current value of the flag that indicates whether current data is to be deleted or not.
array	getFiles($directory_or_file = null)	Gets a list of one or more *.yml files and returns the list in an array
	loadDataFromArray($data)	Manages the insertion of data into the data source
	setDeleteCurrentData($boolean)	Sets a flag to indicate if the current data in the database should be deleted before new data is loaded.

Method details

- ### doLoadData ($fixture_files)

$fixture_files	The path names of the YAML data files

  Manages reading all of the fixture data files and loading them into the data source

- ### doLoadDataFromFile ($fixture_file)

$fixture_file	The path to the YAML file.

  Loads data for the database from a YAML file

- ### (boolean) getDeleteCurrentData ()

  Gets the current value of the flag that indicates whether current data is to be deleted or not.

- ## (array) getFiles ($directory_or_file = null)

$directory_or_file	A directory or file name; if null, then defaults to &apos;sf_data_dir&apos;/fixtures

  Gets a list of one or more *.yml files and returns the list in an array

  returns A list of *.yml files.

  throws sfInitializationException If the directory or file does not exist.

- ## loadDataFromArray ($data)

$data	The data to be inserted into the data source

  Manages the insertion of data into the data source

- ## setDeleteCurrentData ($boolean)

$boolean	The flag value

  Sets a flag to indicate if the current data in the database should be deleted before new data is loaded.

[more]

Class sfEzComponentsBridge

This class makes easy to use ez components classes within symfony

Method summary

autoload($class)

Method details

- ### autoload ($class)

[more]

Class sfMail

sfMail class.

Method summary

__construct()

addAddress($address, $name = null)

addAddresses($addresses)

addAttachment($path, $name = &apos;&apos;, $encoding = &apos;base64&apos;, $type = &apos;application/octet-s

addBcc($address, $name = null)

addCc($address, $name = null)

addCustomHeader($name, $value)

addEmbeddedImage($path, $cid, $name = &apos;&apos;, $encoding = &apos;base64&apos;, $type = &apos;applicat

addReplyTo($address, $name = null)

addStringAttachment($string, $filename, $encoding = &apos;base64&apos;, $type = &apos;application/octet-stream&

clearAddresses()

clearAllRecipients()

clearAttachments()

clearBccs()

clearCcs()

clearCustomHeaders()

clearReplyTos()

getAltBody()

getBody()

getCharset()

getContentType()

getDomain()

getEncoding()

getFrom()

getHostname()

getMailer()

getPassword()

getPort()

getPriority()

getRawBody()

getRawHeader()

getSender()

getSubject()
getUsername()
getWordWrap()
initialize()
prepare()
send()
setAltBody($text)
setAttachments($attachments)
setBody($body)
setCharset($charset)
setContentType($content_type)
setDomain($hostname)
setEncoding($encoding)
setFrom($address, $name = null)
setHostname($hostname)
setMailer($type = &apos;mail&apos;, $options = array())
setPassword($password)
setPort($port)
setPriority($priority)
setSender($address, $name = null)
setSubject($subject)
setUsername($username)
setWordWrap($wordWrap)
smtpClose()

Method details

- ### __construct ()
- ### addAddress ($address, $name = null)
- ### addAddresses ($addresses)
- ### addAttachment ($path, $name = &apos;&apos;, $encoding = &apos;base64&apos;, $type = &apos;application/octet-stream&apos;)

- **addBcc ($address, $name = null)**
- **addCc ($address, $name = null)**
- **addCustomHeader ($name, $value)**
- **addEmbeddedImage ($path, $cid, $name = &apos;&apos;, $encoding = &apos;base64&apos;, $type = &apos;application/octet-stream&apos;)**
- **addReplyTo ($address, $name = null)**
- **addStringAttachment ($string, $filename, $encoding = &apos;base64&apos;, $type = &apos;application/octet-stream&apos;)**
- **clearAddresses ()**
- **clearAllRecipients ()**
- **clearAttachments ()**
- **clearBccs ()**
- **clearCcs ()**
- **clearCustomHeaders ()**
- **clearReplyTos ()**
- **getAltBody ()**
- **getBody ()**
- **getCharset ()**
- **getContentType ()**
- **getDomain ()**

- **getEncoding ()**
- **getFrom ()**
- **getHostname ()**
- **getMailer ()**
- **getPassword ()**
- **getPort ()**
- **getPriority ()**
- **getRawBody ()**
- **getRawHeader ()**
- **getSender ()**
- **getSubject ()**
- **getUsername ()**
- **getWordWrap ()**
- **initialize ()**
- **prepare ()**
- **send ()**
- **setAltBody ($text)**
- **setAttachments ($attachments)**
- **setBody ($body)**
- **setCharset ($charset)**

- **setContentType ($content_type)**
- **setDomain ($hostname)**
- **setEncoding ($encoding)**
- **setFrom ($address, $name = null)**
- **setHostname ($hostname)**
- **setMailer ($type = &apos;mail&apos;, $options = array())**
- **setPassword ($password)**
- **setPort ($port)**
- **setPriority ($priority)**
- **setSender ($address, $name = null)**
- **setSubject ($subject)**
- **setUsername ($username)**
- **setWordWrap ($wordWrap)**
- **smtpClose ()**

[more]

Class sfPager

sfPager class.

Direct sub-classes

- sfPropelPager: sfPropelPager class.

Method summary

__construct($class, $maxPerPage = 10)

getClass()

getCriteria()

getCurrent()

getCurrentMaxLink()

getCursor()

getFirstIndice()

getFirstPage()

getLastIndice()

getLastPage()

getLinks($nb_links = 5)

getMaxPerPage()

getMaxRecordLimit()

getNbResults()

getNext()

getNextPage()

getObjectByCursor($pos)

getPage()

getParameter($name, $default = null, $ns = null)

getParameterHolder()

getPrevious()

getPreviousPage()

getResults()

hasParameter($name, $ns = null)

haveToPaginate()

init()

retrieveObject($offset)

setClass($class)

setCriteria($c)
setCursor($pos)
setLastPage($page)
setMaxPerPage($max)
setMaxRecordLimit($limit)
setNbResults($nb)
setPage($page)
setParameter($name, $value, $ns = null)

Method details

- ### __construct ($class, $maxPerPage = 10)
- ### getClass ()
- ### getCriteria ()
- ### getCurrent ()
- ### getCurrentMaxLink ()
- ### getCursor ()
- ### getFirstIndice ()
- ### getFirstPage ()
- ### getLastIndice ()
- ### getLastPage ()
- ### getLinks ($nb_links = 5)
- ### getMaxPerPage ()
- ### getMaxRecordLimit ()
- ### getNbResults ()

- **getNext ()**
- **getNextPage ()**
- **getObjectByCursor ($pos)**
- **getPage ()**
- **getParameter ($name, $default = null, $ns = null)**
- **getParameterHolder ()**
- **getPrevious ()**
- **getPreviousPage ()**
- **getResults ()**
- **hasParameter ($name, $ns = null)**
- **haveToPaginate ()**
- **init ()**
- **retrieveObject ($offset)**
- **setClass ($class)**
- **setCriteria ($c)**
- **setCursor ($pos)**
- **setLastPage ($page)**
- **setMaxPerPage ($max)**
- **setMaxRecordLimit ($limit)**
- **setNbResults ($nb)**

- **setPage ($page)**
- **setParameter ($name, $value, $ns = null)**

[more]

Class sfPropelBehavior

Method summary

add($class, $behaviors)
registerHooks($name, $hooks)
registerMethods($name, $callables)

Method details

- ### add ($class, $behaviors)
- ### registerHooks ($name, $hooks)
- ### registerMethods ($name, $callables)

[more]

Class sfPropelData

This class is the Propel implementation of sfData. It interacts with the data source and loads data.

Inheritence

sfPropelData < sfData

Method summary

doDeleteCurrentData($fixture_files)	Clears from th by read fixture deletin data fo classes mentio fixtures
dumpData($directory_or_file = null, $tables = &apos;all&apos;, $connectionName = &apos;propel&apos;)	Dumps fixture more ta
loadData($directory_or_file = null, $connectionName = &apos;propel&apos;)	Loads file or a Prope
loadDataFromArray($data)	Implen abstrac loadDa methoc the dat generat model.
loadMapBuilder($class)	Loads for the

Methods inherited from sfData

doLoadData, doLoadDataFromFile, getDeleteCurrentData, getFiles, loadDataFromArray, setDeleteCurrentData

Method details

- ## doDeleteCurrentData ($fixture_files)

$fixture_files	The list of YAML files.

Clears existing data from the data source by reading the fixture files and deleting the existing data for only those classes that are mentioned in the fixtures.

throws sfException If a class mentioned in a fixture can not be found

- ## dumpData ($directory_or_file = null, $tables = &apos;all&apos;, $connectionName = &apos;propel&apos;)

$directory_or_file	directory or file to dump to
$tables	name or names of tables to dump
$connectionName	connection name

Dumps data to fixture from 1 or more tables.

- ## loadData ($directory_or_file = null, $connectionName = &apos;propel&apos;)

$directory_or_file	A file or directory path
$connectionName	The Propel connection name, default &apos;propel&apos;

Loads data from a file or directory into a Propel data source

throws Exception If the database throws an error, rollback transaction and rethrows exception

- ## loadDataFromArray ($data)

$data	The data to be loaded into the data source

Implements the abstract loadDataFromArray method and loads the data using the generated data model.

throws Exception If data is unnamed.

- ## loadMapBuilder ($class)

$class	The name of a data object

Loads the mappings for the classes

throws sfException If the class cannot be found

[more]

Class sfPropelDatabaseSchema

Method summary

asArray()
asXML()
asYAML()
findTable($table_name)
fixXML()
fixXMLColumns()
fixXMLForeignKeys()
fixXMLIndexes()
fixYAMLColumns()
fixYAMLDatabase()
fixYAMLI18n()
getAttributesFor($tag)
getAttributesForColumn($tb_name, $col_name, $column)
getAttributesFromCompactType($type)
getChildren($hash)
getCorrectValueFor($key, $value)
getNameAndAttributes($hash, $name_attribute = &apos;name&apos;)
getTables()
loadXML($file)
loadYAML($file)
removeEmptyKey([[[amp]]]$hash, $key)
setIfNotSet([[[amp]]]$entry, $key, $value)

Method details

- ### asArray ()
- ### asXML ()
- ### asYAML ()

- **findTable ($table_name)**
- **fixXML ()**
- **fixXMLColumns ()**
- **fixXMLForeignKeys ()**
- **fixXMLIndexes ()**
- **fixYAMLColumns ()**
- **fixYAMLDatabase ()**
- **fixYAMLI18n ()**
- **getAttributesFor ($tag)**
- **getAttributesForColumn ($tb_name, $col_name, $column)**
- **getAttributesFromCompactType ($type)**
- **getChildren ($hash)**
- **getCorrectValueFor ($key, $value)**
- **getNameAndAttributes ($hash, $name_attribute = &apos;name&apos;)**
- **getTables ()**
- **loadXML ($file)**
- **loadYAML ($file)**
- **removeEmptyKey ([[[amp]]]$hash, $key)**
- **setIfNotSet ([[[amp]]]$entry, $key, $value)**

[more]

Class sfPropelPager

sfPropelPager class.

Inheritence

sfPropelPager < sfPager

Method summary

__construct($class, $maxPerPage = 10)

getClassPeer()

getPeerCountMethod()

getPeerMethod()

getResults()

init()

retrieveObject($offset)

setPeerCountMethod($peer_count_method_name)

setPeerMethod($peer_method_name)

Methods inherited from sfPager

__construct, getClass, getCriteria, getCurrent, getCurrentMaxLink, getCursor, getFirstIndice, getFirstPage, getLastIndice, getLastPage, getLinks, getMaxPerPage, getMaxRecordLimit, getNbResults, getNext, getNextPage, getObjectByCursor, getPage, getParameter, getParameterHolder, getPrevious, getPreviousPage, getResults, hasParameter, haveToPaginate, init, retrieveObject, setClass, setCriteria, setCursor, setLastPage, setMaxPerPage, setMaxRecordLimit, setNbResults, setPage, setParameter

Method details

- ### __construct ($class, $maxPerPage = 10)
- ### getClassPeer ()
- ### getPeerCountMethod ()
- ### getPeerMethod ()
- ### getResults ()

- **init ()**
- **retrieveObject ($offset)**
- **setPeerCountMethod ($peer_count_method_name)**
- **setPeerMethod ($peer_method_name)**

[more]

Class sfZendFrameworkBridge

This class makes easy to use Zend Framework classes within symfony

Method summary

autoload($class)

Method details

- ### autoload ($class)

[more]

Class sfCache

sfCache is an abstract class for all cache classes in symfony.

Direct sub-classes

- sfSQLiteCache: Cache class that stores content in a sqlite database.
- sfFileCache: Cache class that stores content in files.

Method summary

boolean	clean($namespace = null, $mode = &apos;all&apos;)	Cleans the cache.
string	get($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)	Gets the cache content for a given id and namespace.
int	getLifeTime()	Returns the current life time.
boolean	has($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)	Returns true if there is a cache for the given id and namespace.
int	lastModified($id, $namespace = self::DEFAULT_NAMESPACE)	Returns the cache last modification time.
boolean	remove($id, $namespace = self::DEFAULT_NAMESPACE)	Removes a content from the cache.
boolean	set($id, $namespace = self::DEFAULT_NAMESPACE, $data)	Saves some data in the cache.
	setLifeTime($newLifeTime)	Sets a new life time.

Method details

- ### (boolean) clean ($namespace = null, $mode = &apos;all&apos;)

$namespace	The name of the cache namespace

Cleans the cache.

If no namespace is specified all cache content will be destroyed else only cache contents of the specified namespace will be destroyed.

returns true if no problem

- ### (string) get ($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)

$id	The cache id
$namespace	The name of the cache namespace
$doNotTestCacheValidity	If set to true, the cache validity won&apos;t be tested

Gets the cache content for a given id and namespace.

returns The data of the cache (or null if no cache available)

- ### (int) getLifeTime ()

Returns the current life time.

returns The current life time (in seconds)

- ### (boolean) has ($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)

$id	The cache id
$namespace	The name of the cache namespace
$doNotTestCacheValidity	If set to true, the cache validity won&apos;t be tested

Returns true if there is a cache for the given id and namespace.

returns true if the cache exists, false otherwise

- ### (int) lastModified ($id, $namespace = self::DEFAULT_NAMESPACE)

Returns the cache last modification time.

returns The last modification time

- ### (boolean) remove ($id, $namespace = self::DEFAULT_NAMESPACE)

$id	The cache id
$namespace	The name of the cache namespace

Removes a content from the cache.

returns true if no problem

- ## (boolean) set ($id, $namespace = self::DEFAULT_NAMESPACE, $data)

$id	The cache id
$namespace	The name of the cache namespace
$data	The data to put in cache

Saves some data in the cache.

returns true if no problem

- ## setLifeTime ($newLifeTime)

$newLifeTime	The new life time (in seconds)

Sets a new life time.

[more]

Class sfFileCache

Cache class that stores content in files.

This class is based on the PEAR_Cache_Lite class. All cache files are stored in files in the [sf_root_dir].&apos;/cache/&apos;.[sf_app].&apos;/template&apos; directory. To disable all caching, you can set to false [sf_cache] setting.

Inheritence

sfFileCache < sfCache

Direct sub-classes

- sfFunctionCache: This class can be used to cache the result and output of functions/methods.

Method summary

	__construct($cacheDir = null)	Constructor.
boolean	clean($namespace = null, $mode = &apos;all&apos;)	Cleans the cache.
boolean	cleanDir($dir, $mode, 2)	Recursive function for cleaning cache file in the given directory.
string	get($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)	Tests if a cache is available and (if yes) returns it.
	getCacheDir()	
boolean	getFileLocking()	Gets the value of the fileLocking option.
array	getFileName($id, $namespace)	Makes a file name (with path).
boolean	getWriteControl()	Gets the value of the writeControl option.

boolean	has($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)	Returns true if there is a cache for the given id and namespace.
string	hash($data)	Makes a control key with the string containing datas.
	initialize($options = array())	Initializes the cache.
int	lastModified($id, $namespace = self::DEFAULT_NAMESPACE)	Returns the cache last modification time.
string	read($path, $file)	Reads the cache file and returns the content.
boolean	remove($id, $namespace = self::DEFAULT_NAMESPACE)	Removes a cache file.
boolean	set($id, $namespace = self::DEFAULT_NAMESPACE, $data)	Saves some data in a cache file.
	setCacheDir($cacheDir)	Sets the cache root directory.
	setFileLocking($boolean)	Enables / disables file locking.
	setSuffix($suffix)	Sets the suffix for cache files.
	setWriteControl($boolean)	Enables / disables write control.
boolean	unlink($file)	Removes a file.
boolean	write($path, $file, $data)	Writes the given data in the cache file.
boolean	writeAndControl($path, $file, $data)	

Writes the given data in the cache file and controls it just after to avoid corrupted cache entries.

Method details

- ### __construct ($cacheDir = null)

$cacheDir	The cache root directory

  Constructor.

- ### (boolean) clean ($namespace = null, $mode = &apos;all&apos;)

$namespace	The name of the cache namespace

  Cleans the cache.

  If no namespace is specified all cache files will be destroyed else only cache files of the specified namespace will be destroyed.

  returns true if no problem

- ### (boolean) cleanDir ($dir, $mode, 2)

$dir	The directory complete path
$mode	The name of the cache namespace
2	The flush cache mode : &apos;old&apos;, &apos;all&apos;

  Recursive function for cleaning cache file in the given directory.

  returns true if no problem

  throws sfCacheException

- ### (string) get ($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)

$id	The cache id
$namespace	The name of the cache namespace

$doNotTestCacheValidity	If set to true, the cache validity won&apos;t be tested

Tests if a cache is available and (if yes) returns it.

returns Data of the cache (or null if no cache available)

- ### getCacheDir ()

- ### (boolean) getFileLocking ()

  Gets the value of the fileLocking option.

- ### (array) getFileName ($id, $namespace)

$id	The cache id
$namespace	The name of the namespace

Makes a file name (with path).

returns An array containing the path and the file name

- ### (boolean) getWriteControl ()

  Gets the value of the writeControl option.

- ### (boolean) has ($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)

$id	The cache id
$namespace	The name of the cache namespace
$doNotTestCacheValidity	If set to true, the cache validity won&apos;t be tested

Returns true if there is a cache for the given id and namespace.

returns true if the cache exists, false otherwise

- ### (string) hash ($data)

$data	data

Makes a control key with the string containing datas.

returns control key

- ### initialize ($options = array())

$options	An array of options Available options: ♦ cacheDir: cache root directory ♦ fileLocking: enable / disable file locking (boolean) ♦ writeControl: enable / disable write control (boolean)

- ♦ readControl: enable / disable read control (boolean)
- ♦ fileNameProtection: enable / disable automatic file name protection (boolean)
- ♦ automaticCleaningFactor: disable / tune automatic cleaning process (int)
- ♦ hashedDirectoryLevel: level of the hashed directory system (int)
- ♦ lifeTime: default life time

Initializes the cache.

- ## (int) lastModified ($id, $namespace = self::DEFAULT_NAMESPACE)

  Returns the cache last modification time.

  returns The last modification time

- ## (string) read ($path, $file)

$path	The file path
$file	The file name

  Reads the cache file and returns the content.

  returns The content of the cache file.

  throws sfCacheException

- ## (boolean) remove ($id, $namespace = self::DEFAULT_NAMESPACE)

$id	The cache id
$namespace	The name of the cache namespace

  Removes a cache file.

  returns true if no problem

- ## (boolean) set ($id, $namespace = self::DEFAULT_NAMESPACE, $data)

$id	The cache id
$namespace	The name of the cache namespace
$data	The data to put in cache

  Saves some data in a cache file.

returns true if no problem

- ## setCacheDir ($cacheDir)

$cacheDir	The directory where to put the cache files

Sets the cache root directory.

- ## setFileLocking ($boolean)

Enables / disables file locking.

- ## setSuffix ($suffix)

$suffix	The suffix name (with the leading .)

Sets the suffix for cache files.

- ## setWriteControl ($boolean)

Enables / disables write control.

- ## (boolean) unlink ($file)

$file	The complete file path and name

Removes a file.

returns true if no problem

- ## (boolean) write ($path, $file, $data)

$path	The file path
$file	The file name
$data	The data to put in cache

Writes the given data in the cache file.

returns true if ok

throws sfCacheException

- ## (boolean) writeAndControl ($path, $file, $data)

$path	The file path
$file	The file name
$data	The data to put in cache

Writes the given data in the cache file and controls it just after to avoid corrupted cache entries.

returns true if the test is ok

[more]

Class sfFunctionCache

This class can be used to cache the result and output of functions/methods.

This class is based on the PEAR_Cache_Lite class. All cache files are stored in files in the [sf_root_dir].&apos;/cache/&apos;.[sf_app].&apos;/function&apos; directory. To disable all caching, you can set to false [sf_cache] constant.

Inheritence

sfFunctionCache < sfCache < sfFileCache

Method summary

mixed call() Calls a cacheable function or method (or not if there is already a cache for it).

Methods inherited from sfFileCache

__construct, clean, cleanDir, get, getCacheDir, getFileLocking, getFileName, getWriteControl, has, hash, initialize, lastModified, read, remove, set, setCacheDir, setFileLocking, setSuffix, setWriteControl, unlink, write, writeAndControl

Methods inherited from sfCache

clean, get, getLifeTime, has, lastModified, remove, set, setLifeTime

Method details

- ### (mixed) call ()
  Calls a cacheable function or method (or not if there is already a cache for it).

  Arguments of this method are read with func_get_args. So it doesn&apos;t appear in the function definition. Synopsis : call(&apos;functionName&apos;, $arg1, $arg2, ...) (arg1, arg2... are arguments of &apos;functionName&apos;)

  returns The result of the function/method

[more]

Class sfProcessCache

sfProcessCache stores content in memory if you run a PHP accelerator.

Current PHP accelerator supported: APC, XCache and Eaccelerator.

Method summary

string	cacher()	Gets the cache engine name or false if no PHP accelerator is enabled.
boolean	clear()	Clears the cache.
mixed	get($key)	Gets a value in the cache for the specified key.
string	getPrefix()	Gets the prefix to use for all key name.
boolean	has($key)	Returns true if a given key exists in the cache, false otherwise.
boolean	set($key, $value, $lifeTime)	Sets a value in the cache for the specified key.

Method details

- ### (string) cacher ()

  Gets the cache engine name or false if no PHP accelerator is enabled.

  returns The cache engine name

- ### (boolean) clear ()

  Clears the cache.

  returns true if ok, false otherwise

- ### (mixed) get ($key)

  $key The key name

  Gets a value in the cache for the specified key.

  returns The content associated with the key or null if the key does not exist

- ### (string) getPrefix ()

  Gets the prefix to use for all key name.

  returns The prefix string

- ## (boolean) has ($key)

$key	The key name

  Returns true if a given key exists in the cache, false otherwise.

  returns true if the key exists, false otherwise

- ## (boolean) set ($key, $value, $lifeTime)

$key	The key name
$value	The content to put in cache
$lifeTime	The life time to keep the content in the cache

  Sets a value in the cache for the specified key.

  returns true if ok

[more]

Class sfSQLiteCache

Cache class that stores content in a sqlite database.

Inheritence

sfSQLiteCache < sfCache

Method summary

	__construct($database = null)	Constructor.
	__destruct()	Destructor.
boolean	clean($namespace = null, $mode = &apos;all&apos;)	Cleans the cache.
	createSchema()	Creates the database schema.
string	get($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)	Tests if a cache is available and (if yes) returns it.
string	getDatabase()	Gets the database name.
boolean	has($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)	Returns true if there is a cache for the given id and namespace.
	initialize($options = array())	Initializes the cache.
	lastModified($id, $namespace = self::DEFAULT_NAMESPACE)	
boolean	remove($id, $namespace = self::DEFAULT_NAMESPACE)	Removes an element from the cache.
boolean	set($id, $namespace = self::DEFAULT_NAMESPACE, $data)	Saves some data in the cache.
	setDatabase($database)	Sets the database

name.

Methods inherited from sfCache

clean, get, getLifeTime, has, lastModified, remove, set, setLifeTime

Method details

- ### __construct ($database = null)

$database	The database name

  Constructor.

- ### __destruct ()

  Destructor.

- ### (boolean) clean ($namespace = null, $mode = &apos;all&apos;)

$namespace	The name of the cache namespace

  Cleans the cache.

  If no namespace is specified all cache files will be destroyed else only cache files of the specified namespace will be destroyed.

  returns true if no problem

- ### createSchema ()

  Creates the database schema.

  throws sfException

- ### (string) get ($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)

$id	The cache id
$namespace	The name of the cache namespace
$doNotTestCacheValidity	If set to true, the cache validity won&apos;t be tested

  Tests if a cache is available and (if yes) returns it.

  returns The data in the cache (or null if no cache available)

- ## (string) getDatabase ()

  Gets the database name.

  returns The database name

- ## (boolean) has ($id, $namespace = self::DEFAULT_NAMESPACE, $doNotTestCacheValidity = false)

$id	The cache id
$namespace	The name of the cache namespace
$doNotTestCacheValidity	If set to true, the cache validity won&apos;t be tested

  Returns true if there is a cache for the given id and namespace.

  returns true if the cache exists, false otherwise

- ## initialize ($options = array())

$options	An array of options Available options: ♦ database: database name ♦ automaticCleaningFactor: disable / tune automatic cleaning process (int)

  Initializes the cache.

- ## lastModified ($id, $namespace = self::DEFAULT_NAMESPACE)

- ## (boolean) remove ($id, $namespace = self::DEFAULT_NAMESPACE)

$id	The cache id
$namespace	The name of the cache namespace

  Removes an element from the cache.

  returns true if no problem

- ## (boolean) set ($id, $namespace = self::DEFAULT_NAMESPACE, $data)

$id	The cache id
$namespace	The name of the cache namespace
$data	The data to put in cache

  Saves some data in the cache.

returns true if no problem

- ### setDatabase ($database)

$database	The database name where to store the cache

Sets the database name.

[more]

Class sfAutoloadConfigHandler

Inheritence

sfAutoloadConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfCacheConfigHandler

sfCacheConfigHandler allows you to configure cache.

Inheritence

sfCacheConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	addCache($actionName = &apos;&apos;)	Returns a single addCache statement.
string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) addCache ($actionName = &apos;&apos;)

$actionName	The action name

  Returns a single addCache statement.

  returns PHP code for the addCache statement

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfCompileConfigHandler

sfCompileConfigHandler gathers multiple files and puts them into a single file.

Upon creation of the new file, all comments and blank lines are removed.

Inheritence

sfCompileConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.
	insertConfigFileCallback($matches)	Callback for configuration file insertion in the cache.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ### insertConfigFileCallback ($matches)

  Callback for configuration file insertion in the cache.

[more]

Class sfConfig

sfConfig stores all configuration information for a symfony application.

Method summary

	add($parameters = array())	Sets an array of config parameters.
	clear()	Clears all current config parameters.
mixed	get($name, $default = null)	Retrieves a config parameter.
array	getAll()	Retrieves all configuration parameters.
bool	has($name)	Indicates whether or not a config parameter exists.
	set($name, $value)	Sets a config parameter.

Method details

- ### add ($parameters = array())

$parameters	An associative array of config parameters and their associated values

  Sets an array of config parameters.

  If an existing config parameter name matches any of the keys in the supplied array, the associated value will be overridden.

- ### clear ()

  Clears all current config parameters.

- ### (mixed) get ($name, $default = null)

$name	A config parameter name
$default	A default config parameter value

  Retrieves a config parameter.

  returns A config parameter value, if the config parameter exists, otherwise null

- ### (array) getAll ()

  Retrieves all configuration parameters.

  returns An associative array of configuration parameters.

- ## (bool) has ($name)

  $name A config parameter name

  Indicates whether or not a config parameter exists.

  returns true, if the config parameter exists, otherwise false

- ## set ($name, $value)

  $name A config parameter name

  $value A config parameter value

  Sets a config parameter.

  If a config parameter with the name already exists the value will be overridden.

[more]

Class sfConfigCache

sfConfigCache allows you to customize the format of a configuration file to make it easy-to-use, yet still provide a PHP formatted result for direct inclusion into your modules.

Method summary

	callHandler($handler, $configs, $cache)	Loads a configuration handler.
string	checkConfig($configPath, $optional = false)	Checks to see if a configuration file has been modified and if so recompile the cache file associated with it.
	clear()	Clears all configuration cache files.
string	getCacheName($config)	Converts a normal filename into a cache filename.
	getInstance()	Retrieves the singleton instance of this class.
	import($config, $once = true, $optional = false)	Imports a configuration file.
	loadConfigHandlers()	Loads all configuration application and module level handlers.
	writeCacheFile($config, $cache, [[[amp]]]$data)	Writes a cache file.

Method details

- ### callHandler ($handler, $configs, $cache)

$handler	The handler to use when parsing a configuration file
$configs	An array of absolute filesystem paths to configuration files
$cache	An absolute filesystem path to the cache file that will be written

Loads a configuration handler.

throws sfConfigurationException If a requested configuration file does not have an associated configuration handler

- ### (string) checkConfig ($configPath, $optional = false)

$configPath	A filesystem path to a configuration file

Checks to see if a configuration file has been modified and if so recompile the cache file associated with it.

The recompilation only occurs in a non debug environment.

If the configuration file path is relative, symfony will look in directories defined in the sfLoader::getConfigPaths() method.

returns An absolute filesystem path to the cache filename associated with this specified configuration file

throws sfConfigurationException If a requested configuration file does not exist

- ## clear ()

  Clears all configuration cache files.

- ## (string) getCacheName ($config)

$config	A normal filename

  Converts a normal filename into a cache filename.

  returns An absolute filesystem path to a cache filename

- ## getInstance ()

  Retrieves the singleton instance of this class.

  returns A sfConfigCache instance

- ## import ($config, $once = true, $optional = false)

$config	A filesystem path to a configuration file
$once	Only allow this configuration file to be included once per request?

  Imports a configuration file.

- ## loadConfigHandlers ()

  Loads all configuration application and module level handlers.

  throws sfConfigurationException If a configuration related error occurs.

- ## writeCacheFile ($config, $cache, [[[amp]]]$data)

$config	An absolute filesystem path to a configuration file
$cache	An absolute filesystem path to the cache file that will be written
[[[amp]]]$data	Data to be written to the cache file

  Writes a cache file.

  throws sfCacheException If the cache file cannot be written

[more]

Class sfConfigHandler

sfConfigHandler allows a developer to create a custom formatted configuration file pertaining to any information they like and still have it auto-generate PHP code.

Direct sub-classes

- sfYamlConfigHandler: sfYamlConfigHandler is a base class for YAML (.yml) configuration handlers. This class provides a central location for parsing YAML files and detecting required categories.

Method summary

string	execute($configFiles)	Executes this configuration handler
	getParameterHolder()	Gets the parameter holder for this configuration handler.
bool	initialize($parameters = null)	Initializes this configuration handler.
string	replaceConstants($value)	Replaces constant identifiers in a value.
string	replacePath($path)	Replaces a relative filesystem path with an absolute one.

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of filesystem path to a configuration file

  Executes this configuration handler

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ### getParameterHolder ()

  Gets the parameter holder for this configuration handler.

  returns A sfParameterHolder instance

- ### (bool) initialize ($parameters = null)

$parameters	An associative array of initialization parameters

  Initializes this configuration handler.

returns true, if initialization completes successfully, otherwise false

throws sfInitializationException If an error occurs while initializing this ConfigHandler

- ## (string) replaceConstants ($value)

$value	The value on which to run the replacement procedure

Replaces constant identifiers in a value.

If the value is an array replacements are made recursively.

returns The new value

- ## (string) replacePath ($path)

$path	A relative filesystem path

Replaces a relative filesystem path with an absolute one.

returns The new path

[more]

Class sfDatabaseConfigHandler

sfDatabaseConfigHandler allows you to setup database connections in a configuration file that will be created for you automatically upon first request.

Inheritence

sfDatabaseConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfDefineEnvironmentConfigHandler

Inheritence

sfDefineEnvironmentConfigHandler < sfConfigHandler < sfYamlConfigHandler

Direct sub-classes

- sfLoggingConfigHandler: sfLoggingConfigHandler allows you to configure logging and register loggers with the system.

Method summary

string	execute($configFiles)	Executes this configuration handler.
string	fixCategoryName($category, $prefix)	Fixes the category name.
	fixCategoryValue($category, $key, $value, 3)	Fixes the category name and replaces constants in the value.
	getValues($prefix, $category, $keys, 3)	Gets values from the configuration array.
	mergeEnvironment($config, 1)	Merges default, all and current environment configurations.

Method details

- ### (string) execute ($configFiles)

  $configFiles An absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ### (string) fixCategoryName ($category, $prefix)

$category	The category name
$prefix	The prefix

  Fixes the category name.

  returns The fixed category name

fixCategoryValue ($category, $key, $value, 3)

$category	The category name
$key	The key name
$value	The value
3	Return the new key and value

Fixes the category name and replaces constants in the value.

getValues ($prefix, $category, $keys, 3)

$prefix	The prefix name
$category	The category name
$keys	The key/value array
3	The new key/value array

Gets values from the configuration array.

mergeEnvironment ($config, 1)

$config	The main configuratino array
1	The merged configuration

Merges default, all and current environment configurations.

[more]

Class sfFactoryConfigHandler

sfFactoryConfigHandler allows you to specify which factory implementation the system will use.

Inheritence

sfFactoryConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfFilterConfigHandler

sfFilterConfigHandler allows you to register filters with the system.

Inheritence

sfFilterConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	addFilter($category, $class, $parameters)	Adds a filter statement to the data.
string	addSecurityFilter($category, $class, $parameters)	Adds a security filter statement to the data.
string	execute($configFiles)	Executes this configuration handler

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) addFilter ($category, $class, $parameters)

$category	The category name
$class	The filter class name
$parameters	Filter default parameters

  Adds a filter statement to the data.

  returns The PHP statement

- ### (string) addSecurityFilter ($category, $class, $parameters)

$category	The category name
$class	The filter class name
$parameters	Filter default parameters

  Adds a security filter statement to the data.

returns The PHP statement

- ## (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

Executes this configuration handler

returns Data to be written to a cache file

throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfGeneratorConfigHandler

sfGeneratorConfigHandler.

Inheritence

sfGeneratorConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string execute($configFiles) Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfLoggingConfigHandler

sfLoggingConfigHandler allows you to configure logging and register loggers with the system.

Inheritence

sfLoggingConfigHandler < sfConfigHandler < sfYamlConfigHandler < sfDefineEnvironmentConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.
	getValues($prefix, $category, $keys)	

Methods inherited from sfDefineEnvironmentConfigHandler

execute, fixCategoryName, fixCategoryValue, getValues, mergeEnvironment

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

- ### getValues ($prefix, $category, $keys)

[more]

Class sfPhpConfigHandler

sfPhpConfigHandler allows you to override php.ini configuration at runtime.

Inheritence

sfPhpConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler
string	get_ini_path()	Gets the php.ini path used by PHP.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ### (string) get_ini_path ()

  Gets the php.ini path used by PHP.

  returns the php.ini path

[more]

Class sfRootConfigHandler

sfRootConfigHandler allows you to specify configuration handlers for the application or on a module level.

Inheritence

sfRootConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string execute($configFiles) Executes this configuration handler

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfRoutingConfigHandler

Inheritence

sfRoutingConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfSecurityConfigHandler

sfSecurityConfigHandler allows you to configure action security.

Inheritence

sfSecurityConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

[more]

Class sfSimpleYamlConfigHandler

sfSimpleYamlConfigHandler allows you to load simple configuration files formatted as YAML.

Inheritence

sfSimpleYamlConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	execute($configFiles)	Executes this configuration handler.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

[more]

Class sfValidatorConfigHandler

sfValidatorConfigHandler allows you to register validators with the system.

Inheritence

sfValidatorConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

	convertAlternate2Standard([[[amp]]]$config)
string	execute($configFiles)
boolean	generateRegistration($method, [[[amp]]]$data, [[[amp]]]$methods, [[[amp]]]$names, [[[amp]]]$validators)
	loadAttributes([[[amp]]]$configFiles, [[[amp]]]$methods, [[[amp]]]$names, [[[amp]]]$validators, [[[amp]]]$c
	loadNames([[[amp]]]$configFiles, [[[amp]]]$method, [[[amp]]]$methods, [[[amp]]]$names, [[[amp]]]$config
	loadValidators([[[amp]]]$configFiles, [[[amp]]]$validators, [[[amp]]]$config, [[[amp]]]$list, [[[amp]]]$entry)

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ## convertAlternate2Standard ([[[amp]]]$config)

[[[amp]]]$config	Configuration data

Converts alternate format to standard format.

- ## (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

Executes this configuration handler.

returns Data to be written to a cache file

throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ## (boolean) generateRegistration ($method, [[[amp]]]$data, [[[amp]]]$methods, [[[amp]]]$names, [[[amp]]]$validators)

$method	A request method
[[[amp]]]$data	The data array where our cache code will be appended
[[[amp]]]$methods	An associative array of request method data
[[[amp]]]$names	An associative array of file/parameter data
[[[amp]]]$validators	A validators array

Generates raw cache data.

returns Returns true if there is some validators for this file/parameter

- ## loadAttributes ([[[amp]]]$configFiles, [[[amp]]]$methods, [[[amp]]]$names, [[[amp]]]$validators, [[[amp]]]$config, [[[amp]]]$list)

[[[amp]]]$configFiles	The configuration file name (for exception usage)
[[[amp]]]$methods	An associative array of request method data
[[[amp]]]$names	An associative array of file/parameter names in which to store loaded information
[[[amp]]]$validators	An associative array of validator data
[[[amp]]]$config	The loaded ini configuration that we&apos;ll use for verification purposes
[[[amp]]]$list	A comma delimited list of file/parameter names

Loads the linear list of attributes from the [names] category.

- ## loadNames ([[[amp]]]$configFiles, [[[amp]]]$method, [[[amp]]]$methods, [[[amp]]]$names, [[[amp]]]$config, [[[amp]]]$list)

[[[amp]]]$configFiles	The configuration file name (for exception usage)
[[[amp]]]$method	A request method
[[[amp]]]$methods	An associative array of request method data
[[[amp]]]$names	An associative array of file/parameter names in which to store loaded information
[[[amp]]]$config	The loaded ini configuration that we&apos;ll use for verification purposes
[[[amp]]]$list	A comma delimited list of file/parameter names

Loads all request methods and the file/parameter names that will be validated from the [methods] category.

- ## loadValidators ([[[amp]]]$configFiles, [[[amp]]]$validators, [[[amp]]]$config, [[[amp]]]$list, [[[amp]]]$entry)

[[[amp]]]$configFiles	The configuration file name (for exception usage)
[[[amp]]]$validators	An associative array of validator data
[[[amp]]]$config	The loaded ini configuration that we&apos;ll use for verification purposes
[[[amp]]]$list	A comma delimited list of validator names
[[[amp]]]$entry	A file/parameter name entry

Loads a list of validators.

[more]

Class sfViewConfigHandler

sfViewConfigHandler allows you to configure views.

Inheritence

sfViewConfigHandler < sfConfigHandler < sfYamlConfigHandler

Method summary

string	addComponentSlots($viewName = &apos;&apos;)	Adds a component slot statement to the data.
string	addEscaping($viewName = &apos;&apos;)	Adds an escaping statement to the data.
string	addHtmlAsset($viewName = &apos;&apos;)	Adds stylesheets and javascripts statements to the data.
string	addHtmlHead($viewName = &apos;&apos;)	Adds http metas and metas statements to the data.
string	addLayout($viewName = &apos;&apos;)	Adds a layour statement statement to the data.
string	addTemplate($viewName = &apos;&apos;)	Adds a template setting statement to the data.
string	execute($configFiles)	Executes this configuration handler.
	mergeConfig($myConfig)	Merges assets and environement configuration.

Methods inherited from sfYamlConfigHandler

getConfigValue, mergeConfigValue, parseYaml, parseYamls

Methods inherited from sfConfigHandler

execute, getParameterHolder, initialize, replaceConstants, replacePath

Method details

- ### (string) addComponentSlots ($viewName = &apos;&apos;)

  $viewName The view name
  Adds a component slot statement to the data.

  returns The PHP statement

- ### (string) addEscaping ($viewName = &apos;&apos;)

  $viewName The view name
  Adds an escaping statement to the data.

returns The PHP statement

- ### (string) addHtmlAsset ($viewName = &apos;&apos;)

  $viewName The view name

  Adds stylesheets and javascripts statements to the data.

  returns The PHP statement

- ### (string) addHtmlHead ($viewName = &apos;&apos;)

  $viewName The view name

  Adds http metas and metas statements to the data.

  returns The PHP statement

- ### (string) addLayout ($viewName = &apos;&apos;)

  $viewName The view name

  Adds a layour statement statement to the data.

  returns The PHP statement

- ### (string) addTemplate ($viewName = &apos;&apos;)

  $viewName The view name

  Adds a template setting statement to the data.

  returns The PHP statement

- ### (string) execute ($configFiles)

$configFiles	An array of absolute filesystem path to a configuration file

  Executes this configuration handler.

  returns Data to be written to a cache file

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ### mergeConfig ($myConfig)

  $myConfig A configuration array

  Merges assets and environement configuration.

[more]

Class sfYamlConfigHandler

sfYamlConfigHandler is a base class for YAML (.yml) configuration handlers. This class provides a central location for parsing YAML files and detecting required categories.

Inheritence

sfYamlConfigHandler < sfConfigHandler

Direct sub-classes

- sfRootConfigHandler: sfRootConfigHandler allows you to specify configuration handlers for the application or on a module level.
- sfCacheConfigHandler: sfCacheConfigHandler allows you to configure cache.
- sfSecurityConfigHandler: sfSecurityConfigHandler allows you to configure action security.
- sfAutoloadConfigHandler:
- sfDatabaseConfigHandler: sfDatabaseConfigHandler allows you to setup database connections in a configuration file that will be created for you automatically upon first request.
- sfCompileConfigHandler: sfCompileConfigHandler gathers multiple files and puts them into a single file.
- sfGeneratorConfigHandler: sfGeneratorConfigHandler.
- sfRoutingConfigHandler:
- sfValidatorConfigHandler: sfValidatorConfigHandler allows you to register validators with the system.
- sfViewConfigHandler: sfViewConfigHandler allows you to configure views.
- sfDefineEnvironmentConfigHandler:
- sfSimpleYamlConfigHandler: sfSimpleYamlConfigHandler allows you to load simple configuration files formatted as YAML.
- sfPhpConfigHandler: sfPhpConfigHandler allows you to override php.ini configuration at runtime.
- sfFactoryConfigHandler: sfFactoryConfigHandler allows you to specify which factory implementation the system will use.
- sfFilterConfigHandler: sfFilterConfigHandler allows you to register filters with the system.

Method summary

string	getConfigValue($keyName, $category, $defaultValue = null)	Gets a configuration value for a given key and category.
string	mergeConfigValue($keyName, $category)	Merges configuration values for a given key and category.
string	parseYaml($configFile)	Parses a YAML (.yml) configuration file.
	parseYamls($configFiles, 1)	Parses an array of YAMLs files and merges them in one configuration array.

Method details

- ### (string) getConfigValue ($keyName, $category, $defaultValue = null)

$keyName	The key name
$category	The category name
$defaultValue	The default value

  Gets a configuration value for a given key and category.

  returns The value associated with this key name and category

- ### (string) mergeConfigValue ($keyName, $category)

$keyName	The key name
$category	The category name

  Merges configuration values for a given key and category.

  returns The value associated with this key name and category

- ### (string) parseYaml ($configFile)

$configFile	An absolute filesystem path to a configuration file

  Parses a YAML (.yml) configuration file.

  returns A parsed .yml configuration

  throws sfConfigurationException If a requested configuration file does not exist or is not readable

- ### parseYamls ($configFiles, 1)

$configFiles	An array of configuration file paths
1	A merged configuration array

  Parses an array of YAMLs files and merges them in one configuration array.

[more]

Class sfConsoleController

Inheritence

sfConsoleController < sfController

Method summary

dispatch($moduleName, $actionName, $parameters = array())	Dispatches a request.

Methods inherited from sfController

__call, actionExists, componentExists, controllerExists, forward, getAction, getActionStack, getComponent, getContext, getController, getPresentationFor, getRenderMode, getView, inCLI, initialize, loadFilters, newInstance, sendEmail, setRenderMode

Method details

- ### dispatch ($moduleName, $actionName, $parameters = array())

$moduleName	A module name
$actionName	An action name
$parameters	An associative array of parameters to be set

  Dispatches a request.

[more]

Class sfController

sfController directs application flow.

Direct sub-classes

- sfWebController: sfWebController provides web specific methods to sfController such as, url redirection.
- sfConsoleController:

Method summary

mixed	__call($method, $arguments)	Calls methods defined via the sfMixer class.
bool	actionExists($moduleName, $actionName)	Indicates whether or not a module has a specific action.
bool	componentExists($moduleName, $componentName)	Indicates whether or not a module has a specific component.
boolean	controllerExists($moduleName, $controllerName, $extension, $throwExceptions)	Looks for a controller and optionally throw exceptions if existence is required (i.e.
	forward($moduleName, $actionName)	Forwards the request to another action.
	getAction($moduleName, $actionName)	Retrieves an sfAction implementation instance.
	getActionStack()	Retrieves the action stack.
	getComponent($moduleName, $componentName)	Retrieves a sfComponent implementation

		instance.
	getContext()	Retrieves the current application context.
object A	getController($moduleName, $controllerName, $extension)	Retrieves a controller implementation instance.
string	getPresentationFor($module, $action, $viewName = null)	Returns the rendered view presentation of a given module/action.
int	getRenderMode()	Retrieves the presentation rendering mode.
	getView($moduleName, $actionName, $viewName)	Retrieves a sfView implementation instance.
bool	inCLI()	Indicates whether or not we were called using the CLI version of PHP.
	initialize((sfContext) $context)	Initializes this controller.
	loadFilters((sfFilterChain) $filterChain, (sfAction) $actionInstance)	Loads application nad module filters.
	newInstance($class)	Retrieves a new sfController implementation instance.
string	sendEmail($module, $action)	Sends and email from the current action.
	setRenderMode($mode)	Sets the presentation rendering mode.

Method details

- ### (mixed) __call ($method, $arguments)

$method	The method name
$arguments	The method arguments

Calls methods defined via the sfMixer class.

returns The returned value of the called method

- ### (bool) actionExists ($moduleName, $actionName)

$moduleName	A module name
$actionName	An action name

Indicates whether or not a module has a specific action.

returns true, if the action exists, otherwise false

- ### (bool) componentExists ($moduleName, $componentName)

$moduleName	A module name
$componentName	An component name

Indicates whether or not a module has a specific component.

returns true, if the component exists, otherwise false

- ### (boolean) controllerExists ($moduleName, $controllerName, $extension, $throwExceptions)

$moduleName	The name of the module
$controllerName	The name of the controller within the module
$extension	Either &apos;action&apos; or &apos;component&apos; depending on the type of controller to look for
$throwExceptions	Whether to throw exceptions if the controller doesn&apos;t exist

Looks for a controller and optionally throw exceptions if existence is required (i.e.

in the case of getController).

returns true if the controller exists, false otherwise

throws sfConfigurationException thrown if the module is not enabled

- ## forward ($moduleName, $actionName)

$moduleName	A module name
$actionName	An action name

Forwards the request to another action.

throws sfConfigurationException If an invalid configuration setting has been found

- ## getAction ($moduleName, $actionName)

$moduleName	A module name
$actionName	An action name

Retrieves an sfAction implementation instance.

returns An sfAction implementation instance, if the action exists, otherwise null

- ## getActionStack ()

Retrieves the action stack.

returns An sfActionStack instance, if the action stack is enabled, otherwise null

- ## getComponent ($moduleName, $componentName)

$moduleName	A module name
$componentName	A component name

Retrieves a sfComponent implementation instance.

returns A sfComponent implementation instance, if the component exists, otherwise null

- ## getContext ()

Retrieves the current application context.

returns A sfContext instance

- ## (object A) getController ($moduleName, $controllerName, $extension)

$moduleName	A module name
$controllerName	A component name
$extension	Either &apos;action&apos; or &apos;component&apos; depending on

the type of controller to look for

Retrieves a controller implementation instance.

returns controller implementation instance, if the controller exists, otherwise null

- ## (string) getPresentationFor ($module, $action, $viewName = null)

$module	A module name
$action	An action name
$viewName	A View class name

  Returns the rendered view presentation of a given module/action.

  returns The generated content

- ## (int) getRenderMode ()

  Retrieves the presentation rendering mode.

  returns One of the following:

  - sfView::RENDER_CLIENT
  - sfView::RENDER_VAR

- ## getView ($moduleName, $actionName, $viewName)

$moduleName	A module name
$actionName	An action name
$viewName	A view name

  Retrieves a sfView implementation instance.

  returns A sfView implementation instance, if the view exists, otherwise null

- ## (bool) inCLI ()

  Indicates whether or not we were called using the CLI version of PHP.

  returns true, if using cli, otherwise false.

- ## initialize ((sfContext) $context)

$context	A sfContext implementation instance

  Initializes this controller.

- ## loadFilters ((sfFilterChain) $filterChain, (sfAction) $actionInstance)

$filterChain	A sfFilterChain instance
$actionInstance	A sfAction instance

  Loads application nad module filters.

- ## newInstance ($class)

$class	A sfController class name

  Retrieves a new sfController implementation instance.

  returns A sfController implementation instance

  throws sfFactoryException If a new controller implementation instance cannot be created

- ## (string) sendEmail ($module, $action)

$module	A module name
$action	An action name

  Sends and email from the current action.

  This methods calls a module/action with the sfMailView class.

  returns The generated mail content

- ## setRenderMode ($mode)

$mode	A rendering mode

  Sets the presentation rendering mode.

  throws sfRenderException If an invalid render mode has been set

[more]

Class sfFrontWebController

sfFrontWebController allows you to centralize your entry point in your web application, but at the same time allow for any module and action combination to be requested.

Inheritence

sfFrontWebController < sfController < sfWebController

Method summary

dispatch() Dispatches a request.

Methods inherited from sfWebController

convertUrlStringToParameters, genUrl, redirect

Methods inherited from sfController

__call, actionExists, componentExists, controllerExists, forward, getAction, getActionStack, getComponent, getContext, getController, getPresentationFor, getRenderMode, getView, inCLI, initialize, loadFilters, newInstance, sendEmail, setRenderMode

Method details

- ### dispatch ()
  Dispatches a request.

  This will determine which module and action to use by request parameters specified by the user.

[more]

Class sfRouting

sfRouting class controls the creation of URLs and parses URLs. It maps an array of parameters to URLs definition.

Each map is called a route. It implements the Singleton pattern.

Routing can be disabled when [sf_routing] is set to false.

This class is based on the Routes class of Cake framework.

Method summary

	appendRoute($name, $route, $default = array(), $requirements = array())
	clearRoutes()
array	connect($name, $route, $default = array(), $requirements = array())
string	generate($name, $params, $querydiv = &apos;/&apos;, $divider = &apos;/&apos;, $equals = &apos;/&apos;)
string	getCurrentInternalUri($with_route_name = false)
string	getCurrentRouteName()
	getInstance()
array	getRouteByName($name)
array	getRoutes()

boolean	hasRouteName($name)
boolean	hasRoutes()
array	parse($url)
	prependRoute($name, $route, $default = array(), $requirements = array())
	setCurrentRouteName($name)
array	setRoutes($routes)

Method details

- ### appendRoute ($name, $route, $default = array(), $requirements = array())
  Adds a new route.

  Alias for the connect method.
- ### clearRoutes ()
  Clears all current routes.
- ### (array) connect ($name, $route, $default = array(), $requirements = array())

$name	The route name
$route	The route string

$default	The default parameter values
$requirements	The regexps parameters must match

Adds a new route at the end of the current list of routes.

A route string is a string with 2 special constructions:

- :string: :string denotes a named paramater (available later as $request->getParameter(&apos;string&apos;))
- *: * match an indefinite number of parameters in a route

Here is a very common rule in a symfony project:

```
connect('/:module/:action/*');]]>
```

returns current routes

-

(string) generate ($name, $params, $querydiv = &apos;/&apos;, $divider = &apos;/&apos;, $equals = &apos;/&apos;)

$name	The parameter values
$params	The divider between key/value pairs
$querydiv	The equal sign to use between key and value

Generates a valid URLs for parameters.

returns The generated URL

-

(string) getCurrentInternalUri ($with_route_name = false)

$with_route_name	Whether to give an internal URI with the route name (@route) or with the module/action pair

Gets the internal URI for the current request.

returns The current internal URI

-

(string) getCurrentRouteName ()

Gets the current route name.

returns The route name

-

getInstance ()

Retrieve the singleton instance of this class.

returns The sfRouting implementation instance

- ### (array) getRouteByName ($name)

$name	The route name

  Gets a route by its name.

  returns A route array

- ### (array) getRoutes ()

  Gets the current compiled route array.

  returns The route array

- ### (boolean) hasRouteName ($name)

$name	The route name

  Returns true if the route name given is defined.

- ### (boolean) hasRoutes ()

  Returns true if this instance has some routes.

- ### (array) parse ($url)

$url	URL to be parsed

  Parses a URL to find a matching route.

  Returns null if no route match the URL.

  returns An array of parameters

- ### prependRoute ($name, $route, $default = array(), $requirements = array())

  Adds a new route at the beginning of the current list of routes.

- ### setCurrentRouteName ($name)

$name	The route name

  Sets the current route name.

- ### (array) setRoutes ($routes)

$routes	The route array

  Sets the compiled route array.

returns The route array

[more]

Class sfWebController

sfWebController provides web specific methods to sfController such as, url redirection.

Inheritence

sfWebController < sfController

Direct sub-classes

- sfFrontWebController: sfFrontWebController allows you to centralize your entry point in your web application, but at the same time allow for any module and action combination to be requested.

Method summary

array	convertUrlStringToParameters($url)	Converts an internal URI string to an array of parameters.
string	genUrl($parameters = array(), $absolute = false)	Generates an URL from an array of parameters.
	redirect($url, $delay, $statusCode = 302)	Redirects the request to another URL.

Method details

- ### (array) convertUrlStringToParameters ($url)

$url	An internal URI

  Converts an internal URI string to an array of parameters.

  returns An array of parameters

- ### (string) genUrl ($parameters = array(), $absolute = false)

$parameters	An associative array of URL parameters or an internal URI as a string.
$absolute	Whether to generate an absolute URL

  Generates an URL from an array of parameters.

  returns A URL to a symfony resource

- ### redirect ($url, $delay, $statusCode = 302)

$url	An existing URL

$delay	A delay in seconds before redirecting. This is only needed on browsers that do not support HTTP headers
$statusCode	The status code

Redirects the request to another URL.

[more]

Class Countable

Method summary

count()

Method details

- ### count ()

[more]

Class sfDatabase

sfDatabase is a base abstraction class that allows you to setup any type of database connection via a configuration file.

Direct sub-classes

- sfPostgreSQLDatabase: sfPostgreSQLDatabase provides connectivity for the PostgreSQL brand database.
- sfPDODatabase: sfPDODatabase provides connectivity for the PDO database abstraction layer.
- sfMySQLDatabase: sfMySQLDatabase provides connectivity for the MySQL brand database.

Method summary

	connect()	Connects to the database.
mixed	getConnection()	Retrieves the database connection associated with this sfDatabase implementation.
string	getParameter($name, $default = null, $ns = null)	Gets the parameter associated with the given key.
	getParameterHolder()	Gets the parameter holder for this object.
mixed	getResource()	Retrieves a raw database resource associated with this sfDatabase implementation.
boolean	hasParameter($name, $ns = null)	Returns true if the given key exists in the parameter holder.
bool	initialize($parameters = array())	Initializes this sfDatabase object.
	setParameter($name, $value, $ns = null)	Sets the value for the given key.
	shutdown()	Executes the shutdown procedure.

Method details

-

  ### connect ()

  Connects to the database.

  throws sfDatabaseException If a connection could not be created

-

  ### (mixed) getConnection ()

  Retrieves the database connection associated with this sfDatabase implementation.

  When this is executed on a Database implementation that isn&apos;t an abstraction layer, a copy of the resource will be returned.

  returns A database connection

throws sfDatabaseException If a connection could not be retrieved

- ## (string) getParameter ($name, $default = null, $ns = null)

$name	The key name
$default	The default value
$ns	The namespace to use

Gets the parameter associated with the given key.

This is a shortcut for:

```
getParameterHolder()->get()]]>
```

returns The value associated with the key

- ## getParameterHolder ()

Gets the parameter holder for this object.

returns A sfParameterHolder instance

- ## (mixed) getResource ()

Retrieves a raw database resource associated with this sfDatabase implementation.

returns A database resource

throws sfDatabaseException If a resource could not be retrieved

- ## (boolean) hasParameter ($name, $ns = null)

$name	The key name
$ns	The namespace to use

Returns true if the given key exists in the parameter holder.

This is a shortcut for:

```
getParameterHolder()->has()]]>
```

returns true if the given key exists, false otherwise

- ## (bool) initialize ($parameters = array())

$parameters	An associative array of initialization parameters

Initializes this sfDatabase object.

returns true, if initialization completes successfully, otherwise false

throws sfInitializationException If an error occurs while initializing this sfDatabase object

- ## setParameter ($name, $value, $ns = null)

$name	The key name
$value	The value
$ns	The namespace to use

Sets the value for the given key.

This is a shortcut for:

```
getParameterHolder()->set()]]>
```

- ## shutdown ()

Executes the shutdown procedure.

throws sfDatabaseException If an error occurs while shutting down this database

[more]

Class sfDatabaseManager

sfDatabaseManager allows you to setup your database connectivity before the request is handled. This eliminates the need for a filter to manage database connections.

Method summary

mixed	getDatabase($name = &apos;default&apos;)	Retrieves the database connection associated with this sfDatabase implementation.
bool	initialize()	Initializes this sfDatabaseManager object
	shutdown()	Executes the shutdown procedure

Method details

- ### (mixed) getDatabase ($name = &apos;default&apos;)

  $name A database name

  Retrieves the database connection associated with this sfDatabase implementation.

  returns A Database instance

  throws sfDatabaseException If the requested database name does not exist

- ### (bool) initialize ()

  Initializes this sfDatabaseManager object

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this sfDatabaseManager object

- ### shutdown ()

  Executes the shutdown procedure

  throws sfDatabaseException If an error occurs while shutting down this DatabaseManager

[more]

Class sfMySQLDatabase

sfMySQLDatabase provides connectivity for the MySQL brand database.

Inheritence

sfMySQLDatabase < sfDatabase

Method summary

	connect()	Connects to the database.
array&	loadParameters([[[amp]]]$array)	Loads connection parameters from an existing array.
	shutdown()	Execute the shutdown procedure

Methods inherited from sfDatabase

connect, getConnection, getParameter, getParameterHolder, getResource, hasParameter, initialize, setParameter, shutdown

Method details

- ### connect ()
  Connects to the database.

  throws sfDatabaseException If a connection could not be created
- ### (array&) loadParameters ([[[amp]]]$array)
  Loads connection parameters from an existing array.

  returns An associative array of connection parameters
- ### shutdown ()
  Execute the shutdown procedure

  throws sfDatabaseException If an error occurs while shutting down this database

[more]

Class sfPDODatabase

sfPDODatabase provides connectivity for the PDO database abstraction layer.

Inheritence

sfPDODatabase < sfDatabase

Method summary

connect()	Connects to the database.
shutdown()	Executes the shutdown procedure.

Methods inherited from sfDatabase

connect, getConnection, getParameter, getParameterHolder, getResource, hasParameter, initialize, setParameter, shutdown

Method details

- ### connect ()

  Connects to the database.

  throws sfDatabaseException If a connection could not be created

- ### shutdown ()

  Executes the shutdown procedure.

  throws sfDatabaseException If an error occurs while shutting down this database

[more]

Class sfPostgreSQLDatabase

sfPostgreSQLDatabase provides connectivity for the PostgreSQL brand database.

Inheritence

sfPostgreSQLDatabase < sfDatabase

Method summary

	connect()	Connects to the database.
string	loadParameters([[[amp]]]$array)	Loads connection parameters from an existing array.
	shutdown()	Executes the shutdown procedure.

Methods inherited from sfDatabase

connect, getConnection, getParameter, getParameterHolder, getResource, hasParameter, initialize, setParameter, shutdown

Method details

- ### connect ()
  Connects to the database.

  throws sfDatabaseException If a connection could not be created
- ### (string) loadParameters ([[[amp]]]$array)
  Loads connection parameters from an existing array.

  returns A connection string
- ### shutdown ()
  Executes the shutdown procedure.

  throws sfDatabaseException If an error occurs while shutting down this database

[more]

Class sfPropelDatabase

A symfony database driver for Propel, derived from the native Creole driver.

Method summary

addConfig()
getConfiguration()
initialize($parameters = null, $name = &apos;propel&apos;)
setConnectionParameter($key, $value)
setDefaultConfig()

Method details

- ### addConfig ()
- ### getConfiguration ()
- ### initialize ($parameters = null, $name = &apos;propel&apos;)
- ### setConnectionParameter ($key, $value)
- ### setDefaultConfig ()

[more]

Class sfPropelDataRetriever

sfGenerator is the abstract base class for all generators.

Method summary

retrieveObjects($class, $peerMethod = null)

Method details

- ### retrieveObjects ($class, $peerMethod = null)

[more]

Class sfDebug

sfDebug provides some method to help debugging a symfony application.

Method summary

array	flattenParameterHolder((sfParameterHolder) $parameterHolder)	Returns a parameter holder as an array.
array	globalsAsArray()	Returns PHP globals variables as a sorted array.
array	phpInfoAsArray()	Returns PHP information as an array.
array	requestAsArray((sfRequest) $request)	Returns request parameter holders as an array.
array	responseAsArray((sfResponse) $response)	Returns response parameters as an array.
array	settingsAsArray()	Returns sfConfig variables as a sorted array.

Method details

- ### (array) flattenParameterHolder ((sfParameterHolder) $parameterHolder)

  $parameterHolder A sfParameterHolder instance

  Returns a parameter holder as an array.

  returns The parameter holder as an array

- ### (array) globalsAsArray ()

  Returns PHP globals variables as a sorted array.

  returns PHP globals

- ### (array) phpInfoAsArray ()

  Returns PHP information as an array.

  returns An array of php information

- ### (array) requestAsArray ((sfRequest) $request)

  $request A sfRequest instance

Returns request parameter holders as an array.

returns The request parameter holders

- ### (array) responseAsArray ((sfResponse) $response)

$response	A sfResponse instance

Returns response parameters as an array.

returns The response parameters

- ### (array) settingsAsArray ()

Returns sfConfig variables as a sorted array.

returns sfConfig variables

[more]

Class sfWebDebug

sfWebDebug creates debug information for easy debugging in the browser.

Method summary

string	decorateContentWithDebug($internalUri, $content, $new = false)	Decorates a chunk of HTML with cache information.
string	formatArrayAsHtml($id, $values)	Converts an array to HTML.
string	formatLogLine($log_line)	Formats a log line.
string	getCurrentConfigAsHtml()	Returns the current configuration as HTML.
	getInstance()	Retrieves the singleton instance of this class.
string	getPriority($value)	Converts a propriety value to a string.
string	getResults()	Returns the web debug toolbar as HTML.
	initialize()	
	loadHelpers()	Loads helpers needed for the web debug toolbar.
	log($logEntry)	Logs a message to the web debug toolbar.
	logShortMessage($message)	Logs a short message to be displayed in the web debug toolbar.
	registerAssets()	Registers javascripts and stylesheets needed for the web debug toolbar.

Method details

- ### (string) decorateContentWithDebug ($internalUri, $content, $new = false)

$internalUri	The internalUri representing the content
$content	The HTML content
$new	true if the content is new in the cache, false otherwise

  Decorates a chunk of HTML with cache information.

returns The decorated HTML string

- ## (string) formatArrayAsHtml ($id, $values)

$id	The identifier to use
$values	The array of values

  Converts an array to HTML.

  returns An HTML string

- ## (string) formatLogLine ($log_line)

$log_line	The log line to format

  Formats a log line.

  returns The formatted log lin

- ## (string) getCurrentConfigAsHtml ()

  Returns the current configuration as HTML.

  returns The current configuration as HTML

- ## getInstance ()

  Retrieves the singleton instance of this class.

  returns A sfWebDebug implementation instance

- ## (string) getPriority ($value)

$value	The priority value

  Converts a proprity value to a string.

  returns The priority as a string

- ## (string) getResults ()

  Returns the web debug toolbar as HTML.

  returns The web debug toolbar HTML

- ## initialize ()

- ## loadHelpers ()

  Loads helpers needed for the web debug toolbar.

- ### log ($logEntry)

$logEntry	An array of parameter

  Logs a message to the web debug toolbar.

- ### logShortMessage ($message)

$message	The message string

  Logs a short message to be displayed in the web debug toolbar.

- ### registerAssets ()

  Registers javascripts and stylesheets needed for the web debug toolbar.

[more]

Class sfActionException

sfActionException is thrown when an error occurs in an action.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfAutoloadException

sfAutoloadException is thrown when a class that has been required cannot be loaded.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfCacheException

sfCacheException is thrown when ConfigCache fails to execute properly.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

Class constructor.

[more]

Class sfConfigurationException

sfConfigurationException is thrown when the framework finds an error in a configuration setting.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfContextException

sfContextException is thrown when an instance of the context has not been created.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfControllerException

sfControllerException is thrown when a requested Controller implementation doesn&apos;t exist.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfDatabaseException

sfDatabaseException is thrown when a database related error occurs.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfError404Exception

sfError404Exception is thrown when a 404 error occurs in an action.

Method summary

__construct($message = null, $code)	Class constructor.
printStackTrace((Exception) $exception = null)	Forwards to the 404 action.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

- ### printStackTrace ((Exception) $exception = null)

$exception	An Exception implementation instance

  Forwards to the 404 action.

[more]

Class sfException

sfException is the base class for all symfony related exceptions and provides an additional method for printing up a detailed view of an exception.

Direct sub-classes

- sfCacheException: sfCacheException is thrown when ConfigCache fails to execute properly.
- sfForwardException: sfForwardException is thrown when an error occurs while attempting to forward the request.
- sfStorageException: sfStorageException is thrown when a requested sfStorage implementation doesn&apos;t exist or data cannot be read from or written to the storage.
- sfConfigurationException: sfConfigurationException is thrown when the framework finds an error in a configuration setting.
- sfInitializationException: sfInitializationException is thrown when an initialization procedure fails.
- sfParseException: sfParseException is thrown when a parsing procedure fails to complete successfully.
- sfDatabaseException: sfDatabaseException is thrown when a database related error occurs.
- sfActionException: sfActionException is thrown when an error occurs in an action.
- sfControllerException: sfControllerException is thrown when a requested Controller implementation doesn&apos;t exist.
- sfStopException: sfStopException is thrown when you want to stop action flow.
- sfFactoryException: sfFactoryException is thrown when an error occurs while attempting to create a new factory implementation instance.
- sfViewException: sfViewException is thrown when an error occurs in a view.
- sfFileException: sfFileException is thrown when an error occurs while moving an uploaded file.
- sfFilterException: sfFilterException is thrown when an error occurs while attempting to initialize or execute a filter.
- sfAutoloadException: sfAutoloadException is thrown when a class that has been required cannot be loaded.
- sfError404Exception: sfError404Exception is thrown when a 404 error occurs in an action.
- sfSecurityException: sfSecurityException is thrown when a security related error occurs.
- sfRenderException: sfRenderException is thrown when a view&apos;s pre-render check fails.
- sfValidatorException: sfValidatorException is thrown when an error occurs in a validator.
- sfContextException: sfContextException is thrown when an instance of the context has not been created.

Method summary

	__construct($message = null, $code)	Class constructor.
string	fileExcerpt($file, $line)	Returns an excerpt of a code file around the given line number.
string	formatArgs($args, $single = false, $format = &apos;html&apos;)	Formats an array as a string.
string	formatArrayAsHtml($values)	Returns an HTML version of an array as YAML.

string	getName()	Retrieves the name of this exception.
array	getTraces((Exception) $exception, $format = &apos;plain&apos;)	Returns an array of exception traces.
	printStackTrace((Exception) $exception = null)	Prints the stack trace for this exception.
	setName($name)	Sets the name of this exception.

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

- ### (string) fileExcerpt ($file, $line)

$file	A file path
$line	The selected line number

  Returns an excerpt of a code file around the given line number.

  returns An HTML string

- ### (string) formatArgs ($args, $single = false, $format = &apos;html&apos;)

$args	The argument array
$format	The format string (html or plain)

  Formats an array as a string.

- ### (string) formatArrayAsHtml ($values)

$values	The values array

  Returns an HTML version of an array as YAML.

  returns An HTML string

- ### (string) getName ()

  Retrieves the name of this exception.

  returns This exception&apos;s name

- ### (array) getTraces ((Exception) $exception, $format = &apos;plain&apos;)

$exception	An Exception implementation instance
$format	The trace format (plain or html)

  Returns an array of exception traces.

  returns An array of traces

- ### printStackTrace ((Exception) $exception = null)

$exception	An Exception implementation instance

  Prints the stack trace for this exception.

- ### setName ($name)

$name	An exception name

  Sets the name of this exception.

[more]

Class sfFactoryException

sfFactoryException is thrown when an error occurs while attempting to create a new factory implementation instance.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfFileException

sfFileException is thrown when an error occurs while moving an uploaded file.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfFilterException

sfFilterException is thrown when an error occurs while attempting to initialize or execute a filter.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfForwardException

sfForwardException is thrown when an error occurs while attempting to forward the request.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfInitializationException

sfInitializationException is thrown when an initialization procedure fails.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfParseException

sfParseException is thrown when a parsing procedure fails to complete successfully.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfRenderException

sfRenderException is thrown when a view&apos;s pre-render check fails.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfSecurityException

sfSecurityException is thrown when a security related error occurs.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfStopException

sfStopException is thrown when you want to stop action flow.

Method summary

__construct($message = null, $code)	Class constructor.
printStackTrace($exception = null)	Stops the current action.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

- ### printStackTrace ($exception = null)

  Stops the current action.

[more]

Class sfStorageException

sfStorageException is thrown when a requested sfStorage implementation doesn&apos;t exist or data cannot be read from or written to the storage.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

Class constructor.

[more]

Class sfValidatorException

sfValidatorException is thrown when an error occurs in a validator.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfViewException

sfViewException is thrown when an error occurs in a view.

Method summary

__construct($message = null, $code)	Class constructor.

Methods inherited from sfException

__construct, fileExcerpt, formatArgs, formatArrayAsHtml, getName, getTraces, printStackTrace, setName

Method details

- ### __construct ($message = null, $code)

$message	The error message
$code	The error code

  Class constructor.

[more]

Class sfBasicSecurityFilter

sfBasicSecurityFilter checks security by calling the getCredential() method of the action. Once the credential has been acquired, sfBasicSecurityFilter verifies the user has the same credential by calling the hasCredential() method of SecurityUser.

Inheritence

sfBasicSecurityFilter < sfFilter < sfSecurityFilter

Method summary

execute((sfFilterChain) $filterChain)	Executes this filter.

Methods inherited from sfSecurityFilter

newInstance

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

$filterChain	A sfFilterChain instance

  Executes this filter.

[more]

Class sfCacheFilter

sfCacheFilter deals with page caching and action caching.

Inheritence

sfCacheFilter < sfFilter

Method summary

	execute((sfFilterChain) $filterChain)	Executes this filter.
	executeBeforeExecution()	
	executeBeforeRendering((sfFilterChain) 0)	Executes this filter.
	getActionCache($uri)	Gets an action template from the cache.
	getPageCache($uri)	Gets a page template from the cache.
bool	initialize((sfContext) $context, $parameters = array())	Initializes this Filter.
	setActionCache($uri)	Sets an action template in the cache.
	setPageCache($uri)	Sets a page template in the cache.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

$filterChain	A sfFilterChain instance

  Executes this filter.

- ### executeBeforeExecution ()

- ### executeBeforeRendering ((sfFilterChain) 0)

0	A sfFilterChain instance.

  Executes this filter.

- ## getActionCache ($uri)

$uri	The internal URI

  Gets an action template from the cache.

- ## getPageCache ($uri)

$uri	The internal URI

  Gets a page template from the cache.

- ## (bool) initialize ((sfContext) $context, $parameters = array())

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this Filter.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Filter

- ## setActionCache ($uri)

$uri	The internal URI

  Sets an action template in the cache.

- ## setPageCache ($uri)

$uri	The internal URI

  Sets a page template in the cache.

[more]

Class sfCommonFilter

sfCommonFilter automatically adds javascripts and stylesheets information in the sfResponse content.

Inheritence

sfCommonFilter < sfFilter

Method summary

execute((sfFilterChain) $filterChain)	Executes this filter.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

$filterChain	A sfFilterChain instance

  Executes this filter.

[more]

Class sfExecutionFilter

sfExecutionFilter is the last filter registered for each filter chain. This filter does all action and view execution.

Inheritence

sfExecutionFilter < sfFilter

Method summary

execute((sfFilterChain) $filterChain)	Executes this filter.
registerFillInFilter((sfFilterChain) $filterChain, $parameters)	Registers the fill in filter in the filter chain.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

$filterChain	The filter chain

  Executes this filter.

  throws sfInitializeException If an error occurs during view initialization.

- ### registerFillInFilter ((sfFilterChain) $filterChain, $parameters)

$filterChain	A sfFilterChain implementation instance
$parameters	An array of parameters to pass to the fill in filter.

  Registers the fill in filter in the filter chain.

[more]

Class sfFillInFormFilter

sfFillInFormFilter fills in forms.

Inheritence

sfFillInFormFilter < sfFilter

Method summary

execute((sfFilterChain) $filterChain) Executes this filter.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

$filterChain	A sfFilterChain instance

  Executes this filter.

[more]

Class sfFilter

sfFilter provides a way for you to intercept incoming requests or outgoing responses.

Direct sub-classes

- sfFakeRenderingFilter: sfFilter provides a way for you to intercept incoming requests or outgoing responses.
- sfFlashFilter: sfFlashFilter removes flash attributes from the session.
- sfCommonFilter: sfCommonFilter automatically adds javascripts and stylesheets information in the sfResponse content.
- sfRenderingFilter: sfRenderingFilter is the last filter registered for each filter chain. This filter does the rendering.
- sfFillInFormFilter: sfFillInFormFilter fills in forms.
- sfCacheFilter: sfCacheFilter deals with page caching and action caching.
- sfSecurityFilter: sfSecurityFilter provides a base class that classifies a filter as one that handles security.
- sfExecutionFilter: sfExecutionFilter is the last filter registered for each filter chain. This filter does all action and view execution.
- sfWebDebugFilter:

Method summary

	getContext()	Retrieves the current application context.
string	getParameter($name, $default = null, $ns = null)	Gets the parameter associated with the given key.
	getParameterHolder()	Gets the parameter holder for this object.
boolean	hasParameter($name, $ns = null)	Returns true if the given key exists in the parameter holder.
boolean	initialize((sfContext) $context, $parameters = array())	Initializes this Filter.
boolean	isFirstCall()	Returns true if this is the first call to the sfFilter instance.
	setParameter($name, $value, $ns = null)	Sets the value for the given key.

Method details

- ### getContext ()
  Retrieves the current application context.

  returns The current sfContext instance

- ### (string) getParameter ($name, $default = null, $ns = null)

$name	The key name
$default	The default value
$ns	The namespace to use

  Gets the parameter associated with the given key.

  This is a shortcut for:

  ```
  getParameterHolder()->get()]]>
  ```

  returns The value associated with the key

- ### getParameterHolder ()

  Gets the parameter holder for this object.

  returns A sfParameterHolder instance

- ### (boolean) hasParameter ($name, $ns = null)

$name	The key name
$ns	The namespace to use

  Returns true if the given key exists in the parameter holder.

  This is a shortcut for:

  ```
  getParameterHolder()->has()]]>
  ```

  returns true if the given key exists, false otherwise

- ### (boolean) initialize ((sfContext) $context, $parameters = array())

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this Filter.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Filter

- ### (boolean) isFirstCall ()

  Returns true if this is the first call to the sfFilter instance.

  returns true if this is the first call to the sfFilter instance, false otherwise

- ### setParameter ($name, $value, $ns = null)

$name	The key name
$value	The value
$ns	The namespace to use

Sets the value for the given key.

This is a shortcut for:

```
getParameterHolder()->set()]]>
```

[more]

Class sfFilterChain

sfFilterChain manages registered filters for a specific context.

Method summary

	execute()	Executes the next filter in this chain.
boolean	hasFilter($class)	Returns true if the filter chain contains a filter of a given class.
	register((sfFilter) $filter)	Registers a filter with this chain.

Method details

- ### execute ()

  Executes the next filter in this chain.

- ### (boolean) hasFilter ($class)

$class	The class name of the filter

  Returns true if the filter chain contains a filter of a given class.

  returns true if the filter exists, false otherwise

- ### register ((sfFilter) $filter)

$filter	A sfFilter implementation instance.

  Registers a filter with this chain.

[more]

Class sfFlashFilter

sfFlashFilter removes flash attributes from the session.

Inheritence

sfFlashFilter < sfFilter

Method summary

execute((sfFilterChain) $filterChain) Executes this filter.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

$filterChain	A sfFilterChain instance.

  Executes this filter.

[more]

Class sfRenderingFilter

sfRenderingFilter is the last filter registered for each filter chain. This filter does the rendering.

Inheritence

sfRenderingFilter < sfFilter

Method summary

execute((sfFilterChain) $filterChain) Executes this filter.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ((sfFilterChain) $filterChain)

  $filterChain The filter chain.
  Executes this filter.

  throws sfInitializeException If an error occurs during view initialization

[more]

Class sfSecurityFilter

sfSecurityFilter provides a base class that classifies a filter as one that handles security.

Inheritence

sfSecurityFilter < sfFilter

Direct sub-classes

- sfBasicSecurityFilter: sfBasicSecurityFilter checks security by calling the getCredential() method of the action. Once the credential has been acquired, sfBasicSecurityFilter verifies the user has the same credential by calling the hasCredential() method of SecurityUser.

Method summary

newInstance($class) Returns a new instance of a sfSecurityFilter.

Method details

- ### newInstance ($class)

  $class The security class name

  Returns a new instance of a sfSecurityFilter.

  returns A sfSecurityFilter implementation instance

[more]

Class sfWebDebugFilter

Inheritence

sfWebDebugFilter < sfFilter

Method summary

execute((sfFilterChain) $filterChain) Executes this filter.

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

-

execute ((sfFilterChain) $filterChain)

$filterChain	A sfFilterChain instance

Executes this filter.

[more]

Class sfAdminColumn

Propel admin generator column.

Method summary

	__call($name, $arguments)	
	__construct($phpName, $column = null, $flags = array())	Constructor.
string	getName()	Gets the name of the column.
string	getPhpName()	Gets the php name of the column.
boolean	isComponent()	Returns true if the column is a component.
boolean	isLink()	Returns true if the column has a link.
boolean	isPartial()	Returns true if the column is a partial.
boolean	isReal()	Returns true if the column maps a database column.

Method details

- ### __call ($name, $arguments)

- ### __construct ($phpName, $column = null, $flags = array())

$phpName	The column php name
$column	The column name
$flags	The column flags

  Constructor.

- ### (string) getName ()

  Gets the name of the column.

  returns The column name

- ### (string) getPhpName ()

  Gets the php name of the column.

  returns The php name

- ### (boolean) isComponent ()

  Returns true if the column is a component.

  returns true if the column is a component, false otherwise
- ### (boolean) isLink ()

  Returns true if the column has a link.

  returns true if the column has a link, false otherwise
- ### (boolean) isPartial ()

  Returns true if the column is a partial.

  returns true if the column is a partial, false otherwise
- ### (boolean) isReal ()

  Returns true if the column maps a database column.

  returns true if the column maps a database column, false otherwise

[more]

Class sfAdminGenerator

Admin generator.

This class generates an admin module.

This class calls two ORM specific methods: getAllColumns() and getAdminColumnForField($field, $flag = null)

Inheritence

sfAdminGenerator < sfGenerator < sfCrudGenerator

Direct sub-classes

- sfPropelCrudGenerator: Propel CRUD generator.

Method summary

string	addCredentialCondition($content, $params = array())	Wraps content with a credential condition.
	escapeString($string, 1)	Escapes a string.
string	getButtonToAction($actionName, $params, $pk_link = false)	Returns HTML code for an action button.
array	getColumnCategories($paramName)	Returns all column categories.
string	getColumnEditTag($column, $params = array())	Returns HTML code for a column in edit mode.
string	getColumnFilterTag($column, $params = array())	Returns HTML code for a column in filter mode.
string	getColumnListTag($column, $params = array())	Returns HTML code for a column in list mode.
array	getColumns($paramName, $category = &apos;NONE&apos;)	Gets sfAdminColumn objects for a given category.
mixed	getFieldParameterValue($key, $type = &apos;&apos;, $default = null)	Gets a field parameter value.
string	getHelp($column, $type = &apos;&apos;)	Returns HTML code for a help text.
string	getHelpAsIcon($column, $type = &apos;&apos;)	Returns HTML code for a help icon.
string	getI18NString($key, $default = null, $withEcho = true)	Wraps a content for I18N.
string	getLinkToAction($actionName, $params, $pk_link = false)	Returns HTML code for an action link.

mixed	getParameterValue($key, $default = null)	Gets a parameter value.
mixed	getValueFromKey($key, $default = null)	Gets the value for a given key.
string	replaceConstants($value)	Replaces constants in a string.
array	splitFlag($text)	Gets modifier flags from a column name.

Method details

(string) addCredentialCondition ($content, $params = array())

$content	The content
$params	The parameters

Wraps content with a credential condition.

returns HTML code

escapeString ($string, 1)

Escapes a string.

(string) getButtonToAction ($actionName, $params, $pk_link = false)

$actionName	The action name
$params	The parameters
$pk_link	Whether to add a primary key link or not

Returns HTML code for an action button.

returns HTML code

(array) getColumnCategories ($paramName)

$paramName	The parameter name

Returns all column categories.

returns The column categories

- ### (string) getColumnEditTag ($column, $params = array())

$column	The column name
$params	The parameters

Returns HTML code for a column in edit mode.

returns HTML code

- ### (string) getColumnFilterTag ($column, $params = array())

$column	The column name
$params	The parameters

Returns HTML code for a column in filter mode.

returns HTML code

- ### (string) getColumnListTag ($column, $params = array())

$column	The column name
$params	The parameters

Returns HTML code for a column in list mode.

returns HTML code

- ### (array) getColumns ($paramName, $category = &apos;NONE&apos;)

$paramName	The parameter name

Gets sfAdminColumn objects for a given category.

returns sfAdminColumn array

- ### (mixed) getFieldParameterValue ($key, $type = &apos;&apos;, $default = null)

$key	The key name
$type	The type (list,

	edit)
$default	The default value

Gets a field parameter value.

returns The parameter value

- ## (string) getHelp ($column, $type = &apos;&apos;)

$column	The column name
$type	The field type (list, edit)

Returns HTML code for a help text.

returns HTML code

- ## (string) getHelpAsIcon ($column, $type = &apos;&apos;)

$column	The column name
$type	The field type (list, edit)

Returns HTML code for a help icon.

returns HTML code

- ## (string) getI18NString ($key, $default = null, $withEcho = true)

$key	The key name
$default	The defaul value

Wraps a content for I18N.

returns HTML code

- ## (string) getLinkToAction ($actionName, $params, $pk_link = false)

$actionName	The action name
$params	The parameters
$pk_link	Whether to add a primary key link or not

Returns HTML code for an action link.

returns HTML code

- ### (mixed) getParameterValue ($key, $default = null)

$key	The key name
$default	The default value

  Gets a parameter value.

  returns The parameter value

- ### (mixed) getValueFromKey ($key, $default = null)

$key	The key name
$default	The default value

  Gets the value for a given key.

  returns The key value

- ### (string) replaceConstants ($value)

  Replaces constants in a string.

- ### (array) splitFlag ($text)

$text	The column name

  Gets modifier flags from a column name.

  returns An array of detected flags

[more]

Class sfCrudGenerator

CRUD generator.

This class generates a basic CRUD module.

Inheritence

sfCrudGenerator < sfGenerator

Direct sub-classes

- sfAdminGenerator: Admin generator.

Method summary

string	generate($params = array())	Generates classes and ter in cache.
string	getClassName()	Gets the class name for c scaffolding class.
string	getColumnGetter($column, $developed = false, $prefix = &apos;&apos;)	Returns the getter either non-developped: &apos;getFoo&apos; or developped: &apos;$class->getFoo()
string	getColumnListTag($column, $params = array())	Returns HTML code for column in list mode.
string	getCrudColumnEditTag($column, $params = array())	Returns HTML code for column in edit mode.
object	getMap()	Gets the Map object.
string	getMethodParamsForGetOrCreate()	Returns PHP code for getOrCreate() parameters
string	getObjectTagParams($params, $default_params = array())	Gets object tag paramete
string	getPHPObjectHelper($helperName, $column, $params, $localParams = array())	Generates a PHP call to a object helper.
string	getPeerClassName()	Gets the Peer class name
string	getPluralName()	Gets the plural name for scaffolding class.
string	getPrimaryKey()	Gets the primary key nar
string	getPrimaryKeyIsSet($prefix = &apos;&apos;)	Gets PHP code for prima condition.
string	getPrimaryKeyUrlParams($prefix = &apos;&apos;)	

		Returns PHP code to ad URL for primary keys.
	getRelatedClassName($column)	
string	getRetrieveByPkParamsForAction($indent)	Returns PHP code for pr keys parameters.
string	getRetrieveByPkParamsForGetOrCreate()	Returns PHP code for pr keys parameters used in getOrCreate() method.
string	getSingularName()	Gets the singular name f current scaffolding class
TableMap	getTableMap()	Gets the table map for th current model class.
string	getTestPksForGetOrCreate($fieldNameAsArgument = true)	Returns PHP code for getOrCreate() promary k condition.
	loadMapBuilderClasses()	Loads map builder classe
	loadPrimaryKeys()	Loads primary keys.
	setScaffoldingClassName($className)	Sets the class name to us scaffolding

Method details

- ### (string) generate ($params = array())

$params	The parameters

  Generates classes and templates in cache.

  returns The data to put in configuration cache

- ### (string) getClassName ()

  Gets the class name for current scaffolding class.

- ### (string) getColumnGetter ($column, $developed = false, $prefix = &apos;&apos;)

$column	The column name
$developed	true if you want developped method names, false otherwise
$prefix	The prefix value

  Returns the getter either non-developped: &apos;getFoo&apos; or developped: &apos;$class->getFoo()&apos;.

This method is ORM dependant.

returns PHP code

-
(string) getColumnListTag ($column, $params = array())

$column	The column name
$params	The parameters

Returns HTML code for a column in list mode.

returns HTML code

-
(string) getCrudColumnEditTag ($column, $params = array())

$column	The column name
$params	The parameters

Returns HTML code for a column in edit mode.

returns HTML code

-
(object) getMap ()

Gets the Map object.

-
(string) getMethodParamsForGetOrCreate ()

Returns PHP code for getOrCreate() parameters.

returns The PHP code

-
(string) getObjectTagParams ($params, $default_params = array())

$params	An array of parameters
$default_params	An array of default parameters

Gets object tag parameters.

returns PHP code

-
(string) getPHPObjectHelper ($helperName, $column, $params, $localParams = array())

$helperName	The helper name

$column	The column name
$params	An array of parameters
$localParams	An array of local parameters

Generates a PHP call to an object helper.

This method is ORM dependant.

returns PHP code

- ## (string) getPeerClassName ()
  Gets the Peer class name.
- ## (string) getPluralName ()
  Gets the plural name for current scaffolding class.
- ## (string) getPrimaryKey ()
  Gets the primary key name.
- ## (string) getPrimaryKeyIsSet ($prefix = &apos;&apos;)

$prefix	The prefix value

Gets PHP code for primary key condition.

returns PHP code

- ## (string) getPrimaryKeyUrlParams ($prefix = &apos;&apos;)

$prefix	The prefix value

Returns PHP code to add to a URL for primary keys.

returns PHP code

- ## getRelatedClassName ($column)
- ## (string) getRetrieveByPkParamsForAction ($indent)

$indent	The indentation value

Returns PHP code for primary keys parameters.

returns The PHP code

- ### (string) getRetrieveByPkParamsForGetOrCreate ()
  Returns PHP code for primary keys parameters used in getOrCreate() method.

  returns The PHP code
- ### (string) getSingularName ()
  Gets the singular name for current scaffolding class.
- ### (TableMap) getTableMap ()
  Gets the table map for the current model class.

  returns A TableMap instance
- ### (string) getTestPksForGetOrCreate ($fieldNameAsArgument = true)

$fieldNameAsArgument	true if we pass the field name as an argument, false otherwise

  Returns PHP code for getOrCreate() promary keys condition.

  returns The PHP code
- ### loadMapBuilderClasses ()
  Loads map builder classes.

  This method is ORM dependant.

  throws sfException
- ### loadPrimaryKeys ()
  Loads primary keys.

  This method is ORM dependant.

  throws sfException
- ### setScaffoldingClassName ($className)

$className	class name

  Sets the class name to use for scaffolding

[more]

Class sfGenerator

sfGenerator is the abstract base class for all generators.

Direct sub-classes

- sfCrudGenerator: CRUD generator.

Method summary

mixed	__call($method, $arguments)	Calls methods defined via the sfMixer class.
string	evalTemplate($templateFile)	Evaluates a temp file.
string	generate($params = array())	Generates classe and templates.
	generatePhpFiles($generatedModuleName, $templateFiles = array(), $configFiles = array())	Generates PHP f for a given modu name.
string	getGeneratedModuleName()	Gets the module name of the generated modul
string	getGeneratorClass()	Gets the generat class.
string	getGeneratorManager()	Gets the sfGeneratorMan instance.
string	getModuleName()	Gets the module name.
string	getTheme()	Gets the theme name.
	initialize((sfGeneratorManager) $generatorManager)	Initializes the cu sfGenerator insta
string	replacePhpMarks($text)	Replaces PHP m by <?php ?>.
	setGeneratedModuleName($module_name)	Sets the module name of the generated modul
	setGeneratorClass($generator_class)	Sets the generato class.
	setModuleName($module_name)	

	Sets the module name.
setTheme($theme)	Sets the theme n

Method details

- ### (mixed) __call ($method, $arguments)

$method	The method name
$arguments	The method arguments

  Calls methods defined via the sfMixer class.

  returns The returned value of the called method

- ### (string) evalTemplate ($templateFile)

$templateFile	The template file path

  Evaluates a template file.

  returns The evaluated template

- ### (string) generate ($params = array())

$params	An array of parameters

  Generates classes and templates.

  returns The cache for the configuration file

- ### generatePhpFiles ($generatedModuleName, $templateFiles = array(), $configFiles = array())

$generatedModuleName	The name of module name to generate
$templateFiles	A list of template files to generate
$configFiles	A list of configuration files to generate

  Generates PHP files for a given module name.

- ### (string) getGeneratedModuleName ()

  Gets the module name of the generated module.

  returns The module name

- ### (string) getGeneratorClass ()
  Gets the generator class.

  returns The generator class

- ### (string) getGeneratorManager ()
  Gets the sfGeneratorManager instance.

  returns The sfGeneratorManager instance

- ### (string) getModuleName ()
  Gets the module name.

  returns The module name

- ### (string) getTheme ()
  Gets the theme name.

  returns The theme name

- ### initialize ((sfGeneratorManager) $generatorManager)

$generatorManager	A sfGeneratorManager instance

  Initializes the current sfGenerator instance.

- ### (string) replacePhpMarks ($text)

$text	The PHP code

  Replaces PHP marks by <?php ?>.

  returns The converted PHP code

- ### setGeneratedModuleName ($module_name)

$module_name	The module name

  Sets the module name of the generated module.

- ### setGeneratorClass ($generator_class)

$generator_class	The generator class

  Sets the generator class.

- ## setModuleName ($module_name)

$module_name	The module name

  Sets the module name.

- ## setTheme ($theme)

$theme	The theme name

  Sets the theme name.

[more]

Class sfGeneratorManager

sfGeneratorManager helps generate classes, views and templates for scaffolding, admin interface, ...

Method summary

string	generate($generator_class, $param)	Generates classes and templates for a given generator class.
	getCache()	Returns the current sfCache implementation instance.
	initialize()	Initializes the sfGeneratorManager instance.

Method details

- ### (string) generate ($generator_class, $param)

$generator_class	The generator class name
$param	An array of parameters

  Generates classes and templates for a given generator class.

  returns The cache for the configuration file

- ### getCache ()

  Returns the current sfCache implementation instance.

  returns A sfCache implementation instance

- ### initialize ()

  Initializes the sfGeneratorManager instance.

[more]

Class sfPropelAdminGenerator

Propel Admin generator.

This class generates an admin module with propel.

Inheritence

sfPropelAdminGenerator < sfGenerator < sfCrudGenerator < sfAdminGenerator < sfPropelCrudGenerator

Method summary

getAdminColumnForField($field, $flag = null)

getAllColumns()

getColumnForPhpName($phpName)

initialize((sfGeneratorManager) $generatorManager) Initializes the current sfGenerator instance.

Methods inherited from sfPropelCrudGenerator

getColumnGetter, getPHPObjectHelper, getRelatedClassName, initialize, loadMapBuilderClasses, loadPrimaryKeys

Methods inherited from sfAdminGenerator

addCredentialCondition, escapeString, getButtonToAction, getColumnCategories, getColumnEditTag, getColumnFilterTag, getColumnListTag, getColumns, getFieldParameterValue, getHelp, getHelpAsIcon, getI18NString, getLinkToAction, getParameterValue, getValueFromKey, replaceConstants, splitFlag

Methods inherited from sfCrudGenerator

generate, getClassName, getColumnGetter, getColumnListTag, getCrudColumnEditTag, getMap, getMethodParamsForGetOrCreate, getObjectTagParams, getPHPObjectHelper, getPeerClassName, getPluralName, getPrimaryKey, getPrimaryKeyIsSet, getPrimaryKeyUrlParams, getRelatedClassName, getRetrieveByPkParamsForAction, getRetrieveByPkParamsForGetOrCreate, getSingularName, getTableMap, getTestPksForGetOrCreate, loadMapBuilderClasses, loadPrimaryKeys, setScaffoldingClassName

Methods inherited from sfGenerator

__call, evalTemplate, generate, generatePhpFiles, getGeneratedModuleName, getGeneratorClass, getGeneratorManager, getModuleName, getTheme, initialize, replacePhpMarks, setGeneratedModuleName, setGeneratorClass, setModuleName, setTheme

Method details

- ### getAdminColumnForField ($field, $flag = null)
- ### getAllColumns ()
- ### getColumnForPhpName ($phpName)
- ### initialize ((sfGeneratorManager) $generatorManager)

$generatorManager	A sfGeneratorManager instance

  Initializes the current sfGenerator instance.

[more]

Class sfPropelCrudGenerator

Propel CRUD generator.

This class generates a basic CRUD module with propel.

Inheritence

sfPropelCrudGenerator < sfGenerator < sfCrudGenerator < sfAdminGenerator

Direct sub-classes

- sfPropelAdminGenerator: Propel Admin generator.

Method summary

string	getColumnGetter($column, $developed = false, $prefix = &apos;&apos;)	Returns the getter either non-developped: &apos;getFoo&apos; or developped: &apos;$class->getFoo()&apo
string	getPHPObjectHelper($helperName, $column, $params, $localParams = array())	Generates a PHP call to an object helper.
	getRelatedClassName($column)	
	initialize((sfGeneratorManager) $generatorManager)	Initializes the current sfGenerator instance.
	loadMapBuilderClasses()	Loads map builder classes.
	loadPrimaryKeys()	Loads primary keys.

Method details

- ### (string) getColumnGetter ($column, $developed = false, $prefix = &apos;&apos;)

$column	The column name
$developed	true if you want developped method names, false otherwise
$prefix	The prefix value

  Returns the getter either non-developped: &apos;getFoo&apos; or developped: &apos;$class->getFoo()&apos;.

  returns PHP code

- ### (string) getPHPObjectHelper ($helperName, $column, $params, $localParams = array())

$helperName	The helper name
$column	The column name
$params	An array of parameters
$localParams	An array of local parameters

  Generates a PHP call to an object helper.

  returns PHP code

- ### getRelatedClassName ($column)

- ### initialize ((sfGeneratorManager) $generatorManager)

$generatorManager	A sfGeneratorManager instance

  Initializes the current sfGenerator instance.

- ### loadMapBuilderClasses ()

  Loads map builder classes.

  This method is ORM dependant.

  throws sfException

- ### loadPrimaryKeys ()

  Loads primary keys.

  This method is ORM dependant.

  throws sfException

[more]

Functions in AssetHelper

AssetHelper.

Function summary

	_compute_public_path($source, $dir, $ext, $absolute = false)	
string	auto_discovery_link_tag($type = &apos;rss&apos;, $url_options = array(), $tag_options = array())	Returns a <link> tag that browsers and news readers ca use to auto-detec a RSS or ATOM fe for the current page, to b included i the <head section of HTML document
	decorate_with($layout)	Decorates the curren template with a giv layout.
string	get_javascripts()	Returns <script> tags for al javascript configured in view.y or added t the respon object.
string	get_stylesheets()	Returns <link> tag for all stylesheet configured in view.y

		or added t the respon object.
string	image_path($source, $absolute = false)	Returns th path to an image ass
string	image_tag($source, $options = array())	Returns an <img> image tag for the ass given as argument.
string	include_http_metas()	Returns a set of <me http-equiv tags according the respon attributes, be include in the <head> section of HTML document
	include_javascripts()	Prints <script> tags for al javascripts configured in view.ym or added t the respon object.
string	include_metas()	Prints a se of <meta> tags according the respon attributes, be include in the <head> section of HTML document
	include_stylesheets()	

	Prints <link> tag for all stylesheet configured in view.ym or added t the respon object.
string include_title()	Returns th title of the current pa according the respon attributes, be include in the <title> section of HTML document
string javascript_include_tag(0)	Returns a <script> include ta per source given as argument.
string javascript_path($source, $absolute = false)	Returns th path to a JavaScript asset.
string stylesheet_path($source, $absolute = false)	Returns th path to a stylesheet asset.
string stylesheet_tag(0, 1)	Returns a css <link> tag per source given as argument, to be included i the <head section of HTML document

use_javascript($js, $position = &apos;&apos;)	Adds a javascript the respon object.
use_stylesheet($css, $position = &apos;&apos;, $options = array())	Adds a stylesheet the respon object.

Function details

- ### _compute_public_path ($source, $dir, $ext, $absolute = false)

- ### (string) auto_discovery_link_tag ($type = &apos;rss&apos;, $url_options = array(), $tag_options = array())

$type	feed type (&apos;rss&apos;, &apos;atom&apos;)
$url_options	&apos;module/action&apos; or &apos;@rule&apos; of the feed
$tag_options	additional HTML compliant <link> tag parameters

Returns a <link> tag that browsers and news readers can use to auto-detect a RSS or ATOM feed for the current page, to be included in the <head> section of a HTML document.

Options:

- rel - defaults to &apos;alternate&apos;
- type - defaults to &apos;application/rss+xml&apos;
- title - defaults to the feed type in upper case

Examples:

```
echo auto_discovery_link_tag('rss', 'module/feed', array('title' => 'My RSS'));
  => ]]>
```

returns XHTML compliant <link> tag

- ### decorate_with ($layout)

$layout	The layout name or path or false to disable the layout

Decorates the current template with a given layout.

- ### (string) get_javascripts ()

Returns <script> tags for all javascripts configured in view.yml or added to the response object.

You can use this helper to decide the location of javascripts in pages. By default, if you don&apos;t call this helper, symfony will automatically include javascripts before </head>. Calling this helper disables this behavior.

returns <script> tags

-

(string) get_stylesheets ()

Returns <link> tags for all stylesheets configured in view.yml or added to the response object.

You can use this helper to decide the location of stylesheets in pages. By default, if you don&apos;t call this helper, symfony will automatically include stylesheets before </head>. Calling this helper disables this behavior.

returns <link> tags

-

(string) image_path ($source, $absolute = false)

$source	asset name
$absolute	return absolute path ?

Returns the path to an image asset.

Example:

```
/images/foobar.png]]>
```

Note: The asset name can be supplied as a...

- ♦ full path, like "/my_images/image.gif"
- ♦ file name, like "rss.gif", that gets expanded to "/images/rss.gif"
- ♦ file name without extension, like "logo", that gets expanded to "/images/logo.png"

returns file path to the image file

-

(string) image_tag ($source, $options = array())

$source	image asset name
$options	additional HTML compliant <img> tag parameters

Returns an <img> image tag for the asset given as argument.

Options:

- ♦ &apos;absolute&apos; - to output absolute file paths, useful for embedded images in emails
- ♦ &apos;alt&apos; - defaults to the file name part of the asset (capitalized and without the extension)
- ♦ &apos;size&apos; - Supplied as "XxY", so "30x45" becomes width="30" and height="45"

Examples:

```
echo image_tag('/my_images/image.gif', array('alt' => 'Alternative text', 'size' => '100
```

```
  =>                ]]>
```

returns XHTML compliant <img> tag

(string) include_http_metas ()

Returns a set of <meta http-equiv> tags according to the response attributes, to be included in the <head> section of a HTML document.

Examples:

```
]]>
```

Note: Modify the sfResponse object or the view.yml to change, add or remove metas.

returns XHTML compliant <meta> tag(s)

include_javascripts ()

Prints <script> tags for all javascripts configured in view.yml or added to the response object.

(string) include_metas ()

Prints a set of <meta> tags according to the response attributes, to be included in the <head> section of a HTML document.

Examples:

```
        ]]>
```

Note: Modify the sfResponse object or the view.yml to change, add or remove metas.

returns XHTML compliant <meta> tag(s)

- ## include_stylesheets ()

  Prints <link> tags for all stylesheets configured in view.yml or added to the response object.

- ## (string) include_title ()

  Returns the title of the current page according to the response attributes, to be included in the <title> section of a HTML document.

  Note: Modify the sfResponse object or the view.yml to modify the title of a page.

  returns page title

- ## (string) javascript_include_tag (0)

0	asset names

  Returns a <script> include tag per source given as argument.

  Examples:

  ```
  echo javascript_include_tag('common.javascript', '/elsewhere/cools');
    =>
       ]]>
  ```

  returns XHTML compliant <script> tag(s)

- ## (string) javascript_path ($source, $absolute = false)

$source	asset name
$absolute	return absolute path ?

  Returns the path to a JavaScript asset.

  Example:

  ```
  /js/myscript.js]]>
  ```

  Note: The asset name can be supplied as a...

    - full path, like "/my_js/myscript.css"
    - file name, like "myscript.js", that gets expanded to "/js/myscript.js"
    - file name without extension, like "myscript", that gets expanded to "/js/myscript.js"

  returns file path to the JavaScript file

- ## (string) stylesheet_path ($source, $absolute = false)

$source	asset name
$absolute	return absolute path ?

  Returns the path to a stylesheet asset.

Example:

```
/css/style.css]]>
```

Note: The asset name can be supplied as a...

- full path, like "/my_css/style.css"
- file name, like "style.css", that gets expanded to "/css/style.css"
- file name without extension, like "style", that gets expanded to "/css/style.css"

returns file path to the stylesheet file

-

(string) stylesheet_tag (0, 1)

0	asset names
1	additional HTML compliant <link> tag parameters

Returns a css <link> tag per source given as argument, to be included in the <head> section of a HTML document.

Options:

- rel - defaults to &apos;stylesheet&apos;
- type - defaults to &apos;text/css&apos;
- media - defaults to &apos;screen&apos;

Examples:

```
 echo stylesheet_tag('style', array('media' => 'all'));
   =>
 echo stylesheet_tag('random.styles', '/css/stylish');
   =>
      ]]>
```

returns XHTML compliant <link> tag(s)

-

use_javascript ($js, $position = &apos;&apos;)

Adds a javascript to the response object.

-

use_stylesheet ($css, $position = &apos;&apos;, $options = array())

Adds a stylesheet to the response object.

[more]

Functions in CacheHelper

CacheHelper.

Function summary

cache($name, $lifeTime = 86400)
cache_save()

Function details

- ### cache ($name, $lifeTime = 86400)
- ### cache_save ()

[more]

Functions in DateFormHelper

DateFormHelper.

Function summary

string	_get_I18n_date_locales($culture = null)	Retrieves the proper date format based on the specified setting
string	_parse_value_for_date($value, $key, $format_char)	Converts date values () into its correct date format ()
string	select_ampm_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with AM and PM options for use with 12-Hour time.
string	select_date_tag($name, $value = null, $options = array(), $html_options = array())	Returns three <select> tags populated with a range of months, days, and years.
string	select_datetime_tag($name, $value = null, $options = array(), $html_options = array())	Returns a variable number of <select> tags populated with date and time related select boxes.

string select_day_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with all the days of the month (1 - 31).
string select_hour_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with 24 hours (0 - 23), or optionally 12 hours (1 - 12).
string select_minute_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with 60 minutes (0 - 59).
string select_month_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with all the months of the year (1 - 12).
string select_number_tag($name, $value, $options = array(), $html_options = array())	Returns a <select> tag, populated with a range of numbers
string select_second_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with 60 seconds (0 - 59).
string select_time_tag($name, $value = null, $options = array(), $html_options = array())	Returns three <select> tags populated with hours, minutes, and

		optionally seconds.
string	select_timezone_tag($name, $selected = null, $options = array())	Returns a <select> tag populated with all the timezones in the world.
string	select_year_tag($name, $value = null, $options = array(), $html_options = array())	Returns a <select> tag populated with a range of years.

Function details

- ## (string) _get_I18n_date_locales ($culture = null)

$culture	two or three character culture setting variable

  Retrieves the proper date format based on the specified setting

  Note: If no $culture is defined, the user&apos;s culture setting will be used in its place.

  returns formatted date/time format based on the specified date/time setting

- ## (string) _parse_value_for_date ($value, $key, $format_char)

$value	date or date part
$key	custom key for array values

  Converts date values () into its correct date format ()

  This function is primarily used in select_date_tag, select_time_tag and select_datetime_tag.

  Note: If $value is empty, it will be populated with the current date and time.

  returns properly formatted date part value.

- ## (string) select_ampm_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name
$value	selected value (AM or PM)
$options	

additional HTML compliant <select> tag parameters

Returns a <select> tag populated with AM and PM options for use with 12-Hour time.

By default, the $value parameter is set to the correct AM/PM setting based on the current time. To override this, simply pass either AM or PM to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. For convenience, Symfony also offers the select_time_tag select_datetime_tag helper functions which combine other date and time helpers to easily build date and time select boxes.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.

Examples:

```
true));]]>
```

returns <select> tag populated with AM and PM options for use with 12-Hour time.

- ## (string) select_date_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name (automatically becomes an array of parts: name[year], name[month], year[day])
$value	accepts a valid date string or properly formatted date array
$options	additional HTML compliant <select> tag parameters

Returns three <select> tags populated with a range of months, days, and years.

By default, the $value parameter is set to today&apos;s month, day and year. To override this, simply pass a valid date or a correctly formatted date array (see example) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. Also, the default selectable range of years is set to show five years back and five years forward from today&apos;s year. For instance, if today&apos;s year is 2006, the default &apos;year_start&apos; option will be set to 2001 and the &apos;year_end&apos; option will be set to 2011. These start and end dates can easily be overwritten by setting the &apos;year_start&apos; and &apos;year_end&apos; options in the $options parameter.

Note: The $name parameter will automatically converted to array names. For example, a $name of "date" becomes: <select name="date[month]">...</select> <select name="date[day]">...</select> <select name="date[year]">...</select>

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- discard_month - If set to true, will only return select tags for day and year.
- discard_day - If set to true, will only return select tags for month and year.
- discard_year - If set to true, will only return select tags for month and day.
- use_month_numbers - If set to true, will show the month&apos;s numerical value (1 - 12) instead of the months full name.
- use_short_month - If set to true, will show the month&apos;s short name (i.e. Jan, Feb, Mar) instead of its full name.
- year_start - If set, the range of years will begin at this four-digit date (i.e. 1979)
- year_end - If set, the range of years will end at this four-digit date (i.e. 2025)
- date_seperator - Includes a string of defined text between each generated select tag

Examples:

```
 '1979', 'month' => 10, 'day' => 30);
  echo select_date_tag('date', $date, array('year_start' => $date['year'] - 10, 'year_end'
```

returns three <select> tags populated with a months, days and years

-

(string) select_datetime_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name (automatically becomes an array of date and time parts)
$value	accepts a valid time string or properly formatted time array
$options	additional HTML compliant <select> tag parameters

Returns a variable number of <select> tags populated with date and time related select boxes.

The select_datetime_tag is the culmination of both the select_date_tag and the select_time_tag. By default, the $value parameter is set to the current date and time. To override this, simply pass a valid date, time, datetime string or correctly formatted array (see example) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. To include seconds to the result, use set the &apos;include_second&apos; option in the $options parameter to true. **Note:** The $name parameter will automatically converted to array names. For example, a $name of "datetime" becomes: <select name="datetime[month]">...</select> <select name="datetime[day]">...</select> <select name="datetime[year]">...</select> <select name="datetime[hour]">...</select> <select name="datetime[minute]">...</select> <select name="datetime[second]">...</select>

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- include_second - If set to true, includes the "seconds" select tag as part of the result.
- discard_month - If set to true, will only return select tags for day and year.
- discard_day - If set to true, will only return select tags for month and year.
- discard_year - If set to true, will only return select tags for month and day.
- use_month_numbers - If set to true, will show the month&apos;s numerical value (1 - 12) instead of the months full name.
- use_short_month - If set to true, will show the month&apos;s short name (i.e. Jan, Feb, Mar) instead of its full name.
- year_start - If set, the range of years will begin at this four-digit date (i.e. 1979)
- year_end - If set, the range of years will end at this four-digit date (i.e. 2025)
- second_step - If set, the seconds will be incremented in blocks of X, where X = &apos;second_step&apos;
- minute_step - If set, the minutes will be incremented in blocks of X, where X = &apos;minute_step&apos;
- 12hour_time - If set to true, will return integers 1 through 12 instead of the default 0 through 23.
- date_seperator - Includes a string of defined text between each generated select tag
- time_seperator - Includes a string of defined text between each generated select tag
- ampm_seperator - Includes a string of defined text between the minute/second select box and the AM/PM select box

Examples:

```
 '1979', 'month' => 10, 'day' => 30, 'hour' => '15', 'minute' => 46);
  echo select_datetime_tag('time', $datetime, array('use_short_month' => true, '12hour_tim
```

returns a variable number of <select> tags populated with date and time related select boxes

(string) select_day_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name
$value	selected value (1 - 31)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with all the days of the month (1 - 31).

By default, the $value parameter is set to today&apos;s day. To override this, simply pass an integer (1 - 31) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. For convenience, symfony also offers the select_date_tag helper function which combines the select_year_tag, select_month_tag, and select_day_tag functions into a single helper.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.

Examples:

returns <select> tag populated with all the days of the month (1 - 31).

- ## (string) select_hour_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name
$value	selected value (0 - 23 or 1 - 12 if &apos;12hour_time&apos; = true)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with 24 hours (0 - 23), or optionally 12 hours (1 - 12).

By default, the $value parameter is set to the current hour. To override this, simply pass an integer (0 - 23 or 1 - 12 if &apos;12hour_time&apos; = true) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. For convenience, Symfony also offers the select_time_tag select_datetime_tag helper functions which combine other date and time helpers to easily build date and time select boxes.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- 12hour_time - If set to true, will return integers 1 through 12 instead of the default 0 through 23 as well as an AM/PM select box.

Examples:

```
true));]]>
```

returns <select> tag populated with 24 hours (0 - 23), or optionally 12 hours (1 - 12).

- ## (string) select_minute_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name

$value	selected value (0 - 59)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with 60 minutes (0 - 59).

By default, the $value parameter is set to the current minute. To override this, simply pass an integer (0 - 59) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. In many cases, you have no need for all 60 minutes in an hour. the &apos;minute_step&apos; option in the $options parameter gives you the ability to define intervals to display. So for instance you could define 15 as your &apos;minute_step&apos; interval and the select tag would return the values 0, 15, 30, and 45. For convenience, Symfony also offers the select_time_tag select_datetime_tag helper functions which combine other date and time helpers to easily build date and time select boxes.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- minute_step - If set, the minutes will be incremented in blocks of X, where X = &apos;minute_step&apos;

Examples:

```
15));]]>
```

returns <select> tag populated with 60 minutes (0 - 59).

-

(string) select_month_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name
$value	selected value (1 - 12)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with all the months of the year (1 - 12).

By default, the $value parameter is set to today&apos;s month. To override this, simply pass an integer (1 - 12) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. Also, the each month&apos;s display title is set to return its respective full month name, which can be easily overridden by passing the &apos;use_short_names&apos; or &apos;use_month_numbers&apos; options to the $options parameter. For convenience, Symfony also offers the select_date_tag helper function which combines the select_year_tag, select_month_tag, and select_day_tag functions into a single helper.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value
- use_month_numbers - If set to true, will show the month&apos;s numerical value (1 - 12) instead of the months full name.
- use_short_month - If set to true, will show the month&apos;s short name (i.e. Jan, Feb, Mar) instead of its full name.

Examples:

```
true));]]>
```

```
true, 'include_blank' => true));]]>
```

returns <select> tag populated with all the months of the year (1 - 12).

- ## (string) select_number_tag ($name, $value, $options = array(), $html_options = array())

$name	field name
$value	the selected option
$options	$options to manipulate the output of the tag.
$html_options	additional HTML compliant <select> tag parameters

Returns a <select> tag, populated with a range of numbers

By default, the select_number_tag generates a list of numbers from 1 - 10, with an incremental value of 1. These values can be easily changed by passing one or several $options. Numbers can be either positive or negative, integers or decimals, and can be incremented by any number, decimal or integer. If you require the range of numbers to be listed in descending order, pass the &apos;reverse&apos; option to easily display the list of numbers in the opposite direction.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- multiple - If set to true, the select tag will allow multiple numbers to be selected at once.
- start - The first number in the list. If not specified, the default value is 1.
- end - The last number in the list. If not specified, the default value is 10.
- increment - The number by which to increase each number in the list by until the number is greater than or equal to the &apos;end&apos; option. If not specified, the default value is 1.
- reverse - Reverses the order of numbers so they are display in descending order

Examples:

```
true));]]>

'0.05', 'end' => '0.09', 'increment' => '0.01'));]]>

5, 'end' => 120, 'increment' => 15));]]>
```

returns <select> tag populated with a range of numbers.

- ## (string) select_second_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name
$value	selected value (0 - 59)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with 60 seconds (0 - 59).

By default, the $value parameter is set to the current second (right now). To override this, simply pass an integer (0 - 59) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. In many cases, you have no need for all 60 seconds in a minute. the &apos;second_step&apos; option in the $options parameter gives you the ability to define intervals to display. So for instance you could define 15 as your &apos;minute_step&apos; interval and the select tag would return the values 0, 15, 30, and 45. For convenience, Symfony also offers the select_time_tag select_datetime_tag helper functions which combine other date and time helpers to easily build date and time select boxes.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- second_step - If set, the seconds will be incremented in blocks of X, where X = &apos;second_step&apos;

Examples:

```
15));]]>
```

returns <select> tag populated with 60 seconds (0 - 59).

- ## (string) select_time_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name (automatically becomes an array of parts: name[hour], name[minute], year[second])

$value	accepts a valid time string or properly formatted time array
$options	additional HTML compliant <select> tag parameters

Returns three <select> tags populated with hours, minutes, and optionally seconds.

By default, the $value parameter is set to the current hour and minute. To override this, simply pass a valid time or a correctly formatted time array (see example) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. To include seconds to the result, use set the &apos;include_second&apos; option in the $options parameter to true. **Note:** The $name parameter will automatically converted to array names. For example, a $name of "time" becomes: <select name="time[hour]">...</select> <select name="time[minute]">...</select> <select name="time[second]">...</select>

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value.
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value.
- include_second - If set to true, includes the "seconds" select tag as part of the result.
- second_step - If set, the seconds will be incremented in blocks of X, where X = &apos;second_step&apos;
- minute_step - If set, the minutes will be incremented in blocks of X, where X = &apos;minute_step&apos;
- 12hour_time - If set to true, will return integers 1 through 12 instead of the default 0 through 23 as well as an AM/PM select box.
- time_seperator - Includes a string of defined text between each generated select tag
- ampm_seperator - Includes a string of defined text between the minute/second select box and the AM/PM select box

Examples:

```
'15', 'minute' => 46, 'second' => 01);
 echo select_time_tag('time', $time, array('include_second' => true, '12hour_time' => tru
```

returns three <select> tags populated with a hours, minutes and optionally seconds.

(string) select_timezone_tag ($name, $selected = null, $options = array())

$name	field name
$selected	selected field value (timezone identifier)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with all the timezones in the world.

The select_timezone_tag builds off the traditional select_tag function, and is conveniently populated with all the timezones in the world (sorted alphabetically). Each option in the list has a unique timezone identifier for its value and the timezone&apos;s locale as its display title. The timezone data is retrieved via the sfCultureInfo class, which stores a wide variety of i18n and i10n settings for various countries and cultures throughout the world. Here&apos;s an example of an <option> tag generated by the select_timezone_tag:

Options:

- display - identifer - Display the PHP timezone identifier (e.g. America/Denver) timezone - Display the full timezone name (e.g. Mountain Standard Time) timezone_abbr - Display the timezone abbreviation (e.g. MST) timzone_dst - Display the full timezone name with daylight savings time (e.g. Mountain Daylight Time) timezone_dst_abbr - Display the timezone abbreviation with daylight savings time (e.g. MDT) city - Display the city/region that relates to the timezone (e.g. Denver)

<option value="America/Denver">America/Denver</option>

Examples:

returns <select> tag populated with all the timezones in the world.

-

(string) select_year_tag ($name, $value = null, $options = array(), $html_options = array())

$name	field name
$value	selected value within the range of years.
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with a range of years.

By default, the $value parameter is set to today&apos;s year. To override this, simply pass a four-digit integer (YYYY) to the $value parameter. You can also set the $value parameter to null which will disable the $value, however this will only be useful if you pass &apos;include_blank&apos; or &apos;include_custom&apos; to the $options parameter. Also, the default selectable range of years is set to show five years back and five years forward from today&apos;s year. For instance, if today&apos;s year is 2006, the default &apos;year_start&apos; option will be set to 2001 and the &apos;year_end&apos; option will be set to 2011. These start and end dates can easily be overwritten by setting the &apos;year_start&apos; and &apos;year_end&apos; options in the $options parameter. For convenience, Symfony also offers the select_date_tag helper function which combines the select_year_tag, select_month_tag, and select_day_tag functions into a single helper.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value

- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value
- year_start - If set, the range of years will begin at this four-digit date (i.e. 1979)
- year_end - If set, the range of years will end at this four-digit date (i.e. 2025)

Examples:

```
 $year_start, 'year_end' => $year_end));]]>
```

returns <select> tag populated with a range of years.

[more]

Functions in DateHelper

DateHelper.

Function summary

distance_of_time_in_words($from_time, $to_time = null, $include_seconds = false)
format_date($date, $format = &apos;d&apos;, $culture = null, $charset = null)
format_daterange($start_date, $end_date, $format = &apos;d&apos;, $full_text, $start_text, $end_text, $culture = nul
format_datetime($date, $format = &apos;F&apos;, $culture = null, $charset = null)
time_ago_in_words($from_time, $include_seconds = false)

Function details

- **distance_of_time_in_words ($from_time, $to_time = null, $include_seconds = false)**
- **format_date ($date, $format = &apos;d&apos;, $culture = null, $charset = null)**
- **format_daterange ($start_date, $end_date, $format = &apos;d&apos;, $full_text, $start_text, $end_text, $culture = null, $charset = null)**
- **format_datetime ($date, $format = &apos;F&apos;, $culture = null, $charset = null)**
- **time_ago_in_words ($from_time, $include_seconds = false)**

[more]

Functions in EscapingHelper

The functions are primarily used by the output escaping component.

Each function specifies a way for applying a transformation to a string passed to it. The purpose is for the string to be "escaped" so it is suitable for the format it is being displayed in.

For example, the string: "It&apos;s required that you enter a username & password.\n" If this were to be displayed as HTML it would be sensible to turn the ampersand into &apos;&amp;&apos; and the apostrophe into &apos;&aps;&apos;. However if it were going to be used as a string in JavaScript to be displayed in an alert box it would be right to leave the string as-is, but c-escape the apostrophe and the new line.

For each function there is a define to avoid problems with strings being incorrectly specified.

Function summary

string esc_entities($value)	Runs the PHP function htmlentities on the value passed.
string esc_js($value)	A function that c-escapes a string after applying . The assumption is that the value will be used to generate dynamic HTML in some way and the safest way to prevent mishap is to assume the value should have HTML entities set properly.
string esc_js_no_entities($value)	A function the c-escapes a string, making it suitable to be placed in a JavaScript string.
string esc_raw($value)	An identity function that merely returns that which it is given, the purpose being to be able to specify that the value is not to be escaped in any way.

Function details

- ### (string) esc_entities ($value)

  $value the value to escape
  Runs the PHP function htmlentities on the value passed.

  returns the escaped value
- ### (string) esc_js ($value)

  $value the value to escape
  A function that c-escapes a string after applying . The assumption is that the value will be used to generate dynamic HTML in some way and the safest way to prevent mishap is to assume the value should have HTML entities set properly.

The esc_js_no_entities method should be used to escape a string that is ultimately not going to end up as text in an HTML document.

returns the escaped value

- ## (string) esc_js_no_entities ($value)

  $value the value to escape

  A function the c-escapes a string, making it suitable to be placed in a JavaScript string.

  returns the escaped value

- ## (string) esc_raw ($value)

  $value the value to escape

  An identity function that merely returns that which it is given, the purpose being to be able to specify that the value is not to be escaped in any way.

  returns the escaped value

[more]

Functions in FormHelper

FormHelper.

Function summary

	_convert_include_custom_for_select($options, [[[amp]]]$select_options)	
array	_convert_options($options)	Converts specific to their correct HTML format
string	checkbox_tag($name, $value = &apos;1&apos;, $checked = false, $options = array())	Returns an XHTML compliant <input tag with type="checkbox"
string	form_tag($url_for_options = &apos;&apos;, $options = array())	Returns an HTML <form> tag that points to a valid action, route or URL as defined by .
string	get_id_from_name($name, $value = null)	Returns a formatted ID based on the parameter and optionally the parameter.
string	input_date_range_tag($name, $value, $options = array())	Returns two XHTML compliant <input tags to be used as a free-text date fields for a date range.
string	input_date_tag($name, $value = null, $options = array())	Returns an XHTML compliant <input tag to be used as a free-text date field.
string	input_file_tag($name, $options = array())	Returns an XHTML compliant <input tag with

		type="file".
string	input_hidden_tag($name, $value = null, $options = array())	Returns an XHTML compliant <input tag with type="hidden".
string	input_password_tag($name = &apos;password&apos;, $value = null, $options = array())	Returns an XHTML compliant <input tag with type="password".
string	input_tag($name, $value = null, $options = array())	Returns an XHTML compliant <input tag with type="text".
string	label_for($id, $label, $options = array())	Returns a <label> tag with for the specified parameter.
string	options_for_select($options = array(), $selected = &apos;&apos;, $html_options = array())	Returns a formatted set of <option> tags based on optional array variable.
string	radiobutton_tag($name, $value, $checked = false, $options = array())	Returns an XHTML compliant <input tag with type="radio".
string	reset_tag($value = &apos;Reset&apos;, $options = array())	Returns an XHTML compliant <input tag with type="reset".
string	select_country_tag($name, $selected = null, $options = array())	Returns a <select tag populated wit all the countries i the world.
string	select_language_tag($name, $selected = null, $options = array())	Returns a <select tag populated wit all the languages in the world (or almost).
string	select_tag($name, $option_tags = null, $options = array())	Returns a <select tag, optionally

		comprised of <option> tags.
string	submit_image_tag($source, $options = array())	Returns an XHTML compliant <input tag with type="image".
string	submit_tag($value = &apos;Save changes&apos;, $options = array())	Returns an XHTML compliant <input tag with type="submit".
string	textarea_tag($name, $content = null, $options = array())	Returns a <textarea> tag, optionally wrapped with an inline rich-text JavaScript editor.

Function details

- ### _convert_include_custom_for_select ($options, [[[amp]]]$select_options)
- ### (array) _convert_options ($options)

$options	options

  Converts specific to their correct HTML format

  returns returns properly formatted options
- ### (string) checkbox_tag ($name, $value = &apos;1&apos;, $checked = false, $options = array())

$name	field name
$value	checkbox value (if checked)
$checked	is the checkbox checked? (1 or 0)
$options	additional HTML compliant <input> tag parameters

  Returns an XHTML compliant <input> tag with type="checkbox".

  When creating multiple checkboxes with the same name, be sure to use an array for the $name parameter (i.e. &apos;name[]&apos;). The checkbox_tag is smart enough to create unique ID&apos;s based on the $value parameter like so:

<input type="checkbox" name="status[]" id="status_3" value="3" /> <input type="checkbox" name="status[]" id="status_4" value="4" />

Examples:

```
get('newsletter'));]]>

 'style_a'));]]>


 "Element.toggle('tos'); return false;"));]]>
```

returns XHTML compliant <input> tag with type="checkbox"

- ## (string) form_tag ($url_for_options = &apos;&apos;, $options = array())

$url_for_options	valid action, route or URL
$options	optional HTML parameters for the <form> tag

Returns an HTML <form> tag that points to a valid action, route or URL as defined by .

By default, the form tag is generated in POST format, but can easily be configured along with any additional HTML parameters via the optional $options parameter. If you are using file uploads, be sure to set the multipart option to true.

Options:

♦ multipart - When set to true, enctype is set to "multipart/form-data".

Examples:

```
]]>

 'myformname', 'multipart' => true)); ?>]]>
```

returns opening HTML <form> tag with options

- ## (string) get_id_from_name ($name, $value = null)

$name	field name
$value	field value

Returns a formatted ID based on the parameter and optionally the parameter.

This function determines the proper form field ID name based on the parameters. If a form field has an array value as a name we need to convert them to proper and unique IDs like so: name[] => name (if value == null) name[] => name_value (if value != null) name[bob] => name_bob name[item][total]

=> name_item_total

Examples:

returns <select> tag populated with all the languages in the world.

-
(string) input_date_range_tag ($name, $value, $options = array())

$name	field name
$value	dates: $value&[apos;from&apos;] and $value&[apos;to&apos;]
$options	additional HTML compliant <input> tag parameters

Returns two XHTML compliant <input> tags to be used as a free-text date fields for a date range.

Built on the input_date_tag, the input_date_range_tag combines two input tags that allow the user to specify a from and to date. You can easily implement a JavaScript calendar by enabling the &apos;rich&apos; option in the $options parameter. This includes a button next to the field that when clicked, will open an inline JavaScript calendar. When a date is selected, it will automatically populate the <input> tag with the proper date, formatted to the user&apos;s culture setting.

Note: The $name parameter will automatically converted to array names. For example, a $name of "date" becomes date[from] and date[to]

Options:

- ♦ rich - If set to true, includes an inline JavaScript calendar can auto-populate the date field with the chosen date
- ♦ before - string to be displayed before the input_date_range_tag
- ♦ middle - string to be displayed between the from and to tags
- ♦ after - string to be displayed after the input_date_range_tag

Examples:

```
'2006-05-15', 'to' => '2006-06-15');
 echo input_date_range_tag('date', $date, array('rich' => true));]]>

' through ', 'rich' => true));]]>
```

returns XHTML compliant <input> tag with optional JS calendar integration

-
(string) input_date_tag ($name, $value = null, $options = array())

$name	field name
$value	date
$options	

	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag to be used as a free-text date field.

You can easily implement a JavaScript calendar by enabling the &apos;rich&apos; option in the $options parameter. This includes a button next to the field that when clicked, will open an inline JavaScript calendar. When a date is selected, it will automatically populate the <input> tag with the proper date, formatted to the user&apos;s culture setting. Symfony also conveniently offers the input_date_range_tag, that allows you to specify a to and from date.

Options:

- ♦ rich - If set to true, includes an inline JavaScript calendar can auto-populate the date field with the chosen date

Examples:

```
true));]]>
```

returns XHTML compliant <input> tag with optional JS calendar integration

•

(string) input_file_tag ($name, $options = array())

$name	field name
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="file".

Similar to the input_tag helper, the input_hidden_tag helper generates your basic XHTML <input> tag and can utilize any standard <input> tag parameters passed in the optional $options parameter. The only difference is that it creates the tag with type="file", meaning that next to the field will be a "browse" (or similar) button. This gives the user the ability to choose a file from there computer to upload to the web server. Remember, if you plan to upload files to your website, be sure to set the multipart option form_tag helper function to true or your files will not be properly uploaded to the web server.

Examples:

```
30));]]>
```

returns XHTML compliant <input> tag with type="file"

•

(string) input_hidden_tag ($name, $value = null, $options = array())

$name	field name
$value	populated field value
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="hidden".

Similar to the input_tag helper, the input_hidden_tag helper generates an XHTML <input> tag and can utilize any standard <input> tag parameters passed in the optional $options parameter. The only difference is that it creates the tag with type="hidden", meaning that is not visible on the page.

Examples:

returns XHTML compliant <input> tag with type="hidden"

•

(string) input_password_tag ($name = &apos;password&apos;, $value = null, $options = array())

$name	field name
$value	populated field value
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="password".

Similar to the input_tag helper, the input_hidden_tag helper generates your basic XHTML <input> tag and can utilize any standard <input> tag parameters passed in the optional $options parameter. The only difference is that it creates the tag with type="password", meaning that the text entered into this field will not be visible to the end user. In most cases it is replaced by * * * * * * * *. Even though this text is not readable, it is recommended that you do not populate the optional $value option with a plain-text password or any other sensitive information, as this is a potential security risk.

Examples:

returns XHTML compliant <input> tag with type="password"

•

(string) input_tag ($name, $value = null, $options = array())

$name	field name
$value	selected field value
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="text".

The input_tag helper generates your basic XHTML <input> tag and can utilize any standard <input> tag parameters passed in the optional $options parameter.

Examples:

```
get('amount'), array('size' => 8, 'maxlength' => 8));]]>
```

returns XHTML compliant <input> tag with type="text"

-

(string) label_for ($id, $label, $options = array())

$id	id
$label	label or title
$options	additional HTML compliant <label> tag parameters

Returns a <label> tag with for the specified parameter.

returns <label> tag with $label for the specified $id parameter.

-

(string) options_for_select ($options = array(), $selected = &apos;&apos;, $html_options = array())

$options	dataset to create <option> tags and <optgroup> tags from
$selected	selected option value
$html_options	additional HTML compliant <option> tag parameters

Returns a formatted set of <option> tags based on optional array variable.

The options_for_select helper is usually called in conjunction with the select_tag helper, as it is relatively useless on its own. By passing an array of $options, the helper will automatically generate <option> tags using the array key as the value and the array value as the display title. Additionally the options_for_select tag is smart enough to detect nested arrays as <optgroup> tags. If the helper detects that the array value is an array itself, it creates an <optgroup> tag with the name of the group being the key and the contents of the <optgroup> being the array.

Options:

- include_blank - Includes a blank <option> tag at the beginning of the string with an empty value
- include_custom - Includes an <option> tag with a custom display title at the beginning of the string with an empty value

Examples:

```
'Larry', 2 => 'Moe', 3 => 'Curly')));]]>

'Visa', 'MAST' => 'MasterCard', 'AMEX' => 'American Express', 'DISC' => 'Discover');
 echo select_tag('cc_type', options_for_select($card_list, 'AMEX', array('include_custom'

'Joe', 2 => 'Sue', 'Group A' => array(3 => 'Mary', 4 => 'Tom'), 'Group B' => array(5 => '
 echo select_tag('employee', options_for_select($optgroup_array, null, array('include_bla
```

returns populated with <option> tags derived from the $options array variable

- ## (string) radiobutton_tag ($name, $value, $checked = false, $options = array())

$name	field name
$value	radio button value (if selected)
$checked	is the radio button selected? (1 or 0)
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="radio".

Examples:

returns XHTML compliant <input> tag with type="radio"

- ## (string) reset_tag ($value = &apos;Reset&apos;, $options = array())

$value	field value (title of reset button)
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="reset".

By default, this helper creates a submit tag with a name of <em>reset</em>. Also, the default $value parameter (title of the button) is set to "Reset" which can be easily overwritten by passing a $value parameter.

Examples:

returns XHTML compliant <input> tag with type="reset"

- ## (string) select_country_tag ($name, $selected = null, $options = array())

$name	field name
$selected	selected field value (two-character country code)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with all the countries in the world.

The select_country_tag builds off the traditional select_tag function, and is conveniently populated with all the countries in the world (sorted alphabetically). Each option in the list has a two-character country code for its value and the country&apos;s name as its display title. The country data is retrieved via the sfCultureInfo class, which stores a wide variety of i18n and i10n settings for various countries and cultures throughout the world. Here&apos;s an example of an <option> tag generated by the select_country_tag:

<option value="US">United States</option>

Examples:

returns <select> tag populated with all the countries in the world.

-

(string) select_language_tag ($name, $selected = null, $options = array())

$name	field name
$selected	selected field value (two or threecharacter language/culture code)
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag populated with all the languages in the world (or almost).

The select_language_tag builds off the traditional select_tag function, and is conveniently populated with all the languages in the world (sorted alphabetically). Each option in the list has a two or three character language/culture code for its value and the language&apos;s name as its display title. The country data is retrieved via the sfCultureInfo class, which stores a wide variety of i18n and i10n settings for various countries and cultures throughout the world. Here&apos;s an example of an <option> tag generated by the select_country_tag:

<option value="en">English</option>

Examples:

returns <select> tag populated with all the languages in the world.

-

(string) select_tag ($name, $option_tags = null, $options = array())

$name	field name
$option_tags	contains a string of valid <option></option> tags, or an array of options that will be passed to options_for_select
$options	additional HTML compliant <select> tag parameters

Returns a <select> tag, optionally comprised of <option> tags.

The select tag does not generate <option> tags by default. To do so, you must populate the $option_tags parameter with a string of valid HTML compliant <option> tags. Fortunately, Symfony provides a handy helper function to convert an array of data into option tags (see options_for_select).

If you need to create a "multiple" select tag (ability to select multiple options), set the multiple option to true. Doing so will automatically convert the name field to an array type variable (i.e. name="name" becomes name="name[]").

Options:

- ♦ multiple - If set to true, the select tag will allow multiple options to be selected at once.

Examples:

```
'Larry', 2 => 'Moe', 3 => 'Curly');
 echo select_tag('person', options_for_select($person_list, $sf_params->get('person')), a

true));]]>

'Javascript:this.form.submit();'));]]>
```

returns <select> tag optionally comprised of <option> tags.

- ## (string) submit_image_tag ($source, $options = array())

$source	path to image file
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="image".

The submit_image_tag is very similar to the submit_tag, the only difference being that it uses an image for the submit button instead of the browser-generated default button. The image is defined by the $source parameter and must be a valid image, either local or remote (URL). By default, this helper creates a submit tag with a name of <em>commit</em> to avoid conflicts with other parts of the framework. It is recommended that you do not use the name "submit" for submit tags unless absolutely necessary.

Examples:

returns XHTML compliant <input> tag with type="image"

- ## (string) submit_tag ($value = &apos;Save changes&apos;, $options = array())

$value	field value (title of submit button)
$options	additional HTML compliant <input> tag parameters

Returns an XHTML compliant <input> tag with type="submit".

By default, this helper creates a submit tag with a name of <em>commit</em> to avoid conflicts with other parts of the framework. It is recommended that you do not use the name "submit" for submit tags unless absolutely necessary. Also, the default $value parameter (title of the button) is set to "Save changes", which can be easily overwritten by passing a $value parameter.

Examples:

returns XHTML compliant <input> tag with type="submit"

-

(string) textarea_tag ($name, $content = null, $options = array())

$name	field name
$content	populated field value
$options	additional HTML compliant <textarea> tag parameters

Returns a <textarea> tag, optionally wrapped with an inline rich-text JavaScript editor.

The texarea_tag helper generates a standard HTML <textarea> tag and can be manipulated with any number of standard HTML parameters via the $options array variable. However, the textarea tag also has the unique capability of being transformed into a WYSIWYG rich-text editor such as TinyMCE (http://tinymce.moxiecode.com) very easily with the use of some specific options:

Options:

- rich: A rich text editor class (for example sfRichTextEditorTinyMCE for TinyMCE).

Examples:

```
10, 'cols' => 50));]]>
```

returns <textarea> tag optionally wrapped with a rich-text WYSIWYG editor

[more]

Functions in HelperHelper

HelperHelper.

Function summary

use_helper()

Function details

- ### use_helper ()

[more]

Functions in I18NHelper

I18NHelper.

Function summary

__($text, $args = array(), $catalogue = &apos;messages&apos;)
format_country($country_iso, $culture = null)
format_language($language_iso, $culture = null)
format_number_choice($text, $args = array(), $number, $catalogue = &apos;messages&apos;)

Function details

- ### __ ($text, $args = array(), $catalogue = &apos;messages&apos;)
- ### format_country ($country_iso, $culture = null)
- ### format_language ($language_iso, $culture = null)
- ### format_number_choice ($text, $args = array(), $number, $catalogue = &apos;messages&apos;)

[more]

Functions in JavascriptHelper

JavascriptHelper.

Function summary

	_array_or_string_for_javascript($option)	
string	_auto_complete_field($field_id, $url, $options = array())	wrapp Ajax.A
	_build_callbacks($options)	
	_build_observer($klass, $name, $options = array())	
	_in_place_editor($field_id, $url, $options = array())	
	_method_option_to_s($method)	
	_options_for_ajax($options)	
	_options_for_javascript($options)	
	button_to_function($name, $function, $html_options = array())	Return javasc handle
	button_to_remote($name, $options = array(), $html_options = array())	Return define &apos that&a using
	draggable_element($element_id, $options = array())	Makes specifi dragga
	drop_receiving_element($element_id, $options = array())	Makes specifi receiv (create &apos make a
	end_if_javascript()	Mark t be sho switch
	evaluate_remote_response()	Return &apos which &apos in &ap multip &apos

		calls.
	form_remote_tag($options = array(), $options_html = array())	Return XMLH instead arrang JavaSc the for regula receivi &apos
	get_ajax_options()	
	get_callbacks()	
	if_javascript()	Mark t be sho switch
string	input_auto_complete_tag($name, $value, $url, $tag_options = array(), $completion_options = array())	wrapp Ajax.A
string	input_in_place_editor_tag($name, $url, $editor_options = array())	wrapp Ajax.I
	javascript_cdata_section($content)	
	javascript_tag($content)	Return &apos
	link_to_function($name, $function, $html_options = array())	Return javasc handle
	link_to_remote($name, $options = array(), $html_options = array())	Return by &a &apos that&a using reques object &apos
	observe_field($field_id, $options = array())	Observ specifi makes have c
	observe_form($form_id, $options = array())	Like & operat the DO &apos &apos defaul option string)

periodically_call_remote($options = array())	Period (&apo &apos is 10).
remote_function($options)	Return functio
sortable_element($element_id, $options = array())	Makes specifi sortabl AJAX change the ser param
submit_to_remote($name, $value, $options = array(), $options_html = array())	Return form u backgr POST &apos same a &apos
update_element_function($element_id, $options = array())	Return expres elemen accord passed
visual_effect($name, $element_id = false, $js_options = array())	Return the AJ effects

Function details

- ### _array_or_string_for_javascript ($option)
- ### (string) _auto_complete_field ($field_id, $url, $options = array())

$field_id	id value of input field
$url	url of module/action to execute for autocompletion
$options	completion options

wrapper for script.aculo.us/prototype Ajax.Autocompleter.

returns javascript tag for Ajax.Autocompleter

- ### _build_callbacks ($options)
- ### _build_observer ($klass, $name, $options = array())
- ### _in_place_editor ($field_id, $url, $options = array())
- ### _method_option_to_s ($method)
- ### _options_for_ajax ($options)
- ### _options_for_javascript ($options)
- ### button_to_function ($name, $function, $html_options = array())

  Returns a button that&apos;ll trigger a javascript function using the onclick handler and return false after the fact.

  Examples: <?php echo button_to_function(&apos;Greeting&apos;, "alert(&apos;Hello world!&apos;)") ?>
- ### button_to_remote ($name, $options = array(), $html_options = array())

  Returns an html button to a remote action defined by &apos;url&apos; (using the &apos;url_for()&apos; format) that&apos;s called in the background using XMLHttpRequest.

  See link_to_remote() for details.
- ### draggable_element ($element_id, $options = array())

  Makes the element with the DOM ID specified by &apos;$element_id&apos; draggable.

  Example: <?php echo draggable_element(&apos;my_image&apos;, array(&apos;revert&apos; => true,)) ?>

  You can change the behaviour with various options, see http://script.aculo.us for more documentation.
- ### drop_receiving_element ($element_id, $options = array())

  Makes the element with the DOM ID specified by &apos;$element_id&apos; receive dropped draggable elements (created by &apos;draggable_element()&apos;) and make an AJAX call.

  By default, the action called gets the DOM ID of the element as parameter.

  Example: <?php drop_receiving_element(&apos;my_cart&apos;, array(&apos;url&apos; => &apos;cart/add&apos;,)) ?>

  You can change the behaviour with various options, see http://script.aculo.us for more documentation.

- ## end_if_javascript ()
  Mark the end of a block that should only be shown in the browser if JavaScript is switched on.
- ## evaluate_remote_response ()
  Returns &apos;eval(request.responseText)&apos;, which is the Javascript function that &apos;form_remote_tag()&apos; can call in &apos;complete&apos; to evaluate a multiple update return document using &apos;update_element_function()&apos; calls.
- ## form_remote_tag ($options = array(), $options_html = array())
  Returns a form tag that will submit using XMLHttpRequest in the background instead of the regular reloading POST arrangement. Even though it&apos;s using JavaScript to serialize the form elements, the form submission will work just like a regular submission as viewed by the receiving side (all elements available in &apos;params&apos;).

  The options for specifying the target with &apos;url&apos; and defining callbacks are the same as &apos;link_to_remote()&apos;.

  A "fall-through" target for browsers that don&apos;t do JavaScript can be specified with the &apos;action&apos;/&apos;method&apos; options on &apos;$options_html&apos;

  Example: <?php echo form_remote_tag(array(&apos;url&apos; => &apos;@tag_add&apos;, &apos;update&apos; => &apos;question_tags&apos;, &apos;loading&apos; => "Element.show(&apos;indicator&apos;); \$(&apos;tag&apos;).value = &apos;&apos;", &apos;complete&apos; => "Element.hide(&apos;indicator&apos;);".visual_effect(&apos;highlight&apos;, &apos;question_tags&apos;),)) ?>

  The hash passed as a second argument is equivalent to the options (2nd) argument in the form_tag() helper.

  By default the fall-through action is the same as the one specified in the &apos;url&apos; (and the default method is &apos;post&apos;).
- ## get_ajax_options ()
- ## get_callbacks ()
- ## if_javascript ()
  Mark the start of a block that should only be shown in the browser if JavaScript is switched on.
- ## (string) input_auto_complete_tag ($name, $value, $url, $tag_options = array(), $completion_options = array())

$name	name value of input field
$value	default value for input field
$url	

	input tag options. (size, autocomplete, etc...)
$tag_options	completion options. (use_style, etc...)

wrapper for script.aculo.us/prototype Ajax.Autocompleter.

returns input field tag, div for completion results, and auto complete javascript tags

- ## (string) input_in_place_editor_tag ($name, $url, $editor_options = array())

$name	name id of field that can be edited
$url	url of module/action to be called when ok is clicked
$editor_options	editor tag options. (rows, cols, highlightcolor, highlightendcolor, etc...)

wrapper for script.aculo.us/prototype Ajax.InPlaceEditor.

returns javascript to manipulate the id field to allow click and edit functionality

- ## javascript_cdata_section ($content)

- ## javascript_tag ($content)

Returns a JavaScript tag with the &apos;$content&apos; inside.

Example: <?php echo javascript_tag("alert(&apos;All is good&apos;)") ?> => <script type="text/javascript">alert(&apos;All is good&apos;)</script>

- ## link_to_function ($name, $function, $html_options = array())

Returns a link that&apos;ll trigger a javascript function using the onclick handler and return false after the fact.

Examples: <?php echo link_to_function(&apos;Greeting&apos;, "alert(&apos;Hello world!&apos;)") ?> <?php echo link_to_function(image_tag(&apos;delete&apos;), "if confirm(&apos;Really?&apos;){ do_delete(); }") ?>

- ## link_to_remote ($name, $options = array(), $html_options = array())

Returns a link to a remote action defined by &apos;url&apos; (using the &apos;url_for()&apos; format) that&apos;s called in the background using XMLHttpRequest. The result of that request can then be inserted into a DOM object whose id can be specified with &apos;update&apos;.

Usually, the result would be a partial prepared by the controller with either &apos;render_partial()&apos;.

Examples: <?php echo link_to_remote(&apos;Delete this post&apos;), array(&apos;update&apos; => &apos;posts&apos;, &apos;url&apos; => &apos;destroy?id=&apos;.$post.id,)) ?> <?php echo link_to_remote(image_tag(&apos;refresh&apos;), array(&apos;update&apos; => &apos;emails&apos;, &apos;url&apos; => &apos;@list_emails&apos;,)) ?>

You can also specify a hash for &apos;update&apos; to allow for easy redirection of output to an other DOM element if a server-side error occurs:

Example: <?php echo link_to_remote(&apos;Delete this post&apos;, array(&apos;update&apos; => array(&apos;success&apos; => &apos;posts&apos;, &apos;failure&apos; => &apos;error&apos;), &apos;url&apos; => &apos;destroy?id=&apos;.$post.id,)) ?>

Optionally, you can use the &apos;position&apos; parameter to influence how the target DOM element is updated. It must be one of &apos;before&apos;, &apos;top&apos;, &apos;bottom&apos;, or &apos;after&apos;.

By default, these remote requests are processed asynchronous during which various JavaScript callbacks can be triggered (for progress indicators and the likes). All callbacks get access to the &apos;request&apos; object, which holds the underlying XMLHttpRequest.

To access the server response, use &apos;request.responseText&apos;, to find out the HTTP status, use &apos;request.status&apos;.

Example: <?php echo link_to_remote($word, array(&apos;url&apos; => &apos;@undo?n=&apos;.$word_counter, &apos;complete&apos; => &apos;undoRequestCompleted(request)&apos;)) ?>

The callbacks that may be specified are (in order):

&apos;loading&apos; Called when the remote document is being loaded with data by the browser. &apos;loaded&apos; Called when the browser has finished loading the remote document. &apos;interactive&apos; Called when the user can interact with the remote document, even though it has not finished loading. &apos;success&apos; Called when the XMLHttpRequest is completed, and the HTTP status code is in the 2XX range. &apos;failure&apos; Called when the XMLHttpRequest is completed, and the HTTP status code is not in the 2XX range. &apos;complete&apos; Called when the XMLHttpRequest is complete (fires after success/failure if they are present).,

You can further refine &apos;success&apos; and &apos;failure&apos; by adding additional callbacks for specific status codes:

Example: <?php echo link_to_remote($word, array(&apos;url&apos; => &apos;@rule&apos;, &apos;404&apos; => "alert(&apos;Not found...? Wrong URL...?&apos;)", &apos;failure&apos; => "alert(&apos;HTTP Error &apos; + request.status + &apos;!&apos;)",)) ?>

A status code callback overrides the success/failure handlers if present.

If you for some reason or another need synchronous processing (that&apos;ll block the browser while the request is happening), you can specify &apos;type&apos; => &apos;synchronous&apos;.

You can customize further browser side call logic by passing in JavaScript code snippets via some optional parameters. In their order of use these are:

&apos;confirm&apos; Adds confirmation dialog. &apos;condition&apos; Perform remote request conditionally by this expression. Use this to describe browser-side conditions when request should not be initiated. &apos;before&apos; Called before request is initiated. &apos;after&apos; Called immediately after request was initiated and before &apos;loading&apos;. &apos;submit&apos; Specifies the DOM element ID that&apos;s used as the parent of the form elements. By default this is the current form, but it could just as well be the ID of a table row or any other DOM element.

- ## observe_field ($field_id, $options = array())

  Observes the field with the DOM ID specified by &apos;$field_id&apos; and makes an AJAX call when its contents have changed.

  Required &apos;$options&apos; are: &apos;url&apos; &apos;url_for()&apos;-style options for the action to call when the field has changed.

  Additional options are: &apos;frequency&apos; The frequency (in seconds) at which changes to this field will be detected. Not setting this option at all or to a value equal to or less than zero will use event based observation instead of time based observation. &apos;update&apos; Specifies the DOM ID of the element whose innerHTML should be updated with the XMLHttpRequest response text. &apos;with&apos; A JavaScript expression specifying the parameters for the XMLHttpRequest. This defaults to &apos;value&apos;, which in the evaluated context refers to the new field value.

  Additionally, you may specify any of the options documented in link_to_remote().

- ## observe_form ($form_id, $options = array())

  Like &apos;observe_field()&apos;, but operates on an entire form identified by the DOM ID &apos;$form_id&apos;. &apos;$options&apos; are the same as &apos;observe_field()&apos;, except the default value of the &apos;with&apos; option evaluates to the serialized (request string) value of the form.

- ## periodically_call_remote ($options = array())

  Periodically calls the specified url (&apos;url&apos;) every &apos;frequency&apos; seconds (default is 10).

  Usually used to update a specified div (&apos;update&apos;) with the results of the remote call. The options for specifying the target with &apos;url&apos; and defining callbacks is the same as &apos;link_to_remote()&apos;.

- ## remote_function ($options)

  Returns the javascript needed for a remote function.

  Takes the same arguments as &apos;link_to_remote()&apos;.

  Example: <select id="options" onchange="<?php echo remote_function(array(&apos;update&apos; => &apos;options&apos;, &apos;url&apos; => &apos;@update_options&apos;)) ?>"> <option value="0">Hello</option> <option value="1">World</option> </select>

- ## sortable_element ($element_id, $options = array())

  Makes the elements with the DOM ID specified by &apos;$element_id&apos; sortable by drag-and-drop and make an AJAX call whenever the sort order has changed. By default, the action called gets the serialized sortable element as parameters.

  Example: <php echo sortable_element($my_list, array(&apos;url&apos; => &apos;@order&apos;,)) ?>

  In the example, the action gets a &apos;$my_list&apos; array parameter containing the values of the ids of elements the sortable consists of, in the current order.

  You can change the behaviour with various options, see http://script.aculo.us for more documentation.

- ## submit_to_remote ($name, $value, $options = array(), $options_html = array())

  Returns a button input tag that will submit form using XMLHttpRequest in the background instead of regular reloading POST arrangement. The &apos;$options&apos; argument is the same as in &apos;form_remote_tag()&apos;.

- ## update_element_function ($element_id, $options = array())

  Returns a Javascript function (or expression) that will update a DOM element &apos;$element_id&apos; according to the &apos;$options&apos; passed.

  Possible &apos;$options&apos; are: &apos;content&apos; The content to use for updating. Can be left out if using block, see example. &apos;action&apos; Valid options are &apos;update&apos; (assumed by default), &apos;empty&apos;, &apos;remove&apos; &apos;position&apos; If the &apos;action&apos; is &apos;update&apos;, you can optionally specify one of the following positions: &apos;before&apos;, &apos;top&apos;, &apos;bottom&apos;, &apos;after&apos;.

  Example: <?php echo javascript_tag(update_element_function(&apos;products&apos;, array(&apos;position&apos; => &apos;bottom&apos;, &apos;content&apos; => "<p>New product!</p>",))) ?>

  This method can also be used in combination with remote method call where the result is evaluated afterwards to cause multiple updates on a page.

  Example:

  - ♦ Calling view

  <?php echo form_remote_tag(array(&apos;url&apos; => &apos;@buy&apos;, &apos;complete&apos; => evaluate_remote_response())) ?> all the inputs here...

  - ♦ Target action

  public function executeBuy() { $this->product = ProductPeer::retrieveByPk(1); }

  - ♦ Returning view

  <php echo update_element_function(&apos;cart&apos;, array(&apos;action&apos; =>

&apos;update&apos;, &apos;position&apos; => &apos;bottom&apos;, &apos;content&apos; => &apos;<p>New Product: &apos;.$product->getName().&apos;</p>&apos;,)) ?>

- ### visual_effect ($name, $element_id = false, $js_options = array())
  Returns a JavaScript snippet to be used on the AJAX callbacks for starting visual effects.

  Example: <?php echo link_to_remote(&apos;Reload&apos;, array(&apos;update&apos; => &apos;posts&apos;, &apos;url&apos; => &apos;@reload&apos;, &apos;complete => visual_effect(&apos;highlight&apos;, &apos;posts&apos;, array(&apos;duration&apos; => 0.5)),)) ?>

  If no &apos;$element_id&apos; is given, it assumes "element" which should be a local variable in the generated JavaScript execution context. This can be used for example with drop_receiving_element():

  <?php echo drop_receving_element(..., array(... &apos;loading&apos; => visual_effect(&apos;fade&apos;),)) ?>

  This would fade the element that was dropped on the drop receiving element.

  You can change the behaviour with various options, see http://script.aculo.us for more documentation.

[more]

Functions in NumberHelper

NumberHelper.

Function summary

_current_language($culture)
format_currency($amount, $currency = null, $culture = null)
format_number($number, $culture = null)

Function details

- ### _current_language ($culture)
- ### format_currency ($amount, $currency = null, $culture = null)
- ### format_number ($number, $culture = null)

[more]

Functions in ObjectAdminHelper

ObjectHelper for admin generator.

Function summary

_get_object_list($object, $method, $options, $callback)
_get_propel_object_list($object, $method, $options)
object_admin_check_list($object, $method, $options = array(), $callback = null)
object_admin_double_list($object, $method, $options = array(), $callback = null)
object_admin_input_file_tag($object, $method, $options = array())
object_admin_select_list($object, $method, $options = array(), $callback = null)

Function details

- ### _get_object_list ($object, $method, $options, $callback)
- ### _get_propel_object_list ($object, $method, $options)
- ### object_admin_check_list ($object, $method, $options = array(), $callback = null)
- ### object_admin_double_list ($object, $method, $options = array(), $callback = null)
- ### object_admin_input_file_tag ($object, $method, $options = array())
- ### object_admin_select_list ($object, $method, $options = array(), $callback = null)

[more]

Functions in ObjectHelper

ObjectHelper.

Function summary

	_convert_method_to_name($method, [[[amp]]]$options)
string	_get_class_decorated((object An) $object)
	_get_object_value($object, $method, $default_value = null, $param = null)
	_get_options_from_objects($objects, $text_method = null)
string	object_checkbox_tag((object An) $object, $method, $options = array(), $default_value = null)
string	object_input_date_tag((object An) $object, $method, $options = array(), $default_value = null)
string	object_input_hidden_tag((object An) $object, $method, $options = array(), $default_value = null)
string	object_input_tag((object An) $object, $method, $options = array(), $default_value = null)
	object_select_country_tag($object, $method, $options = array(), $default_value = null)
	object_select_language_tag($object, $method, $options = array(), $default_value = null)
string	object_select_tag((object An) $object, $method, $options = array(), $default_value = null)
string	object_textarea_tag((object An) $object, $method, $options = array(), $default_value = null)
	objects_for_select($options = array(), $value_method, $text_method = null, $selected = null, $html_options = a

Function details

- ### _convert_method_to_name ($method, [[[amp]]]$options)
- ### (string) _get_class_decorated ((object An) $object)

$object	object that might be wrapped in an sfOutputEscaperObjectDecorator(-derivative)

Returns the name of the class of an decorated object

returns The name of the class of the object being decorated for escaping, or the class of the object if it isn&apos;t decorated

- ### _get_object_value ($object, $method, $default_value = null, $param = null)
- ### _get_options_from_objects ($objects, $text_method = null)
- ### (string) object_checkbox_tag ((object An) $object, $method, $options = array(), $default_value = null)

$object	object.
$method	An object column.
$options	Checkbox options.
$default_value	Checkbox value.

Returns a checkbox html tag.

returns An html string which represents a checkbox tag.

- ### (string) object_input_date_tag ((object An) $object, $method, $options = array(), $default_value = null)

$object	object.

$method	An object column.
$options	Date options.
$default_value	Date default value.

Returns a html date control.

returns An html string which represents a date control.

- ### (string) object_input_hidden_tag ((object An) $object, $method, $options = array(), $default_value = null)

$object	object.
$method	An object column.
$options	Input options.
$default_value	Input default value.

Returns a hidden input html tag.

returns An html string which represents a hidden input tag.

- ### (string) object_input_tag ((object An) $object, $method, $options = array(), $default_value = null)

$object	object.
$method	An object column.
$options	Input options.
$default_value	Input default value.

Returns a input html tag.

returns An html string which represents an input tag.

- ### object_select_country_tag ($object, $method, $options = array(), $default_value = null)
- ### object_select_language_tag ($object, $method, $options = array(), $default_value = null)
- ### (string) object_select_tag ((object An) $object, $method, $options = array(), $default_value = null)

$object	object or the selected value
$method	An object column.

$options	Input options (related_class option is mandatory).
$default_value	Input default value.

Returns a list html tag.

returns A list string which represents an input tag.

- ### (string) object_textarea_tag ((object An) $object, $method, $options = array(), $default_value = null)

$object	object.
$method	An object column.
$options	Textarea options.
$default_value	Textarea default value.

Returns a textarea html tag.

returns An html string which represents a textarea tag.

- ### objects_for_select ($options = array(), $value_method, $text_method = null, $selected = null, $html_options = array())

Accepts a container of objects, the method name to use for the value, and the method name to use for the display.

It returns a string of option tags.

NOTE: Only the option tags are returned, you have to wrap this call in a regular HTML select tag.

[more]

Functions in PartialHelper

PartialHelper.

Function summary

	_get_cache($cacheManager, $uri)	
	_set_cache($cacheManager, $uri, $retval)	
	end_slot()	Stops the content capture and save the content in the slot.
string	get_component($moduleName, $componentName, $vars = array())	Evaluates and returns a component.
string	get_component_slot($name, $vars = array())	Evaluates and returns a component slot.
string	get_partial($templateName, $vars = array())	Evaluates and returns a partial.
string	get_slot($name)	Evaluates and returns a slot.
boolean	has_slot($name)	Returns true if the slot exists.
	include_component($moduleName, $componentName, $vars = array())	Evaluates and echoes a component.
	include_component_slot($name, $vars = array())	Evaluates and echoes a component slot.
	include_partial($templateName, $vars = array())	Evaluates and echoes a partial.
	include_slot($name)	Evaluates and echoes a slot.
	slot($name)	Begins the capturing of the slot.

Function details

- ### _get_cache ($cacheManager, $uri)
- ### _set_cache ($cacheManager, $uri, $retval)

- ## end_slot ()

  Stops the content capture and save the content in the slot.

- ## (string) get_component ($moduleName, $componentName, $vars = array())

$moduleName	module name
$componentName	component name
$vars	variables to be made accessible to the component

  Evaluates and returns a component.

  The syntax is similar to the one of include_component.

  Example:

  ```
  12345));]]>
  ```

  returns result of the component execution

- ## (string) get_component_slot ($name, $vars = array())

$name	slot name
$vars	variables to be made accessible to the component

  Evaluates and returns a component slot.

  The syntax is similar to the one of include_component_slot.

  Example:

  ```
  12345));]]>
  ```

  returns result of the component execution

- ## (string) get_partial ($templateName, $vars = array())

$templateName	partial name
$vars	variables to be made accessible to the partial

  Evaluates and returns a partial.

  The syntax is similar to the one of include_partial

  Example:

  ```
  12345));]]>
  ```

  returns result of the partial execution

- ## (string) get_slot ($name)

$name	slot name

Evaluates and returns a slot.

Example:

returns content of the slot

- ## (boolean) has_slot ($name)

$name	slot name

Returns true if the slot exists.

returns true, if the slot exists

- ## include_component ($moduleName, $componentName, $vars = array())

$moduleName	module name
$componentName	component name
$vars	variables to be made accessible to the component

Evaluates and echoes a component.

For a variable to be accessible to the component and its partial, it has to be passed in the third argument.

Example:

```
12345));]]>
```

- ## include_component_slot ($name, $vars = array())

$name	slot name
$vars	variables to be made accessible to the component

Evaluates and echoes a component slot.

The component name is deduced from the definition of the view.yml For a variable to be accessible to the component and its partial, it has to be passed in the second argument.

Example:

```
12345));]]>
```

- ## include_partial ($templateName, $vars = array())

$templateName	partial name
$vars	variables to be made accessible to the partial

  Evaluates and echoes a partial.

  The partial name is composed as follows: &apos;mymodule/mypartial&apos;. The partial file name is _mypartial.php and is looked for in modules/mymodule/templates/. If the partial name doesn&apos;t include a module name, then the partial file is searched for in the caller&apos;s template/ directory. If the module name is &apos;global&apos;, then the partial file is looked for in myapp/templates/. For a variable to be accessible to the partial, it has to be passed in the second argument.

  Example:

  ```
  12345));]]>
  ```

- ## include_slot ($name)

  $name slot name

  Evaluates and echoes a slot.

  Example:

- ## slot ($name)

  $name slot name

  Begins the capturing of the slot.

[more]

Functions in sfRichTextEditor

sfRichTextEditor is an abstract class for rich text editor classes.

Direct sub-classes

- sfRichTextEditorFCK: sfRichTextEditorFCK implements the FCK rich text editor.
- sfRichTextEditorTinyMCE: sfRichTextEditorTinyMCE implements the TinyMCE rich text editor.

Function summary

	initialize($name, $content, $options = array())	Initializes this rich text editor.
string	toHTML()	Returns the rich text editor as HTML.

Function details

- ### initialize ($name, $content, $options = array())

$name	The tag name
$content	The rich text editor content
$options	An array of options

  Initializes this rich text editor.

- ### (string) toHTML ()

  Returns the rich text editor as HTML.

  returns Rich text editor HTML representation

[more]

Functions in sfRichTextEditorFCK

sfRichTextEditorFCK implements the FCK rich text editor.

Inheritence

sfRichTextEditorFCK < sfRichTextEditor

Function summary

string toHTML() Returns the rich text editor as HTML.

Methods inherited from sfRichTextEditor

initialize, toHTML

Function details

- ### (string) toHTML ()
  Returns the rich text editor as HTML.

  returns Rich text editor HTML representation

[more]

Functions in sfRichTextEditorTinyMCE

sfRichTextEditorTinyMCE implements the TinyMCE rich text editor.

Inheritence

sfRichTextEditorTinyMCE < sfRichTextEditor

Function summary

string	toHTML()	Returns the rich text editor as HTML.

Methods inherited from sfRichTextEditor

initialize, toHTML

Function details

- ### (string) toHTML ()
  Returns the rich text editor as HTML.

  returns Rich text editor HTML representation

[more]

Functions in TagHelper

TagHelper defines some base helpers to construct html tags.

Function summary

	_get_option([[[amp]]]$options, $name, $default = null)	
	_parse_attributes($string)	
	_tag_options($options = array())	
	cdata_section($content)	
	content_tag($name, $content = &apos;&apos;, $options = array())	
	escape_javascript($javascript = &apos;&apos;)	Escape carrier returns and single and double quotes for Javascript segments.
string	escape_once($html)	Escapes an HTML string.
string	fix_double_escape($escaped)	Fixes double escaped strings.
string	tag(($name) $name, ($options) $options = array(), ($open) $open = false)	Constructs an html tag.

Function details

- ### _get_option ([[[amp]]]$options, $name, $default = null)
- ### _parse_attributes ($string)
- ### _tag_options ($options = array())
- ### cdata_section ($content)
- ### content_tag ($name, $content = &apos;&apos;, $options = array())
- ### escape_javascript ($javascript = &apos;&apos;)
  Escape carrier returns and single and double quotes for Javascript segments.
- ### (string) escape_once ($html)

  $html HTML string to escape

  Escapes an HTML string.

  returns escaped string

- ### (string) fix_double_escape ($escaped)

$escaped	HTML string to fix

  Fixes double escaped strings.

  returns escaped string

- ### (string) tag (($name) $name, ($options) $options = array(), ($open) $open = false)

$name	string tag name
$options	array tag options
$open	boolean true to leave tag open

  Constructs an html tag.

[more]

Functions in TextHelper

Function summary

auto_link_text($text, $link = &apos;all&apos;, $href_options = array())	Turns all urls ar addresses into c links. The +link can limit what s linked.
excerpt_text($text, $phrase, $radius = 100, $excerpt_string = &apos;...&apos;)	Extracts an exce +text+ surround +phrase+ with a characters on ea determined
highlight_text($text, $phrase, $highlighter = &apos;<strong class="highlight">\\1</strong>&apos;)	Highlights the + where it is foun +text+ by surro <strong class="highligh a highlight phra The highlighter specialized by p +highlighter+ a single-quoted st where the phras to be inserted.
simple_format_text($text, $options = array())	
strip_links_text($text)	
truncate_text($text, $length = 30, $truncate_string = &apos;...&apos;, $truncate_lastspace = false)	Truncates +text length of +lengt replaces the last characters with +truncate_string +text+ is longer +length+.
wrap_text($text, $line_width = 80)	Word wrap long line_width.

Function details

- **auto_link_text ($text, $link = &apos;all&apos;, $href_options = array())**
  Turns all urls and email addresses into clickable links. The +link+ parameter can limit what should be linked.

  Options are :all (default), :email_addresses, and :urls.

  Example: auto_link("Go to http://www.symfony-project.com and say hello to fabien.potencier@example.com") => Go to <a href="http://www.symfony-project.com">http://www.symfony-project.com</a> and say hello to <a href="mailto:fabien.potencier@example.com">fabien.potencier@example.com</a>
- **excerpt_text ($text, $phrase, $radius = 100, $excerpt_string = &apos;...&apos;)**
  Extracts an excerpt from the +text+ surrounding the +phrase+ with a number of characters on each side determined

  by +radius+. If the phrase isn&apos;t found, nil is returned. Ex: excerpt("hello my world", "my", 3) => "...lo my wo..."
- **highlight_text ($text, $phrase, $highlighter = &apos;<strong class="highlight">\\1</strong>&apos;)**
  Highlights the +phrase+ where it is found in the +text+ by surrounding it like <strong class="highlight">I&apos;m a highlight phrase</strong>. The highlighter can be specialized by passing +highlighter+ as single-quoted string with \1 where the phrase is supposed to be inserted.

  N.B.: The +phrase+ is sanitized to include only letters, digits, and spaces before use.
- **simple_format_text ($text, $options = array())**
- **strip_links_text ($text)**
- **truncate_text ($text, $length = 30, $truncate_string = &apos;...&apos;, $truncate_lastspace = false)**
  Truncates +text+ to the length of +length+ and replaces the last three characters with the +truncate_string+ if the +text+ is longer than +length+.
- **wrap_text ($text, $line_width = 80)**
  Word wrap long lines to line_width.

[more]

Functions in UrlHelper

UrlHelper.

Function summary

	_confirm_javascript_function($confirm)	
	_convert_options_to_javascript($html_options, $internal_uri = &apos;&apos;)	
	_encodeText($text)	
	_popup_javascript_function($popup, $internal_uri = &apos;&apos;)	
	_post_javascript_function()	
string	button_to($name, $internal_uri, $options = array())	Creates an <input> bu of the give pointing to routed URI on the module/act passed as argument a routing configurati
string	link_to($name = &apos;&apos;, $internal_uri = &apos;&apos;, $options = array())	Creates a < tag of the g name using routed URI on the module/act passed as argument a routing configurati
string	link_to_if($condition, $name = &apos;&apos;, $internal_uri = &apos;&apos;, $options = array())	If the cond passed as f argument i creates a < tag of the g name using routed URI on the module/act passed as argument a routing

		configurati
string	link_to_unless($condition, $name = &apos;&apos;, $url = &apos;&apos;, $options = array())	If the cond passed as f argument i creates a < tag of the g name using routed UR on the module/act passed as argument a routing configurati
string	mail_to($email, $name = &apos;&apos;, $options = array(), $default_value = array())	Creates a < tag to the g email (with href="mail
string	url_for($internal_uri, $absolute = false)	Returns a r URL based module/act passed as argument a routing configurati

Function details

- ### _confirm_javascript_function ($confirm)
- ### _convert_options_to_javascript ($html_options, $internal_uri = &apos;&apos;)
- ### _encodeText ($text)
- ### _popup_javascript_function ($popup, $internal_uri = &apos;&apos;)
- ### _post_javascript_function ()
- ### (string) button_to ($name, $internal_uri, $options = array())

$name	name of the button

$internal_uri	&apos;module/action&apos; or &apos;@rule&apos; of the action
$options	additional HTML compliant <input> tag parameters

Creates an <input> button tag of the given name pointing to a routed URL based on the module/action passed as argument and the routing configuration.

The syntax is similar to the one of link_to.

Options:

- &apos;absolute&apos; - if set to true, the helper outputs an absolute URL
- &apos;query_string&apos; - to append a query string (starting by ?) to the routed url
- &apos;confirm&apos; - displays a javascript confirmation alert when the button is clicked
- &apos;popup&apos; - if set to true, the button opens a new browser window
- &apos;post&apos; - if set to true, the button submits a POST request instead of GET (caution: do not use inside a form)

Examples:

```
]]>
```

returns XHTML compliant <input> tag

- ## (string) link_to ($name = &apos;&apos;, $internal_uri = &apos;&apos;, $options = array())

$name	name of the link, i.e. string to appear between the <a> tags
$internal_uri	&apos;module/action&apos; or &apos;@rule&apos; of the action
$options	additional HTML compliant <a> tag parameters

Creates a <a> link tag of the given name using a routed URL based on the module/action passed as argument and the routing configuration.

It&apos;s also possible to pass a string instead of a module/action pair to get a link tag that just points without consideration. If null is passed as a name, the link itself will become the name. If an object is passed as a name, the object string representation is used. One of the options serves for for creating javascript confirm alerts where if you pass &apos;confirm&apos; => &apos;Are you sure?&apos;, the link will be guarded with a JS popup asking that question. If the user accepts, the link is processed, otherwise not.

Options:

- &apos;absolute&apos; - if set to true, the helper outputs an absolute URL
- &apos;query_string&apos; - to append a query string (starting by ?) to the routed url
- &apos;confirm&apos; - displays a javascript confirmation alert when the link is clicked
- &apos;popup&apos; - if set to true, the link opens a new browser window

- &apos;post&apos; - if set to true, the link submits a POST request instead of GET (caution: do not use inside a form)

Note: The &apos;popup&apos; and &apos;post&apos; options are not compatible with each other.

Examples:

```
Delete this page
  echo link_to('Visit Hoogle', 'http://www.hoogle.com');
    => Visit Hoogle
  echo link_to('Delete this page', 'my_module/my_action', array('id' => 'myid', 'confirm'
    => Delete this page]]>
```

returns XHTML compliant <a href> tag

-

(string) link_to_if ($condition, $name = &apos;&apos;, $internal_uri = &apos;&apos;, $options = array())

$condition	condition
$name	name of the link, i.e. string to appear between the <a> tags
$internal_uri	&apos;module/action&apos; or &apos;@rule&apos; of the action
$options	additional HTML compliant <a> tag parameters

If the condition passed as first argument is true, creates a <a> link tag of the given name using a routed URL based on the module/action passed as argument and the routing configuration.

If the condition is false, the given name is returned between <span> tags

Options:

- &apos;tag&apos; - the HTML tag that must enclose the name if the condition is false, defaults to <span>
- &apos;absolute&apos; - if set to true, the helper outputs an absolute URL
- &apos;query_string&apos; - to append a query string (starting by ?) to the routed url
- &apos;confirm&apos; - displays a javascript confirmation alert when the link is clicked
- &apos;popup&apos; - if set to true, the link opens a new browser window
- &apos;post&apos; - if set to true, the link submits a POST request instead of GET (caution: do not use inside a form)

Examples:

```
isAdministrator(), 'Delete this page', 'my_module/my_action');
    => Delete this page
  echo link_to_if(!$user->isAdministrator(), 'Delete this page', 'my_module/my_action');
    => Delete this page]]>
```

returns XHTML compliant <a href> tag or name

(string) link_to_unless ($condition, $name = &apos;&apos;, $url = &apos;&apos;, $options = array())

$condition	condition
$name	name of the link, i.e. string to appear between the <a> tags
$url	&apos;module/action&apos; or &apos;@rule&apos; of the action
$options	additional HTML compliant <a> tag parameters

If the condition passed as first argument is false, creates a <a> link tag of the given name using a routed URL based on the module/action passed as argument and the routing configuration.

If the condition is true, the given name is returned between <span> tags

Options:

- &apos;tag&apos; - the HTML tag that must enclose the name if the condition is true, defaults to <span>
- &apos;absolute&apos; - if set to true, the helper outputs an absolute URL
- &apos;query_string&apos; - to append a query string (starting by ?) to the routed url
- &apos;confirm&apos; - displays a javascript confirmation alert when the link is clicked
- &apos;popup&apos; - if set to true, the link opens a new browser window
- &apos;post&apos; - if set to true, the link submits a POST request instead of GET (caution: do not use inside a form)

Examples:

```
isAdministrator(), 'Delete this page', 'my_module/my_action');
    => Delete this page
  echo link_to_unless(!$user->isAdministrator(), 'Delete this page', 'my_module/my_action'
    => Delete this page]]>
```

returns XHTML compliant <a href> tag or name

(string) mail_to ($email, $name = &apos;&apos;, $options = array(), $default_value = array())

$email	target email
$name	name of the link, i.e. string to appear between the <a> tags
$options	additional HTML compliant <a> tag parameters

Creates a <a> link tag to the given email (with href="mailto:...").

If null is passed as a name, the email itself will become the name.

Options:

- &apos;encode&apos; - if set to true, the email address appears with various random encoding for each letter.

The mail link still works when encoded, but the address doesn&apos;t appear in clear in the source. Use it to prevent spam (efficiency not guaranteed).

Examples:

```
webmaster@example.com
  echo mail_to('webmaster@example.com', 'send us an email');
    => send us an email
  echo mail_to('webmaster@example.com', 'send us an email', array('encode' => true));
    => send us an email]]>
```

returns XHTML compliant <a href> tag

- ## (string) url_for ($internal_uri, $absolute = false)

$internal_uri	&apos;module/action&apos; or &apos;@rule&apos; of the action
$absolute	return absolute path?

Returns a routed URL based on the module/action passed as argument and the routing configuration.

Examples:

```
 /path/to/my/action
  echo url_for('@my_rule');
    => /path/to/my/action
  echo url_for('@my_rule', true);
    => http://myapp.example.com/path/to/my/action]]>
```

returns routed URL

[more]

Functions in ValidationHelper

ValidationHelper.

Function summary

form_error($param, $options = array(), $catalogue = &apos;messages&apos;)
form_has_error($param)

Function details

- ### form_error ($param, $options = array(), $catalogue = &apos;messages&apos;)
- ### form_has_error ($param)

[more]

Class sfI18N

Method summary

__($string, $args = array(), $catalogue = &apos;messages&apos;)
createMessageFormat($source)
createMessageSource($dir)
getCountry($iso, $culture)
getDateForCulture($date, $culture)
getGlobalMessageFormat()
getGlobalMessageSource()
getInstance()
getMessageFormat()
getMessageSource()
getNativeName($culture)
getTimestampForCulture($date, $culture)
initialize($context)
setCulture($culture)
setMessageSourceDir($dir, $culture)

Method details

- ### __ ($string, $args = array(), $catalogue = &apos;messages&apos;)
- ### createMessageFormat ($source)
- ### createMessageSource ($dir)
- ### getCountry ($iso, $culture)
- ### getDateForCulture ($date, $culture)
- ### getGlobalMessageFormat ()
- ### getGlobalMessageSource ()

- **getInstance ()**
- **getMessageFormat ()**
- **getMessageSource ()**
- **getNativeName ($culture)**
- **getTimestampForCulture ($date, $culture)**
- **initialize ($context)**
- **setCulture ($culture)**
- **setMessageSourceDir ($dir, $culture)**

[more]

Class sfFileLogger

Method summary

initialize($options = array())	Initializes the file logger.
log($message, $priority, $priorityName)	Logs a message.
shutdown()	Executes the shutdown method.

Method details

- ### initialize ($options = array())

$options	Options for the logger

  Initializes the file logger.

- ### log ($message, $priority, $priorityName)

$message	Message
$priority	Message priority
$priorityName	Message priority name

  Logs a message.

- ### shutdown ()

  Executes the shutdown method.

[more]

Class sfLogger

sfLogger manages all logging in symfony projects.

sfLogger can be configuration via the logging.yml configuration file. Loggers can also be registered directly in the logging.yml configuration file.

This level list is ordered by highest priority (SF_LOG_EMERG) to lowest priority (SF_LOG_DEBUG):

Method summary

	alert($message)	Sets an alert message.
	crit($message)	Sets a critical message.
	debug($message)	Sets a debug message.
	emerg($message)	Sets an emerg message.
	err($message)	Sets an error message.
object the	getInstance()	Returns the sfLogger instance.
string	getLogLevel()	Retrieves the log level for the current logger instance.
array	getLoggers()	Retrieves current loggers.
	info($message)	Sets an info message.
	initialize()	Initializes the logger.
	log($message, $priority = SF_LOG_INFO)	Logs a message.
	notice($message)	Sets a notice message.
	registerLogger($logger)	Registers a logger.
	setLogLevel($level)	Sets a log level for the current logger instance.
	shutdown()	Executes the shutdown procedure.
	warning($message)	Sets a warning message.

Method details

- ### alert ($message)

  $message Message

  Sets an alert message.

- ### crit ($message)

  $message Message

Sets a critical message.

- ## debug ($message)

  $message Message

  Sets a debug message.

- ## emerg ($message)

  $message Message

  Sets an emerg message.

- ## err ($message)

  $message Message

  Sets an error message.

- ## (object the) getInstance ()

  Returns the sfLogger instance.

  returns sfLogger instance

- ## (string) getLogLevel ()

  Retrieves the log level for the current logger instance.

  returns Log level

- ## (array) getLoggers ()

  Retrieves current loggers.

  returns List of loggers

- ## info ($message)

  $message Message

  Sets an info message.

- ## initialize ()

  Initializes the logger.

- ## log ($message, $priority = SF_LOG_INFO)

  $message Message

  $priority Message priority

  Logs a message.

- ### notice ($message)

  $message Message

  Sets a notice message.

- ### registerLogger ($logger)

  $logger Logger name

  Registers a logger.

- ### setLogLevel ($level)

  $level Log level

  Sets a log level for the current logger instance.

- ### shutdown ()

  Executes the shutdown procedure.

  Cleans up the current logger instance.

- ### warning ($message)

  $message Message

  Sets a warning message.

[more]

Class sfLogManager

Log manager

Method summary

rotate($app, $env, $period = null, $history = null, $override = false)	Rotates log file.

Method details

- ### rotate ($app, $env, $period = null, $history = null, $override = false)

$app	Application name
$env	Enviroment name
$period	Period
$history	History
$override	Override

Rotates log file.

[more]

Class sfWebDebugLogger

Method summary

initialize($options = array())	Initializes the web debug logger.
log($message, $priority, $priorityName)	Logs a message.

Method details

- ### initialize ($options = array())

$options	Logger options

  Initializes the web debug logger.

- ### log ($message, $priority, $priorityName)

$message	Message
$priority	Message priority
$priorityName	Message priority name

  Logs a message.

[more]

Class sfConsoleRequest

Inheritence

sfConsoleRequest < sfRequest

Method summary

boolean	initialize((sfContext) $context, $parameters = array(), $attributes = array())	Initializes this sfRequest.
	shutdown()	Executes the shutdown procedure.

Methods inherited from sfRequest

__call, extractParameters, getAttribute, getAttributeHolder, getContext, getError, getErrorNames, getErrors, getMethod, getParameter, getParameterHolder, hasAttribute, hasError, hasErrors, hasParameter, initialize, newInstance, removeError, setAttribute, setError, setErrors, setMethod, setParameter, shutdown

Method details

- ### (boolean) initialize ((sfContext) $context, $parameters = array(), $attributes = array())

$context	A sfContext instance
$parameters	An associative array of initialization parameters
$attributes	An associative array of initialization attributes

  Initializes this sfRequest.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Request

- ### shutdown ()

  Executes the shutdown procedure.

[more]

Class sfRequest

sfRequest provides methods for manipulating client request information such as attributes, errors and parameters. It is also possible to manipulate the request method originally sent by the user.

Direct sub-classes

- sfConsoleRequest:
- sfWebRequest: sfWebRequest class.

Method summary

mixed	__call($method, $arguments)	Overloads a given method.
array&	extractParameters($names)	Extracts parameter values from the request.
mixed	getAttribute($name, $default = null, $ns = null)	Retrieves an attribute from the current request.
	getAttributeHolder()	Retrieves the attributes holder.
	getContext()	Retrieves the current application context.
string	getError($name, $catalogue = &apos;messages&apos;)	Retrieves an error message.
array	getErrorNames()	Retrieves an array of error names.
array	getErrors()	Retrieves an array of errors.
int	getMethod()	Retrieves this request&apos;s method.
	getParameter($name, $default = null, $ns = null)	Retrieves a paramater for the current request.
	getParameterHolder()	Retrieves the parameters for the current request.
boolean	hasAttribute($name, $ns = null)	Indicates whether or not an attribute exist for the current request.
boolean	hasError($name)	

		Indicates whether or not an error exists.
boolean	hasErrors()	Indicates whether or not any errors exist.
boolean	hasParameter($name, $ns = null)	Indicates whether or not a parameter exist for the current request.
boolean	initialize((sfContext) $context, $parameters = array(), $attributes = array())	Initializes this sfRequest.
	newInstance($class)	Retrieves a new sfRequest implementation instance.
string&	removeError($name)	Removes an error.
	setAttribute($name, $value, $ns = null)	Sets an attribute for the request.
	setError($name, $message)	Sets an error.
	setErrors($errors)	Sets an array of errors
	setMethod($methodCode)	Sets the request method.
	setParameter($name, $value, $ns = null)	Sets a parameter for the current request.
	shutdown()	Executes the shutdown procedure.

Method details

- ### (mixed) __call ($method, $arguments)

$method	Method name
$arguments	Method arguments

  Overloads a given method.

  returns User function callback

  throws sfException if call fails

- ### (array&) extractParameters ($names)

$names	An indexed array of parameter names to extract

  Extracts parameter values from the request.

returns An associative array of parameters and their values. If a specified parameter doesn&apos;t exist an empty string will be returned for its value

- ## (mixed) getAttribute ($name, $default = null, $ns = null)

$name	Attribute name
$default	Default attribute value
$ns	Namespace for the current request

Retrieves an attribute from the current request.

returns An attribute value

- ## getAttributeHolder ()

Retrieves the attributes holder.

returns The attribute holder

- ## getContext ()

Retrieves the current application context.

returns Current application context

- ## (string) getError ($name, $catalogue = &apos;messages&apos;)

$name	An error name

Retrieves an error message.

returns An error message, if the error exists, otherwise null

- ## (array) getErrorNames ()

Retrieves an array of error names.

returns An indexed array of error names

- ## (array) getErrors ()

Retrieves an array of errors.

returns An associative array of errors

- ## (int) getMethod ()

Retrieves this request&apos;s method.

returns One of the following constants:

  - ♦ sfRequest::GET

- ♦ sfRequest::POST

getParameter ($name, $default = null, $ns = null)

$name	Parameter name
$default	Parameter default value
$ns	Namespace for the current request

Retrieves a paramater for the current request.

getParameterHolder ()

Retrieves the parameters for the current request.

returns The parameter holder

(boolean) hasAttribute ($name, $ns = null)

$name	Attribute name
$ns	Namespace for the current request

Indicates whether or not an attribute exist for the current request.

returns true, if the attribute exists otherwise false

(boolean) hasError ($name)

$name	An error name

Indicates whether or not an error exists.

returns true, if the error exists, otherwise false

(boolean) hasErrors ()

Indicates whether or not any errors exist.

returns true, if any error exist, otherwise false

(boolean) hasParameter ($name, $ns = null)

$name	Parameter name
$ns	Namespace for the current request

Indicates whether or not a parameter exist for the current request.

returns true, if the paramater exists otherwise false

(boolean) initialize ((sfContext) $context, $parameters = array(), $attributes = array())

$context	A sfContext instance
$parameters	An associative array of initialization parameters
$attributes	An associative array of initialization attributes

Initializes this sfRequest.

returns true, if initialization completes successfully, otherwise false

throws sfInitializationException If an error occurs while initializing this Request

newInstance ($class)

$class	A sfRequest implementation name

Retrieves a new sfRequest implementation instance.

returns A sfRequest implementation instance

throws sfFactoryException If a request implementation instance cannot be created

(string&) removeError ($name)

$name	An error name

Removes an error.

returns An error message, if the error was removed, otherwise null

setAttribute ($name, $value, $ns = null)

$name	Attribute name
$value	Value for the attribute
$ns	Namespace for the current request

Sets an attribute for the request.

setError ($name, $message)

$name	An error name
$message	An error message

Sets an error.

setErrors ($errors)

$errors	An associative array of errors and their associated messages

Sets an array of errors

If an existing error name matches any of the keys in the supplied array, the associated message will be overridden.

setMethod ($methodCode)

$methodCode	One of the following constants: ♦ sfRequest::GET ♦ sfRequest::POST ♦ sfRequest::PUT ♦ sfRequest::DELETE ♦ sfRequest::HEAD

Sets the request method.

throws sfException

- If the specified request method is invalid

setParameter ($name, $value, $ns = null)

$name	Parameter name
$value	Parameter value
$ns	Namespace for the current request

Sets a parameter for the current request.

shutdown ()

Executes the shutdown procedure.

[more]

Class sfWebRequest

sfWebRequest class.

This class manages web requests. It parses input from the request and store them as parameters. sfWebRequest is able to parse request with routing support enabled.

Inheritence

sfWebRequest < sfRequest

Method summary

array	getAcceptableContentTypes()	Gets a list of content types ac client browser
array	getCharsets()	Gets a list of charsets accepta browser.
mixed	getCookie($name, $defaultValue = null)	Gets a cookie value.
array	getFile($name)	Retrieves an array of file info
int	getFileError($name)	Retrieves a file error.
string	getFileExtension($name)	Retrieves an extension for a g
string	getFileName($name)	Retrieves a file name.
array	getFileNames()	Retrieves an array of file nam
string	getFilePath($name)	Retrieves a file path.
int	getFileSize($name)	Retrieve a file size.
string	getFileType($name)	Retrieves a file type.
string	getFileValue($name, $key)	Retrieves a file value.
array	getFileValues($name)	Retrieves all the values from
array	getFiles()	Retrieves an array of files.
string	getHost()	Returns current host name.
	getHttpHeader($name, $prefix = &apos;http&apos;)	
array	getLanguages()	Gets a list of languages accep browser
string	getMethodName()	Returns request method.
string	getPathInfo()	Retrieves the path info for the request.
array	getPathInfoArray()	Returns the array that contain information ($_SERVER or $
string	getReferer()	Returns referer.
string	getRelativeUrlRoot()	Retrieves relative root url.

string	getScriptName()	Returns current script name.
string	getUri()	Retrieves the uniform resourc current web request.
string	getUriPrefix()	Returns Uri prefix, including hostname and server port.
boolean	hasFile($name)	Indicates whether or not a file
boolean	hasFileError($name)	Indicates whether or not a file
boolean	hasFileErrors()	Indicates whether or not any
boolean	hasFiles()	Indicates whether or not any
boolean	initialize((sfContext) $context, $parameters = array(), $attributes = array())	Initializes this sfRequest.
boolean	isAbsUri()	See if the client is using abso
boolean	isSecure()	Returns true if the current req (HTTPS protocol).
boolean	isXmlHttpRequest()	Returns true id the request is XMLHttpRequest (via protot &apos;HTTP_X_REQUESTE header).
	loadParameters()	Loads GET, PATH_INFO an the parameter list.
boolean	moveFile($name, $file, $fileMode = 0666, $create = true, $dirMode = 0777)	Moves an uploaded file.
	setRelativeUrlRoot($value)	Sets the relative root url for t request.
	shutdown()	Executes the shutdown proce
	splitHttpAcceptHeader($header)	Splits an HTTP header for the request.

Methods inherited from sfRequest

__call, extractParameters, getAttribute, getAttributeHolder, getContext, getError, getErrorNames, getErrors, getMethod, getParameter, getParameterHolder, hasAttribute, hasError, hasErrors, hasParameter, initialize, newInstance, removeError, setAttribute, setError, setErrors, setMethod, setParameter, shutdown

Method details

- ### (array) getAcceptableContentTypes ()

  Gets a list of content types acceptable by the client browser

  returns Languages ordered in the user browser preferences

- ### (array) getCharsets ()

  Gets a list of charsets acceptable by the client browser.

  returns List of charsets in preferable order

- ### (mixed) getCookie ($name, $defaultValue = null)

  Gets a cookie value.

- ### (array) getFile ($name)

$name	A file name

  Retrieves an array of file information.

  returns An associative array of file information, if the file exists, otherwise null

- ### (int) getFileError ($name)

$name	A file name

  Retrieves a file error.

  returns

  One of the following error codes:

  - **UPLOAD_ERR_OK** (no error)
  - **UPLOAD_ERR_INI_SIZE** (the uploaded file exceeds the upload_max_filesize directive in php.ini)
  - **UPLOAD_ERR_FORM_SIZE** (the uploaded file exceeds the MAX_FILE_SIZE directive that was specified in the HTML form)
  - **UPLOAD_ERR_PARTIAL** (the uploaded file was only partially uploaded)
  - **UPLOAD_ERR_NO_FILE** (no file was uploaded)

- ### (string) getFileExtension ($name)

$name	A file name

  Retrieves an extension for a given file.

  returns Extension for the file

- ### (string) getFileName ($name)

$name	A file nam.

  Retrieves a file name.

  returns A file name, if the file exists, otherwise null

- ### (array) getFileNames ()

  Retrieves an array of file names.

  returns An indexed array of file names

- ### (string) getFilePath ($name)

$name	A file name

  Retrieves a file path.

  returns A file path, if the file exists, otherwise null

- ### (int) getFileSize ($name)

$name	A file name

  Retrieve a file size.

  returns A file size, if the file exists, otherwise null

- ### (string) getFileType ($name)

$name	A file name

  Retrieves a file type.

  This may not be accurate. This is the mime-type sent by the browser during the upload.

  returns A file type, if the file exists, otherwise null

- ### (string) getFileValue ($name, $key)

$name	A file name
$key	Value to search in the file

  Retrieves a file value.

  returns File value

- ### (array) getFileValues ($name)

$name	A file name

  Retrieves all the values from a file.

  returns Associative list of the file values

- ### (array) getFiles ()
  Retrieves an array of files.

  returns An associative array of files
- ### (string) getHost ()
  Returns current host name.
- ### getHttpHeader ($name, $prefix = &apos;http&apos;)
- ### (array) getLanguages ()
  Gets a list of languages acceptable by the client browser

  returns Languages ordered in the user browser preferences
- ### (string) getMethodName ()
  Returns request method.
- ### (string) getPathInfo ()
  Retrieves the path info for the current web request.

  returns Path info
- ### (array) getPathInfoArray ()
  Returns the array that contains all request information ($_SERVER or $_ENV).

  This information is stored in the [sf_path_info_array] constant.

  returns Path information
- ### (string) getReferer ()
  Returns referer.
- ### (string) getRelativeUrlRoot ()
  Retrieves relative root url.

  returns URL
- ### (string) getScriptName ()
  Returns current script name.
- ### (string) getUri ()
  Retrieves the uniform resource identifier for the current web request.

  returns Unified resource identifier

- ## (string) getUriPrefix ()

  Returns Uri prefix, including protocol, hostname and server port.

  returns Uniform resource identifier prefix

- ## (boolean) hasFile ($name)

$name	A file name

  Indicates whether or not a file exists.

  returns true, if the file exists, otherwise false

- ## (boolean) hasFileError ($name)

$name	A file name

  Indicates whether or not a file error exists.

  returns true, if the file error exists, otherwise false

- ## (boolean) hasFileErrors ()

  Indicates whether or not any file errors occured.

  returns true, if any file errors occured, otherwise false

- ## (boolean) hasFiles ()

  Indicates whether or not any files exist.

  returns true, if any files exist, otherwise false

- ## (boolean) initialize ((sfContext) $context, $parameters = array(), $attributes = array())

$context	A sfContext instance
$parameters	An associative array of initialization parameters
$attributes	An associative array of initialization attributes

  Initializes this sfRequest.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Request

(boolean) isAbsUri ()

See if the client is using absolute uri

returns true, if is absolute uri otherwise false

(boolean) isSecure ()

Returns true if the current request is secure (HTTPS protocol).

(boolean) isXmlHttpRequest ()

Returns true id the request is a XMLHttpRequest (via prototype &apos;HTTP_X_REQUESTED_WITH&apos; header).

loadParameters ()

Loads GET, PATH_INFO and POST data into the parameter list.

(boolean) moveFile ($name, $file, $fileMode = 0666, $create = true, $dirMode = 0777)

$name	A file name
$file	An absolute filesystem path to where you would like the file moved. This includes the new filename as well, since uploaded files are stored with random names
$fileMode	The octal mode to use for the new file
$create	Indicates that we should make the directory before moving the file
$dirMode	The octal mode to use when creating the directory

Moves an uploaded file.

returns true, if the file was moved, otherwise false

throws sfFileException If a major error occurs while attempting to move the file

setRelativeUrlRoot ($value)

$value	Value for the url

Sets the relative root url for the current web request.

shutdown ()

Executes the shutdown procedure.

splitHttpAcceptHeader ($header)

$header	Header to split

Splits an HTTP header for the current web request.

[more]

Class sfConsoleResponse

sfConsoleResponse provides methods for manipulating client response in cli environment.

Inheritence

sfConsoleResponse < sfResponse

Method summary

shutdown() Executes the shutdown procedure.

Methods inherited from sfResponse

__call, getContent, getContext, getParameter, getParameterHolder, hasParameter, initialize, newInstance, sendContent, setContent, setContext, setParameter, shutdown

Method details

- ### shutdown ()
  Executes the shutdown procedure.

[more]

Class sfResponse

sfResponse provides methods for manipulating client response information such as headers, cookies and content.

Direct sub-classes

- sfConsoleResponse: sfConsoleResponse provides methods for manipulating client response in cli environment.
- sfWebResponse: sfWebResponse class.

Method summary

mixed	__call($method, $arguments)	Overloads a given method.
string	getContent()	Gets the current response content
	getContext()	Retrieves the current application context.
mixed	getParameter($name, $default = null, $ns = null)	Retrieves a parameter from the current response.
	getParameterHolder()	Retrieves the parameters from the current response.
boolean	hasParameter($name, $ns = null)	Indicates whether or not a parameter exist for the current response.
boolean	initialize((sfContext) $context, $parameters = array())	Initializes this sfResponse.
	newInstance($class)	Retrieves a new sfResponse implementation instance.
	sendContent()	Outputs the response content
	setContent($content)	Sets the response content
	setContext((sfContext) $context)	Sets the context for the current response.
	setParameter($name, $value, $ns = null)	Sets a parameter for the current response.
	shutdown()	Executes the shutdown procedure.

Method details

- ### (mixed) __call ($method, $arguments)

$method	Method name
$arguments	Method arguments

  Overloads a given method.

  returns User function callback

throws sfException If the calls fails

- ### (string) getContent ()
  Gets the current response content

  returns Content

- ### getContext ()
  Retrieves the current application context.

  returns The application context

- ### (mixed) getParameter ($name, $default = null, $ns = null)

$name	A parameter name
$default	A default paramter value
$ns	Namespace for the current response

  Retrieves a parameter from the current response.

  returns A parameter value

- ### getParameterHolder ()
  Retrieves the parameters from the current response.

  returns List of parameters

- ### (boolean) hasParameter ($name, $ns = null)

$name	A parameter name
$ns	Namespace for the current response

  Indicates whether or not a parameter exist for the current response.

  returns true, if the parameter exists otherwise false

- ### (boolean) initialize ((sfContext) $context, $parameters = array())

$context	A sfContext instance

  Initializes this sfResponse.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Response

- ## newInstance ($class)

$class	A sfResponse implementation name

  Retrieves a new sfResponse implementation instance.

  returns A sfResponse implementation instance

  throws sfFactoryException If a request implementation instance cannot be created

- ## sendContent ()

  Outputs the response content

- ## setContent ($content)

$content	Content

  Sets the response content

- ## setContext ((sfContext) $context)

$context	A sfContext instance

  Sets the context for the current response.

- ## setParameter ($name, $value, $ns = null)

$name	A parameter name
$value	The parameter value to be set
$ns	Namespace for the current response

  Sets a parameter for the current response.

- ## shutdown ()

  Executes the shutdown procedure.

[more]

Class sfWebResponse

sfWebResponse class.

This class manages web reponses. It supports cookies and headers management.

Inheritence

sfWebResponse < sfResponse

Method summary

array	__sleep()
	__wakeup()
	addCacheControlHttpHeader($name, $value = null)
	addHttpMeta($key, $value, $replace = true)
	addJavascript($js, $position = &apos;&apos;)
	addMeta($key, $value, $replace = true, $escape = true)
	addStylesheet($css, $position = &apos;&apos;, $options = array())
	addVaryHttpHeader($header)
	clearHttpHeaders()

array	getContentType()
array	getCookies()
string	getDate($timestamp, $type = &apos;rfc1123&apos;)
array	getHttpHeader($name, $default = null)
string	getHttpHeaders()
string	getHttpMetas()
string	getJavascripts($position = &apos;&apos;)
array	getMetas()
string	getStatusCode()
string	getStylesheets($position = &apos;&apos;)
string	getTitle()
boolean	hasHttpHeader($name)
boolean	initialize((sfContext) $context, $parameters = array())
boolean	isHeaderOnly()

mergeProperties((sfResponse) $response)

string normalizeHeaderName($name)

sendContent()

sendHttpHeaders()

setContentType($value)

setCookie($name, $value, $expire = null, $path = &apos;/&apos;, $domain = &apos;&apos;, $secure = false,

setHeaderOnly($value = true)

setHttpHeader($name, $value, $replace = true)

setStatusCode($code, $name = null)

setTitle($title, $escape = true)

shutdown()

Methods inherited from sfResponse

__call, getContent, getContext, getParameter, getParameterHolder, hasParameter, initialize, newInstance, sendContent, setContent, setContext, setParameter, shutdown

Method details

- ### (array) __sleep ()

  Retrieves all objects handlers for the current web response.

  returns Objects instance

- ## __wakeup ()
  Reconstructs any result that web response instance needs.
- ## addCacheControlHttpHeader ($name, $value = null)

$name	HTTP header
$value	Value for the http header

  Adds an control cache http header.
- ## addHttpMeta ($key, $value, $replace = true)

$key	Key to replace
$value	Value for the replacement
$replace	Replace or not

  Adds meta headers to the current web response.
- ## addJavascript ($js, $position = &apos;&apos;)

$js	Javascript code
$position	Directory delimiter

  Adds javascript code to the current web response.
- ## addMeta ($key, $value, $replace = true, $escape = true)

$key	Name of the header
$value	Meta header to be set
$replace	true if it&apos;s replaceable
$escape	true for escaping the header

  Adds a meta header to the current web response.
- ## addStylesheet ($css, $position = &apos;&apos;, $options = array())

$css	Stylesheet
$position	Direcotry delimiter
$options	Stylesheet options

  Adds an stylesheet to the current web response.

- ### addVaryHttpHeader ($header)

$header	HTTP header

  Adds vary to a http header.

- ### clearHttpHeaders ()

  Cleans HTTP headers from the current web response.

- ### (array) getContentType ()

  Gets response content type.

- ### (array) getCookies ()

  Retrieves cookies from the current web response.

  returns Cookies

- ### (string) getDate ($timestamp, $type = &apos;rfc1123&apos;)

$timestamp	Timestamp
$type	Format type

  Retrieves a formated date.

  returns Formated date

- ### (array) getHttpHeader ($name, $default = null)

  Gets HTTP header current value.

- ### (string) getHttpHeaders ()

  Retrieves HTTP headers from the current web response.

  returns HTTP headers

- ### (string) getHttpMetas ()

  Retrieves meta headers for the current web response.

  returns Meta headers

- ### (string) getJavascripts ($position = &apos;&apos;)

$position	Directory delimiter

  Retrieves javascript code from the current web response.

returns Javascript code

- ### (array) getMetas ()

  Retrieves all meta headers for the current web response.

  returns List of meta headers

- ### (string) getStatusCode ()

  Retrieves status code for the current web response.

  returns Status code

- ### (string) getStylesheets ($position = &apos;&apos;)

$position	Direcotry delimiter

  Retrieves stylesheets for the current web response.

  returns Stylesheets

- ### (string) getTitle ()

  Retrieves title for the current web response.

  returns Title

- ### (boolean) hasHttpHeader ($name)

  Has a HTTP header.

- ### (boolean) initialize ((sfContext) $context, $parameters = array())

$context	A sfContext instance

  Initializes this sfWebResponse.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Response

- ### (boolean) isHeaderOnly ()

  Returns if the response must only consist of HTTP headers.

  returns returns true if, false otherwise

- ### mergeProperties ((sfResponse) $response)

$response	Response instance

Copies a propertie to a new one.

- ### (string) normalizeHeaderName ($name)

$name	Header name

Retrieves a normalized Header.

returns Normalized header

- ### sendContent ()

Send content for the current web response.

- ### sendHttpHeaders ()

Send HTTP headers and cookies.

- ### setContentType ($value)

$value	Content type

Sets response content type.

- ### setCookie ($name, $value, $expire = null, $path = &apos;/&apos;, $domain = &apos;&apos;, $secure = false, $httpOnly = false)

$name	HTTP header name
$value	Value for the cookie
$expire	Cookie expiration period
$path	Path
$domain	Domain name
$secure	If secure
$httpOnly	If uses only HTTP

Sets a cookie.

throws sfException If fails to set the cookie

- ### setHeaderOnly ($value = true)

Sets if the response consist of just HTTP headers.

- ## setHttpHeader ($name, $value, $replace = true)

$name	HTTP header name
$value	Value
$replace	Replace for the value

  Sets a HTTP header.

- ## setStatusCode ($code, $name = null)

$code	HTTP status code
$name	HTTP status text

  Sets response status code.

- ## setTitle ($title, $escape = true)

$title	Title name
$escape	true, for escaping the title

  Sets title for the current web response.

- ## shutdown ()

  Executes the shutdown procedure.

[more]

Class sfCore

core symfony class.

Method summary

	addAutoloadCallable($callable)	
	bootstrap($sf_symfony_lib_dir, $sf_symfony_data_dir)	
	callBootstrap()	
	checkLock()	
	checkSymfonyVersion()	
	getAutoloadCallables()	
	getClassPath($class)	
	initAutoload()	
	initConfiguration($sf_symfony_lib_dir, $sf_symfony_data_dir, $test = false)	
	initIncludePath()	
	initSimpleAutoload($dirs)	
boolean	splAutoload($class)	Handles autoloading of classes that have been specified in autoload.yml.
	splSimpleAutoload($class)	

Method details

- **addAutoloadCallable ($callable)**
- **bootstrap ($sf_symfony_lib_dir, $sf_symfony_data_dir)**
- **callBootstrap ()**
- **checkLock ()**
- **checkSymfonyVersion ()**
- **getAutoloadCallables ()**

- ### getClassPath ($class)
- ### initAutoload ()
- ### initConfiguration ($sf_symfony_lib_dir, $sf_symfony_data_dir, $test = false)
- ### initIncludePath ()
- ### initSimpleAutoload ($dirs)
- ### (boolean) splAutoload ($class)

  $class A class name.

  Handles autoloading of classes that have been specified in autoload.yml.

  returns Returns true if the class has been loaded
- ### splSimpleAutoload ($class)

[more]

Class sfMySQLSessionStorage

Provides support for session storage using a MySQL brand database.

Inheritence

sfMySQLSessionStorage < sfStorage < sfSessionStorage

Method summary

boolean	initialize((sfContext) $context, $parameters = null)	Initializes this Storage instance.
boolean	sessionClose()	Closes a session.
boolean	sessionDestroy($id)	Destroys a session.
boolean	sessionGC($lifetime)	Cleans up old sessions.
boolean	sessionOpen($path, $name)	Opens a session.
boolean	sessionRead($id)	Reads a session.
boolean	sessionWrite($id, [[[amp]]]$data)	Writes session data.
	shutdown()	Executes the shutdown procedure.

Methods inherited from sfSessionStorage

initialize, read, remove, shutdown, write

Methods inherited from sfStorage

getContext, getParameter, getParameterHolder, hasParameter, initialize, newInstance, read, remove, setParameter, shutdown, write

Method details

- ### (boolean) initialize ((sfContext) $context, $parameters = null)

$context	A sfContext instance
$parameters	An associative array of initialization parameters

  Initializes this Storage instance.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Storage

(boolean) sessionClose ()

Closes a session.

returns true, if the session was closed, otherwise false

(boolean) sessionDestroy ($id)

$id	A session ID

Destroys a session.

returns true, if the session was destroyed, otherwise an exception is thrown

throws sfDatabaseException If the session cannot be destroyed.

(boolean) sessionGC ($lifetime)

$lifetime	The lifetime of a session

Cleans up old sessions.

returns true, if old sessions have been cleaned, otherwise an exception is thrown

throws sfDatabaseException If any old sessions cannot be cleaned

(boolean) sessionOpen ($path, $name)

Opens a session.

returns true, if the session was opened, otherwise an exception is thrown

throws sfDatabaseException If a connection with the database does not exist or cannot be created

(boolean) sessionRead ($id)

$id	A session ID

Reads a session.

returns true, if the session was read, otherwise an exception is thrown

throws sfDatabaseException If the session cannot be read

(boolean) sessionWrite ($id, [[[amp]]]$data)

$id	A session ID
[[[amp]]]$data	A serialized chunk of session data

Writes session data.

returns true, if the session was written, otherwise an exception is thrown

throws sfDatabaseException If the session data cannot be written

- ### shutdown ()

  Executes the shutdown procedure.

[more]

Class sfPDOSessionStorage

Provides support for session storage using a PDO database abstraction layer.

Inheritence

sfPDOSessionStorage < sfStorage < sfSessionStorage

Method summary

boolean	initialize((sfContext) $context, $parameters = null)	Initializes this Storage instance.
boolean	sessionClose()	Closes a session.
boolean	sessionDestroy($id)	Destroys a session.
boolean	sessionGC($lifetime)	Cleans up old sessions.
boolean	sessionOpen($path, $name)	Opens a session.
boolean	sessionRead($id)	Reads a session.
boolean	sessionWrite($id, $data)	Writes session data.
	shutdown()	Executes the shutdown procedure.

Methods inherited from sfSessionStorage

initialize, read, remove, shutdown, write

Methods inherited from sfStorage

getContext, getParameter, getParameterHolder, hasParameter, initialize, newInstance, read, remove, setParameter, shutdown, write

Method details

- ### (boolean) initialize ((sfContext) $context, $parameters = null)

$context	A sfContext instance
$parameters	An associative array of initialization parameters

Initializes this Storage instance.

returns true, if initialization completes successfully, otherwise false

throws InitializationException If an error occurs while initializing this Storage

(boolean) sessionClose ()

Closes a session.

returns true, if the session was closed, otherwise false

(boolean) sessionDestroy ($id)

$id	A session ID

Destroys a session.

returns true, if the session was destroyed, otherwise an exception is thrown

throws DatabaseException If the session cannot be destroyed

(boolean) sessionGC ($lifetime)

$lifetime	The lifetime of a session

Cleans up old sessions.

returns true, if old sessions have been cleaned, otherwise an exception is thrown

throws DatabaseException If any old sessions cannot be cleaned

(boolean) sessionOpen ($path, $name)

Opens a session.

returns true, if the session was opened, otherwise an exception is thrown

throws DatabaseException If a connection with the database does not exist or cannot be created

(boolean) sessionRead ($id)

$id	A session ID

Reads a session.

returns true, if the session was read, otherwise an exception is thrown

throws DatabaseException If the session cannot be read

(boolean) sessionWrite ($id, $data)

$id	A session ID
$data	A serialized chunk of session data

Writes session data.

returns true, if the session was written, otherwise an exception is thrown

throws DatabaseException If the session data cannot be written

- ### shutdown ()
  Executes the shutdown procedure.

[more]

Class sfPostgreSQLSessionStorage

Provides support for session storage using a PostgreSQL brand database.

Inheritence

sfPostgreSQLSessionStorage < sfStorage < sfSessionStorage

Method summary

boolean	initialize((sfContext) $context, $parameters = null)	Initializes this Storage instance.
boolean	sessionClose()	Closes a session.
boolean	sessionDestroy($id)	Destroys a session.
boolean	sessionGC($lifetime)	Cleans up old sessions.
boolean	sessionOpen($path, $name)	Opens a session.
boolean	sessionRead($id)	Reads a session.
boolean	sessionWrite($id, [[[amp]]]$data)	Writes session data.
	shutdown()	Executes the shutdown procedure.

Methods inherited from sfSessionStorage

initialize, read, remove, shutdown, write

Methods inherited from sfStorage

getContext, getParameter, getParameterHolder, hasParameter, initialize, newInstance, read, remove, setParameter, shutdown, write

Method details

- ### (boolean) initialize ((sfContext) $context, $parameters = null)

$context	A sfContext instance
$parameters	An associative array of initialization parameters

  Initializes this Storage instance.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Storage

(boolean) sessionClose ()

Closes a session.

returns true, if the session was closed, otherwise false

(boolean) sessionDestroy ($id)

$id	A session ID

Destroys a session.

returns true, if the session was destroyed, otherwise an exception is thrown

throws sfDatabaseException If the session cannot be destroyed

(boolean) sessionGC ($lifetime)

$lifetime	The lifetime of a session

Cleans up old sessions.

returns true, if old sessions have been cleaned, otherwise an exception is thrown

throws sfDatabaseException If any old sessions cannot be cleaned

(boolean) sessionOpen ($path, $name)

Opens a session.

returns true, if the session was opened, otherwise an exception is thrown

throws sfDatabaseException If a connection with the database does not exist or cannot be created

(boolean) sessionRead ($id)

$id	A session ID

Reads a session.

returns true, if the session was read, otherwise an exception is thrown

throws sfDatabaseException If the session cannot be read

(boolean) sessionWrite ($id, [[[amp]]]$data)

$id	A session ID
[[[amp]]]$data	A serialized chunk of session data

Writes session data.

returns true, if the session was written, otherwise an exception is thrown

throws sfDatabaseException If the session data cannot be written

- ## shutdown ()
  Executes the shutdown procedure.

[more]

Class sfSessionStorage

sfSessionStorage allows you to store persistent symfony data in the user session.

Inheritence

sfSessionStorage < sfStorage

Direct sub-classes

- sfMySQLSessionStorage: Provides support for session storage using a MySQL brand database.
- sfPostgreSQLSessionStorage: Provides support for session storage using a PostgreSQL brand database.
- sfPDOSessionStorage: Provides support for session storage using a PDO database abstraction layer.

Method summary

boolean	initialize((sfContext) $context, $parameters = null)	Initializes this Storage instance.
mixed&	read($key)	Reads data from this storage.
mixed&	remove($key)	Removes data from this storage.
	shutdown()	Executes the shutdown procedure.
	write($key, [[[amp]]]$data)	Writes data to this storage.

Method details

- ### (boolean) initialize ((sfContext) $context, $parameters = null)

$context	A sfContext instance
$parameters	An associative array of initialization parameters

  Initializes this Storage instance.

  returns true, if initialization completes successfully, otherwise false

  throws sfInitializationException If an error occurs while initializing this Storage

- ### (mixed&) read ($key)

$key	A unique key identifying your data

  Reads data from this storage.

  The preferred format for a key is directory style so naming conflicts can be avoided.

returns Data associated with the key

- ## (mixed&) remove ($key)

$key	A unique key identifying your data

  Removes data from this storage.

  The preferred format for a key is directory style so naming conflicts can be avoided.

  returns Data associated with the key

- ## shutdown ()

  Executes the shutdown procedure.

- ## write ($key, [[[amp]]]$data)

$key	A unique key identifying your data
[[[amp]]]$data	Data associated with your key

  Writes data to this storage.

  The preferred format for a key is directory style so naming conflicts can be avoided.

[more]

Class sfSessionTestStorage

sfSessionTestStorage is a fake sfSessionStorage implementation to allow easy testing.

Inheritence

sfSessionTestStorage < sfStorage

Method summary

	clear()	Clears all test sessions.
string	getSessionId()	Gets session id for the current session storage instance.
boolean	initialize((sfContext) $context, $parameters = null)	Initializes this Storage instance.
mixed&	read($key)	Reads data from this storage.
mixed&	remove($key)	Removes data from this storage.
	shutdown()	Executes the shutdown procedure.
	write($key, [[[amp]]]$data)	Writes data to this storage.

Methods inherited from sfStorage

getContext, getParameter, getParameterHolder, hasParameter, initialize, newInstance, read, remove, setParameter, shutdown, write

Method details

- ### clear ()
  Clears all test sessions.
- ### (string) getSessionId ()
  Gets session id for the current session storage instance.

  returns Session id
- ### (boolean) initialize ((sfContext) $context, $parameters = null)

$context	A sfContext instance
$parameters	An associative array of initialization parameters

  Initializes this Storage instance.

  returns true, if initialization completes successfully, otherwise false

throws sfInitializationException If an error occurs while initializing this Storage

- ### (mixed&) read ($key)

$key	A unique key identifying your data

  Reads data from this storage.

  The preferred format for a key is directory style so naming conflicts can be avoided.

  returns Data associated with the key

- ### (mixed&) remove ($key)

$key	A unique key identifying your data

  Removes data from this storage.

  The preferred format for a key is directory style so naming conflicts can be avoided.

  returns Data associated with the key

- ### shutdown ()

  Executes the shutdown procedure.

- ### write ($key, [[[amp]]]$data)

$key	A unique key identifying your data
[[[amp]]]$data	Data associated with your key

  Writes data to this storage.

  The preferred format for a key is directory style so naming conflicts can be avoided

[more]

Class sfStorage

sfStorage allows you to customize the way symfony stores its persistent data.

Direct sub-classes

- sfSessionTestStorage: sfSessionTestStorage is a fake sfSessionStorage implementation to allow easy testing.
- sfSessionStorage: sfSessionStorage allows you to store persistent symfony data in the user session.

Method summary

	getContext()	Retrieves the current application context.
mixed	getParameter($name, $default = null, $ns = null)	Retrieves a parameter from the validator.
	getParameterHolder()	Retrieves the parameters from the storage.
boolean	hasParameter($name, $ns = null)	Indicates whether or not a parameter exist for the storage instance.
boolean	initialize((sfContext) $context, $parameters = array())	Initializes this Storage instance.
Storage	newInstance($class)	Retrieves a new Storage implementation instance.
mixed&	read($key)	Reads data from this storage.
mixed&	remove($key)	Removes data from this storage.
	setParameter($name, $value, $ns = null)	Sets a parameter for the current storage instance.
	shutdown()	Executes the shutdown procedure.
	write($key, [[[amp]]]$data)	Writes data to this storage.

Method details

- ### getContext ()
  Retrieves the current application context.

  returns A sfContext instance
- ### (mixed) getParameter ($name, $default = null, $ns = null)

$name	Parameter name
$default	A default parameter
$ns	Namespace for the current storage

Retrieves a parameter from the validator.

returns A parameter value

getParameterHolder ()

Retrieves the parameters from the storage.

returns List of parameters

(boolean) hasParameter ($name, $ns = null)

$name	A parameter name
$ns	A parameter namespace

Indicates whether or not a parameter exist for the storage instance.

returns true, if parameter exists, otherwise false

(boolean) initialize ((sfContext) $context, $parameters = array())

$context	A sfContext instance
$parameters	An associative array of initialization parameters

Initializes this Storage instance.

returns true, if initialization completes successfully, otherwise false

throws sfInitializationException If an error occurs while initializing this sfStorage

(Storage) newInstance ($class)

$class	A Storage implementation name

Retrieves a new Storage implementation instance.

returns A Storage implementation instance

throws sfFactoryException If a storage implementation instance cannot be created

(mixed&) read ($key)

$key	A unique key identifying your data

Reads data from this storage.

The preferred format for a key is directory style so naming conflicts can be avoided.

returns Data associated with the key

throws sfStorageException If an error occurs while reading data from this storage

- ## (mixed&) remove ($key)

$key	A unique key identifying your data

Removes data from this storage.

The preferred format for a key is directory style so naming conflicts can be avoided.

returns Data associated with the key

throws sfStorageException If an error occurs while removing data from this storage

- ## setParameter ($name, $value, $ns = null)

$name	A parameter name
$value	A parameter value
$ns	Namespace for the current storage

Sets a parameter for the current storage instance.

- ## shutdown ()

Executes the shutdown procedure.

throws sfStorageException If an error occurs while shutting down this storage

- ## write ($key, [[[amp]]]$data)

$key	A unique key identifying your data
[[[amp]]]$data	Data associated with your key

Writes data to this storage.

The preferred format for a key is directory style so naming conflicts can be avoided.

throws sfStorageException If an error occurs while writing to this storage

[more]

Class sfTestBrowser

sfTestBrowser simulates a fake browser which can test a symfony application.

Inheritence

sfTestBrowser < sfBrowser

Method summary

	back()	Simulates the browser back button.
	call($uri, $method = &apos;get&apos;, $parameters = array(), $changeStack = true)	Calls a request.
	check($uri, $text = null)	Checks that the current response contains a given text.
	checkResponseElement($selector, $value = true, $options = array())	Tests that the current response matches a given CSS selector.
	forward()	Simulates the browser forward button.
	getAndCheck($module, $action, $url = null, $code = 200)	Retrieves and checks an action.
	initialize($hostname = null, $remote = null, $options = array())	Initializes the browser tester instance.
	isCached($boolean, $with_layout = false)	Tests if the given uri is cached.
object this	isForwardedTo($moduleName, $actionName, $position = &apos;last&apos;)	Checks that the request is forwarded to a given module/action.
	isRedirected($boolean = true)	Tests if the current request

	has been redirected.
isRequestParameter($key, $value)	Tests whether or not a given key and value exists in the current request.
isResponseHeader($key, $value)	Tests for a response header.
isStatusCode($statusCode = 200)	Test an status code for the current test browser.
isUriCached($uri, $boolean, $with_layout = false, (sfTestBrowser) 3)	Tests if the given uri is cached.
responseContains($text)	Tests whether or not a given string is in the response.
test()	Retrieves the lime_test instance.
throwsException($class = null, $message = null)	Tests if an exception is thrown by the latest request.

Methods inherited from sfBrowser

back, call, click, fixUri, followRedirect, forward, get, getContext, getCurrentException, getRequest, getResponse, getResponseDom, getResponseDomCssSelector, initialize, newSession, parseArgumentAsArray, post, reload, restart, setAuth, setField, setVar, shutdown

Method details

- ### back ()
  Simulates the browser back button.

  returns The current sfBrowser instance

- ### call ($uri, $method = &apos;get&apos;, $parameters = array(), $changeStack = true)

  Calls a request.

  returns The current sfBrowser instance

- ### check ($uri, $text = null)

$uri	Uniform resource identifier
$text	Text in the response

  Checks that the current response contains a given text.

  returns The current sfBrowser instance

- ### checkResponseElement ($selector, $value = true, $options = array())

$selector	The response selector
$value	Flag for the selector
$options	Options for the current test

  Tests that the current response matches a given CSS selector.

  returns The current sfBrowser instance

- ### forward ()

  Simulates the browser forward button.

  returns The current sfBrowser instance

- ### getAndCheck ($module, $action, $url = null, $code = 200)

$module	Module name
$action	Action name
$url	Url
$code	The expected return status code

  Retrieves and checks an action.

  returns The current sfBrowser instance

- ## initialize ($hostname = null, $remote = null, $options = array())

$hostname	Hostname
$remote	Remote IP address
$options	Options

Initializes the browser tester instance.

- ## isCached ($boolean, $with_layout = false)

$boolean	Flag for checking the cache
$with_layout	If have or not layout

Tests if the given uri is cached.

returns The current sfBrowser instance

- ## (object this) isForwardedTo ($moduleName, $actionName, $position = &apos;last&apos;)

$moduleName	The module name
$actionName	The action name
$position	The position in the action stack (default to the last entry)

Checks that the request is forwarded to a given module/action.

- ## isRedirected ($boolean = true)

$boolean	Flag for redirection mode

Tests if the current request has been redirected.

returns The current sfBrowser instance

- ## isRequestParameter ($key, $value)

$key	Key
$value	Value

Tests whether or not a given key and value exists in the current request.

returns The current sfBrowser instance

- ## isResponseHeader ($key, $value)

$key	Key

$value	Value

Tests for a response header.

returns The current sfBrowser instance

- ## isStatusCode ($statusCode = 200)

$statusCode	Status code to check, default 200

  Test an status code for the current test browser.

  returns The current sfBrowser instance

- ## isUriCached ($uri, $boolean, $with_layout = false, (sfTestBrowser) 3)

$uri	Uniform resource identifier
$boolean	Flag for checking the cache
$with_layout	If have or not layout
3	Test browser instance

  Tests if the given uri is cached.

- ## responseContains ($text)

$text	Text to check

  Tests whether or not a given string is in the response.

  returns The current sfBrowser instance

- ## test ()

  Retrieves the lime_test instance.

  returns The lime_test instance

- ## throwsException ($class = null, $message = null)

$class	Class name
$message	Message name

  Tests if an exception is thrown by the latest request.

  returns The current sfBrowser instance

[more]

Class sfBasicSecurityUser

sfBasicSecurityUser will handle any type of data as a credential.

Inheritence

sfBasicSecurityUser < sfUser

Method summary

	addCredential($credential)	Adds a credential.
	addCredentials(0)	Adds several credential at once.
	clearCredentials()	Clears all credentials.
	getLastRequestTime(0)	Returns the timestamp of the last user request.
boolean	hasCredential($credentials, $useAnd = true)	Returns true if user has credential.
	initialize($context, $parameters = null)	
boolean	isAuthenticated()	Returns true if user is authenticated.
	isTimedOut()	
	listCredentials()	returns an array containing the credentials
	removeCredential($credential)	Removes a credential.
	setAuthenticated($authenticated)	Sets authentication for user.
	setTimedOut()	
	shutdown()	

Methods inherited from sfUser

__call, getAttribute, getAttributeHolder, getContext, getCulture, getParameter, getParameterHolder, hasAttribute, hasParameter, initialize, newInstance, setAttribute, setCulture, setParameter, shutdown

Method details

- ### addCredential ($credential)

$credential	credential

  Adds a credential.

- ### addCredentials (0)

0	array or list of credentials

  Adds several credential at once.

- ### clearCredentials ()
  Clears all credentials.
- ### getLastRequestTime (0)
  Returns the timestamp of the last user request.
- ### (boolean) hasCredential ($credentials, $useAnd = true)

$credentials	credentials
$useAnd	useAnd specify the mode, either AND or OR

  Returns true if user has credential.
- ### initialize ($context, $parameters = null)
- ### (boolean) isAuthenticated ()
  Returns true if user is authenticated.
- ### isTimedOut ()
- ### listCredentials ()
  returns an array containing the credentials
- ### removeCredential ($credential)

$credential	credential

  Removes a credential.
- ### setAuthenticated ($authenticated)
  Sets authentication for user.
- ### setTimedOut ()
- ### shutdown ()

[more]

Class sfSecurityUser

sfSecurityUser interface provides advanced security manipulation methods.

Method summary

	addCredential($credential)	Add a credential to this user.
	clearCredentials()	Clear all credentials associated with this user.
bool	hasCredential($credential)	Indicates whether or not this user has a credential.
bool	isAuthenticated()	Indicates whether or not this user is authenticated.
	removeCredential($credential)	Remove a credential from this user.
	setAuthenticated($authenticated)	Set the authenticated status of this user.

Method details

- ### addCredential ($credential)

$credential	Credential data.

  Add a credential to this user.

- ### clearCredentials ()

  Clear all credentials associated with this user.

- ### (bool) hasCredential ($credential)

$credential	Credential data.

  Indicates whether or not this user has a credential.

  returns true, if this user has the credential, otherwise false.

- ### (bool) isAuthenticated ()

  Indicates whether or not this user is authenticated.

  returns true, if this user is authenticated, otherwise false.

- ### removeCredential ($credential)

$credential	Credential data.

Remove a credential from this user.

- ### setAuthenticated ($authenticated)

  $authenticated A flag indicating the authenticated status of this user.

  Set the authenticated status of this user.

[more]

Class sfUser

sfUser wraps a client session and provides accessor methods for user attributes. It also makes storing and retrieving multiple page form data rather easy by allowing user attributes to be stored in namespaces, which help organize data.

Direct sub-classes

- sfBasicSecurityUser: sfBasicSecurityUser will handle any type of data as a credential.

Method summary

	__call($method, $arguments)	
	getAttribute($name, $default = null, $ns = null)	
	getAttributeHolder()	
Context	getContext()	Retrieve the current application context.
string	getCulture()	Gets culture.
	getParameter($name, $default = null, $ns = null)	
	getParameterHolder()	
	hasAttribute($name, $ns = null)	
	hasParameter($name, $ns = null)	
bool	initialize((Context) $context, $parameters = array())	Initialize this User.
User	newInstance($class)	Retrieve a new sfUser implementation instance.
	setAttribute($name, $value, $ns = null)	
	setCulture($culture)	Sets culture.
	setParameter($name, $value, $ns = null)	
	shutdown()	Execute the shutdown procedure.

Method details

- ### __call ($method, $arguments)
- ### getAttribute ($name, $default = null, $ns = null)
- ### getAttributeHolder ()

- ### (Context) getContext ()

  Retrieve the current application context.

  returns A Context instance.

- ### (string) getCulture ()

  Gets culture.

- ### getParameter ($name, $default = null, $ns = null)

- ### getParameterHolder ()

- ### hasAttribute ($name, $ns = null)

- ### hasParameter ($name, $ns = null)

- ### (bool) initialize ((Context) $context, $parameters = array())

$context	A Context instance.
$parameters	An associative array of initialization parameters.

  Initialize this User.

  returns true, if initialization completes successfully, otherwise false.

  throws sfInitializationException If an error occurs while initializing this User.

- ### (User) newInstance ($class)

$class	A sfUser implementation name

  Retrieve a new sfUser implementation instance.

  returns A sfUser implementation instance.

  throws sfFactoryException If a user implementation instance cannot

- ### setAttribute ($name, $value, $ns = null)

- ### setCulture ($culture)

$culture	culture

  Sets culture.

- **setParameter ($name, $value, $ns = null)**
- **shutdown ()**
  Execute the shutdown procedure.

[more]

Class sfBrowser

sfBrowser simulates a fake browser which can surf a symfony application.

Direct sub-classes

- sfTestBrowser: sfTestBrowser simulates a fake browser which can test a symfony application.

Method summary

back()

call($uri, $method = &apos;get&apos;, $parameters = array(), $changeStack = true)

click($name, $arguments = array())

fixUri($uri)

followRedirect()

forward()

get($uri, $parameters = array())

getContext()

getCurrentException()

getRequest()

getResponse()

getResponseDom()

getResponseDomCssSelector()

initialize($hostname = null, $remote = null, $options = array())

newSession()

parseArgumentAsArray($name, $value, [[[amp]]]$vars)

post($uri, $parameters = array())

reload()

restart()

setAuth($login, $password)

setField($name, $value)

setVar($name, $value)

shutdown()

Method details

- **back ()**
- **call ($uri, $method = &apos;get&apos;, $parameters = array(), $changeStack = true)**
- **click ($name, $arguments = array())**
- **fixUri ($uri)**
- **followRedirect ()**
- **forward ()**
- **get ($uri, $parameters = array())**
- **getContext ()**
- **getCurrentException ()**
- **getRequest ()**
- **getResponse ()**
- **getResponseDom ()**
- **getResponseDomCssSelector ()**
- **initialize ($hostname = null, $remote = null, $options = array())**
- **newSession ()**
- **parseArgumentAsArray ($name, $value, [[[amp]]]$vars)**
- **post ($uri, $parameters = array())**
- **reload ()**
- **restart ()**
- **setAuth ($login, $password)**

- **setField ($name, $value)**
- **setVar ($name, $value)**
- **shutdown ()**

[more]

Class sfContext

sfContext provides information about the current application context, such as the module and action names and the module directory. References to the current controller, request, and user implementation instances are also provided.

Method summary

string	getActionName()	Retrieve the action name for this context.
	getActionStack()	Retrieve the ActionStack.
	getController()	Retrieve the controller.
	getCurrentViewInstance()	Retrieve the curretn view instance for this context.
mixed	getDatabaseConnection((name) $name = &apos;default&apos;)	Retrieve a database connection from the database manager.
	getDatabaseManager()	Retrieve the database manager.
	getI18N()	Retrieve the i18n instance
	getInstance()	Retrieve the singleton instance of this class.
	getLogger()	
string	getModuleDirectory()	Retrieve the module directory for this context.
string	getModuleName()	Retrieve the module name for this context.
	getRequest()	Retrieve the request.
	getResponse()	Retrieve the response.
	getStorage()	Retrieve the storage.
	getUser()	Retrieve the user.
	getViewCacheManager()	Retrieve the view cache manager
	hasInstance()	
	initialize()	
	removeInstance()	Removes current sfContext instance
	retrieveObjects($class, $peerMethod)	
	setResponse((sfResponse) $response)	Set the response object.
	shutdown()	Execute the shutdown procedure.

Method details

- ### (string) getActionName ()
  Retrieve the action name for this context.

  returns The currently executing action name, if one is set, otherwise null.
- ### getActionStack ()
  Retrieve the ActionStack.

  returns the sfActionStack instance
- ### getController ()
  Retrieve the controller.

  returns The current sfController implementation instance.
- ### getCurrentViewInstance ()
  Retrieve the curretn view instance for this context.

  returns The currently view instance, if one is set, otherwise null.
- ### (mixed) getDatabaseConnection ((name) $name = &apos;default&apos;)
  $name A database name.
  Retrieve a database connection from the database manager.

  This is a shortcut to manually getting a connection from an existing database implementation instance.

  If the [sf_use_database] setting is off, this will return null.

  returns A Database instance.

  throws sfDatabaseException If the requested database name does not exist.
- ### getDatabaseManager ()
  Retrieve the database manager.

  returns The current sfDatabaseManager instance.

- ### getI18N ()
  Retrieve the i18n instance

  returns The current sfI18N implementation instance.
- ### getInstance ()
  Retrieve the singleton instance of this class.

  returns A sfContext implementation instance.
- ### getLogger ()
- ### (string) getModuleDirectory ()
  Retrieve the module directory for this context.

  returns An absolute filesystem path to the directory of the currently executing module, if one is set, otherwise null.
- ### (string) getModuleName ()
  Retrieve the module name for this context.

  returns The currently executing module name, if one is set, otherwise null.
- ### getRequest ()
  Retrieve the request.

  returns The current sfRequest implementation instance.
- ### getResponse ()
  Retrieve the response.

  returns The current sfResponse implementation instance.
- ### getStorage ()
  Retrieve the storage.

  returns The current sfStorage implementation instance.
- ### getUser ()
  Retrieve the user.

  returns The current sfUser implementation instance.

- ## getViewCacheManager ()
  Retrieve the view cache manager

  returns The current sfViewCacheManager implementation instance.
- ## hasInstance ()
- ## initialize ()
- ## removeInstance ()
  Removes current sfContext instance

  This method only exists for testing purpose. Don&apos;t use it in your application code.
- ## retrieveObjects ($class, $peerMethod)
- ## setResponse ((sfResponse) $response)

$response	A sfResponse instance.

  Set the response object.
- ## shutdown ()
  Execute the shutdown procedure.

[more]

Class sfDomCssSelector

sfDomCssSelector allows to navigate a DOM with CSS selector.

based on getElementsBySelector version 0.4 - Simon Willison, March 25th 2003
http://simon.incutio.com/archive/2003/03/25/getElementsBySelector

Method summary

__construct($dom)
getElements($selector)
getElementsByTagName($nodes, $tagName, $combinator = &apos; &apos;)
getTexts($selector)
tokenize($selector)
tokenize_selectors($selector)

Method details

- ### __construct ($dom)
- ### getElements ($selector)
- ### getElementsByTagName ($nodes, $tagName, $combinator = &apos; &apos;)
- ### getTexts ($selector)
- ### tokenize ($selector)
- ### tokenize_selectors ($selector)

[more]

Class sfFakeRenderingFilter

sfFilter provides a way for you to intercept incoming requests or outgoing responses.

Inheritence

sfFakeRenderingFilter < sfFilter

Method summary

execute($filterChain)

Methods inherited from sfFilter

getContext, getParameter, getParameterHolder, hasParameter, initialize, isFirstCall, setParameter

Method details

- ### execute ($filterChain)

[more]

Class sfFillInForm

Method summary

addConverter($callable, $fields)
escapeValue($value, $name)
fillInDom($dom, $formName, $formId, $values)
fillInHtml($html, $formName, $formId, $values)
fillInXml($xml, $formName, $formId, $values)
getValue($values, $name)
hasValue($values, $name)
setSkipFields($fields)
setTypes($types)

Method details

- ### addConverter ($callable, $fields)
- ### escapeValue ($value, $name)
- ### fillInDom ($dom, $formName, $formId, $values)
- ### fillInHtml ($html, $formName, $formId, $values)
- ### fillInXml ($xml, $formName, $formId, $values)
- ### getValue ($values, $name)
- ### hasValue ($values, $name)
- ### setSkipFields ($fields)
- ### setTypes ($types)

[more]

Class sfFinder

Allow to build rules to find files and directories.

All rules may be invoked several times, except for ->in() method. Some rules are cumulative (->name() for example) whereas others are destructive (most recent value is used, ->maxdepth() method for example).

All methods return the current sfFinder object to allow easy chaining:

$files = sfFinder::type(&apos;file&apos;)->name(&apos;*.php&apos;)->in(.);

Interface loosely based on perl File::Find::Rule module.

Method summary

	args_to_array($arg_list, $not = false)	
object current	discard((list) 0)	Discards elements that matches.
object current	exec(0)	Executes function or method for each element.
	exec_ok($dir, $entry)	
object current	follow_link()	Symlink following.
	get_type()	
object current	ignore_version_control()	Ignores version control directories.
array	in()	Searches files and directories which match defined rules.
	isPathAbsolute($path)	
	is_discarded($dir, $entry)	
	is_pruned($dir, $entry)	
	match_names($dir, $entry)	
object current	maxdepth($level)	Sets maximum directory depth.
object current	mindepth($level)	Sets minimum directory depth.
object current	name((list) 0)	Adds rules that files must match.
object current	not_name((list) 0)	Adds rules that files must not match.
object current	prune((list) 0)	Traverses no further.
object current	relative()	Returns relative paths for all files and directories.
	search_in($dir, $depth)	
object current	size((list) 0)	Adds tests for file sizes.
	size_ok($dir, $entry)	
	to_regex($str)	
object new	type($name)	Sets the type of elements to returns.

Method details

- ### args_to_array ($arg_list, $not = false)

- ### (object current) discard ((list) 0)

0	a list of patterns, globs to match

  Discards elements that matches.

  returns sfFinder object

- ### (object current) exec (0)

0	function or method to call

  Executes function or method for each element.

  Element match if functino or method returns true.

  $finder->exec(&apos;myfunction&apos;); $finder->exec(array($object, &apos;mymethod&apos;));

  returns sfFinder object

- ### exec_ok ($dir, $entry)

- ### (object current) follow_link ()

  Symlink following.

  returns sfFinder object

- ### get_type ()

- ### (object current) ignore_version_control ()

  Ignores version control directories.

  Currently supports subversion, CVS, DARCS, Gnu Arch, Monotone, Bazaar-NG

  returns pakeFinder object

- ### (array) in ()

  Searches files and directories which match defined rules.

  returns list of files and directories

- ## isPathAbsolute ($path)
- ## is_discarded ($dir, $entry)
- ## is_pruned ($dir, $entry)
- ## match_names ($dir, $entry)
- ## (object current) maxdepth ($level)

  $level level

  Sets maximum directory depth.

  Finder will descend at most $level levels of directories below the starting point.

  returns sfFinder object
- ## (object current) mindepth ($level)

  $level level

  Sets minimum directory depth.

  Finder will start applying tests at level $level.

  returns sfFinder object
- ## (object current) name ((list) 0)

  0 a list of patterns, globs or strings

  Adds rules that files must match.

  You can use patterns (delimited with / sign), globs or simple strings.

  $finder->name(&apos;*.php&apos;) $finder->name(&apos;/\.php$/&apos;) // same as above $finder->name(&apos;test.php&apos;)

  returns sfFinder object
- ## (object current) not_name ((list) 0)

  0 a list of patterns, globs or strings

  Adds rules that files must not match.

returns sfFinder object

- ### (object current) prune ((list) 0)

0	a list of patterns, globs to match

  Traverses no further.

  returns sfFinder object

- ### (object current) relative ()

  Returns relative paths for all files and directories.

  returns sfFinder object

- ### search_in ($dir, $depth)

- ### (object current) size ((list) 0)

0	a list of comparison strings

  Adds tests for file sizes.

  $finder->size(&apos;> 10K&apos;); $finder->size(&apos;<= 1Ki&apos;); $finder->size(4);

  returns sfFinder object

- ### size_ok ($dir, $entry)

- ### to_regex ($str)

- ### (object new) type ($name)

$name	directory or file or any (for both file and directory)

  Sets the type of elements to returns.

  returns sfFinder object

[more]

Class sfGlobToRegex

Match globbing patterns against text.

if match_glob("foo.*", "foo.bar") echo "matched\n";

// prints foo.bar and foo.baz $regex = glob_to_regex("foo.*"); for (array(&apos;foo.bar&apos;, &apos;foo.baz&apos;, &apos;foo&apos;, &apos;bar&apos;) as $t) { if (/$regex/) echo "matched: $car\n"; }

sfGlobToRegex implements glob(3) style matching that can be used to match against text, rather than fetching names from a filesystem.

based on perl Text::Glob module.

Method summary

string	glob_to_regex($glob)	Returns a compiled regex which is the equiavlent of the globbing pattern.
	setStrictLeadingDot($boolean)	
	setStrictWildcardSlash($boolean)	

Method details

- ### (string) glob_to_regex ($glob)

$glob	glob pattern

  Returns a compiled regex which is the equiavlent of the globbing pattern.

  returns regex

- ### setStrictLeadingDot ($boolean)

- ### setStrictWildcardSlash ($boolean)

[more]

Class sfInflector

Method summary

string	camelize($lower_case_and_underscored_word)	Returns a camelized string from a lower case and underscored string by replaceing slash with double-colol and upper-casing each letter preceded by an underscore.
string	classify($table_name)	Returns model class name for given database table.
string	demodulize($class_name_in_module)	Returns classname::module with classname:: stripped off.
strong	foreign_key($class_name, $separate_class_name_and_id_with_underscore = true)	Returns classname in underscored form, with "_id" tacked on at the end.
string	humanize($lower_case_and_underscored_word)	Returns a human-readable string from a lower case and underscored word by replacing underscores with a space, and by upper-casing the initial characters.
string	tableize($class_name)	Returns corresponding table name for given classname.
string	underscore($camel_cased_word)	Returns an underscore-syntaxed version or the CamelCased string.

Method details

- ### (string) camelize ($lower_case_and_underscored_word)

$lower_case_and_underscored_word	String to camelize.

Returns a camelized string from a lower case and underscored string by replaceing slash with double-colol and upper-casing each letter preceded by an underscore.

returns Camelized string.

- ### (string) classify ($table_name)

$table_name	Table name.

Returns model class name for given database table.

returns Classified table name.

- ### (string) demodulize ($class_name_in_module)

$class_name_in_module	Classname and module pair.

Returns classname::module with classname:: stripped off.

returns Module name.

- ### (strong) foreign_key ($class_name, $separate_class_name_and_id_with_underscore = true)

$class_name	Class name.
$separate_class_name_and_id_with_underscore	Seperate with underscore.

Returns classname in underscored form, with "_id" tacked on at the end.

This is for use in dealing with foreign keys in the database.

returns Foreign key

- ### (string) humanize ($lower_case_and_underscored_word)

$lower_case_and_underscored_word	String to make more readable.

Returns a human-readable string from a lower case and underscored word by replacing underscores with a space, and by upper-casing the initial characters.

returns Human-readable string.

- ### (string) tableize ($class_name)

$class_name	Name of class to get database table name for.

  Returns corresponding table name for given classname.

  returns Name of the databse table for given class.

- ### (string) underscore ($camel_cased_word)

$camel_cased_word	String to underscore.

  Returns an underscore-syntaxed version or the CamelCased string.

  returns Underscored string.

[more]

Class sfLoader

sfLoader is a class which contains the logic to look for files/classes in symfony.

Method summary

array	getConfigPaths($configPath)	Gets the configuration file paths for a given relative configuration path.
array	getControllerDirs($moduleName)	Gets directories where controller classes are stored for a given module.
array	getGeneratorSkeletonDirs($class, $theme)	Gets directories where the skeleton is stored for a generator class and a specific theme.
string	getGeneratorTemplate($class, $theme, $path)	Gets the template to use for a generator class.
array	getGeneratorTemplateDirs($class, $theme)	Gets directories where template files are stored for a generator class and a specific theme.
array	getHelperDirs($moduleName = &apos;&apos;)	Gets the helper directories for a given module name.
string	getI18NDir($moduleName)	Gets the i18n directory to use for a given module.
array	getModelDirs()	Gets directories where model classes are stored.
string	getTemplateDir($moduleName, $templateFile)	Gets the template directory to use for a given module and template file.
array	getTemplateDirs($moduleName)	Gets directories where template files are stored for a given module.
string	getTemplatePath($moduleName, $templateFile)	Gets the template to use for a given module and template file.
	loadHelpers($helpers, $moduleName = &apos;&apos;)	Loads helpers.
	loadPluginConfig()	

Method details

-

(array) getConfigPaths ($configPath)

$configPath	The configuration path

Gets the configuration file paths for a given relative configuration path.

returns An array of paths

- ### (array) getControllerDirs ($moduleName)

$moduleName	The module name

  Gets directories where controller classes are stored for a given module.

  returns An array of directories

- ### (array) getGeneratorSkeletonDirs ($class, $theme)

$class	The generator class name
$theme	The theme name

  Gets directories where the skeleton is stored for a generator class and a specific theme.

  returns An array of directories

- ### (string) getGeneratorTemplate ($class, $theme, $path)

$class	The generator class name
$theme	The theme name
$path	The template path

  Gets the template to use for a generator class.

  returns A template path

  throws sfException

- ### (array) getGeneratorTemplateDirs ($class, $theme)

$class	The generator class name
$theme	The theme name

  Gets directories where template files are stored for a generator class and a specific theme.

  returns An array of directories

- ### (array) getHelperDirs ($moduleName = &apos;&apos;)

$moduleName	The module name

  Gets the helper directories for a given module name.

  returns An array of directories

- ## (string) getI18NDir ($moduleName)

$moduleName	The module name

Gets the i18n directory to use for a given module.

returns An i18n directory

- ## (array) getModelDirs ()

Gets directories where model classes are stored.

returns An array of directories

- ## (string) getTemplateDir ($moduleName, $templateFile)

$moduleName	The module name
$templateFile	The template file

Gets the template directory to use for a given module and template file.

returns A template directory

- ## (array) getTemplateDirs ($moduleName)

$moduleName	The module name

Gets directories where template files are stored for a given module.

returns An array of directories

- ## (string) getTemplatePath ($moduleName, $templateFile)

$moduleName	The module name
$templateFile	The template file

Gets the template to use for a given module and template file.

returns A template path

- ## loadHelpers ($helpers, $moduleName = &apos;&apos;)

$helpers	An array of helpers to load

$moduleName	A module name (optional)

Loads helpers.

throws sfViewException

- ## loadPluginConfig ()

[more]

Class sfMixer

sfMixer implements mixins and hooks.

Method summary

callMixins($hookName = null, $moreParams = array())
getCallable($name)
getCallables($name)
getMixinInstance($name)
register($name, $callable)

Method details

- ### callMixins ($hookName = null, $moreParams = array())
- ### getCallable ($name)
- ### getCallables ($name)
- ### getMixinInstance ($name)
- ### register ($name, $callable)

[more]

Class sfNumberCompare

Numeric comparisons.

sfNumberCompare compiles a simple comparison to an anonymous subroutine, which you can call with a value to be tested again. Now this would be very pointless, if sfNumberCompare didn&apos;t understand magnitudes. The target value may use magnitudes of kilobytes (k, ki), megabytes (m, mi), or gigabytes (g, gi). Those suffixed with an i use the appropriate 2**n version in accordance with the IEC standard: http://physics.nist.gov/cuu/Units/binary.html

based on perl Number::Compare module.

Method summary

__construct($test)
test($number)

Method details

- ### __construct ($test)
- ### test ($number)

[more]

Class sfParameterHolder

sfParameterHolder provides a base class for managing parameters.

Parameters, in this case, are used to extend classes with additional data that requires no additional logic to manage.

Method summary

	__construct($namespace = &apos;symfony/default&apos;)	The constructor for sfParameterHolder.
	add($parameters, $ns = null)	Set an array of parameters.
	addByRef([[[amp]]]$parameters, $ns = null)	Set an array of parameters by reference.
	clear()	Clear all parameters associated with this request.
mixed&	get($name, $default = null, $ns = null)	Retrieve a parameter with an optionally specified namespace.
array&	getAll($ns = null)	Retrieve an array of parameters, within a namespace.
string	getDefaultNamespace()	Get the default namespace value.
array	getNames($ns = null)	Retrieve an array of parameter names from an optionally specified namespace.
array	getNamespaces()	Retrieve an array of parameter namespaces.
bool	has($name, $ns = null)	Indicates whether or not a parameter exists.
bool	hasNamespace($ns)	Indicates whether or not A parameter namespace exists.
string&	remove($name, $ns = null)	Remove a parameter.
	removeNamespace($ns = null)	Remove A parameter namespace and all of its associated parameters.
	set($name, $value, $ns = null)	Set a parameter.
	setByRef($name, [[[amp]]]$value, $ns = null)	Set a parameter by reference.

Method details

-

__construct ($namespace = &apos;symfony/default&apos;)

The constructor for sfParameterHolder.

The default namespace may be overridden at initialization as follows:

```
]]>
```

- ### add ($parameters, $ns = null)

$parameters	An associative array of parameters and their associated values.
$ns	A parameter namespace.

Set an array of parameters.

If an existing parameter name matches any of the keys in the supplied array, the associated value will be overridden.

- ### addByRef ([[[amp]]]$parameters, $ns = null)

[[[amp]]]$parameters	An associative array of parameters and references to their associated values.
$ns	A parameter namespace.

Set an array of parameters by reference.

If an existing parameter name matches any of the keys in the supplied array, the associated value will be overridden.

- ### clear ()

Clear all parameters associated with this request.

- ### (mixed&) get ($name, $default = null, $ns = null)

$name	A parameter name.
$default	A default parameter value.
$ns	A parameter namespace.

Retrieve a parameter with an optionally specified namespace.

An isolated namespace may be identified by providing a value for the third argument. If not specified, the default namespace &apos;symfony/default&apos; is used.

returns A parameter value, if the parameter exists, otherwise null.

- ### (array&) getAll ($ns = null)

$ns	A parameter namespace.

Retrieve an array of parameters, within a namespace.

This method is limited to a namespace. Without any argument, it returns the parameters of the default namespace. If a namespace is passed as an argument, only the parameters of the specified namespace are returned.

returns An associative array of parameters.

- ## (string) getDefaultNamespace ()

  Get the default namespace value.

  The $default_namespace is defined as &apos;symfony/default&apos;.

  returns The default namespace.

- ## (array) getNames ($ns = null)

$ns	A parameter namespace.

  Retrieve an array of parameter names from an optionally specified namespace.

  returns An indexed array of parameter names, if the namespace exists, otherwise null.

- ## (array) getNamespaces ()

  Retrieve an array of parameter namespaces.

  returns An indexed array of parameter namespaces.

- ## (bool) has ($name, $ns = null)

$name	A parameter name.
$ns	A parameter namespace.

  Indicates whether or not a parameter exists.

  returns true, if the parameter exists, otherwise false.

- ## (bool) hasNamespace ($ns)

$ns	A parameter namespace.

  Indicates whether or not A parameter namespace exists.

  returns true, if the namespace exists, otherwise false.

- ## (string&) remove ($name, $ns = null)

$name	A parameter name.

$ns	A parameter namespace.

Remove a parameter.

returns A parameter value, if the parameter was removed, otherwise null.

- ## removeNamespace ($ns = null)

$ns	A parameter namespace.

Remove A parameter namespace and all of its associated parameters.

- ## set ($name, $value, $ns = null)

$name	A parameter name.
$value	A parameter value.
$ns	A parameter namespace.

Set a parameter.

If a parameter with the name already exists the value will be overridden.

- ## setByRef ($name, [[[amp]]]$value, $ns = null)

$name	A parameter name.
[[[amp]]]$value	A reference to a parameter value.
$ns	A parameter namespace.

Set a parameter by reference.

If a parameter with the name already exists the value will be overridden.

[more]

Class sfPropelManyToMany

Method summary

getAllObjects($object, $middleClass, $criteria = null)	
getColumn($class, $middleClass)	
getRelatedClass($class, $middleClass)	
getRelatedColumn($class, $middleClass)	
getRelatedObjects(($object) $object, ($middleClass) $middleClass, ($criteria) $criteria = null)	Gets objects related by a many-to-many relationship, with a middle table.

Method details

- ### getAllObjects ($object, $middleClass, $criteria = null)
- ### getColumn ($class, $middleClass)
- ### getRelatedClass ($class, $middleClass)
- ### getRelatedColumn ($class, $middleClass)
- ### getRelatedObjects (($object) $object, ($middleClass) $middleClass, ($criteria) $criteria = null)

$object	The object to get related objects for.
$middleClass	The middle class used for the many-to-many relationship.
$criteria	Criteria to apply to the selection.

  Gets objects related by a many-to-many relationship, with a middle table.

[more]

Class sfTimer

sfTimer class allows to time some PHP code.

Method summary

	__construct($name = &apos;&apos;)	Creates a new sfTimer instance.
integer	addTime()	Stops the timer and add the amount of time since the start to the total time.
integer	getCalls()	Gets the number of calls this timer has been called to time code.
integer	getElapsedTime()	Gets the total time elapsed for all calls of this timer.
	startTimer()	Starts the timer.

Method details

- ### __construct ($name = &apos;&apos;)

  $name The name of the timer

  Creates a new sfTimer instance.

- ### (integer) addTime ()

  Stops the timer and add the amount of time since the start to the total time.

  returns Time spend for the last call

- ### (integer) getCalls ()

  Gets the number of calls this timer has been called to time code.

  returns Number of calls

- ### (integer) getElapsedTime ()

  Gets the total time elapsed for all calls of this timer.

  returns Time in milliseconds

- ### startTimer ()

  Starts the timer.

[more]

Class sfTimerManager

sfTimerManager is a container for sfTimer objects.

Method summary

	clearTimers()	Clears all sfTimer instances stored in sfTimerManager.
	getTimer($name)	Gets a sfTimer instance.
array	getTimers()	Gets all sfTimer instances stored in sfTimerManager.

Method details

- ### clearTimers ()
  Clears all sfTimer instances stored in sfTimerManager.
- ### getTimer ($name)

  $name The name of the timer

  Gets a sfTimer instance.

  It returns the timer named $name or create a new one if it does not exist.

  returns The timer instance
- ### (array) getTimers ()
  Gets all sfTimer instances stored in sfTimerManager.

  returns An array of all sfTimer instances

[more]

Class sfToolkit

sfToolkit provides basic utility methods.

Method summary

	arrayDeepMerge()	
	clearDirectory($directory)	Clear all files in a given directory.
	clearGlob($pattern)	Clear all files and directories corresponding to a glob
string	extractClassName($filename)	Extract the class or interface name from filename.
	getArrayValueForPath($values, $name, $default = null)	
	getPhpCli()	
	getTmpDir()	From PEAR System.php
bool	hasLockFile($lockFile, $maxLockFileLifeTime)	Determine if a lock file is present.
	isArrayValuesEmpty($array)	
bool	isPathAbsolute((path) $path)	Determine if a filesystem path is absolute.
bool	isUTF8($string)	Check if a string is an utf8 using a W3C regular expr http://fr3.php.net/manual/en/function.mb-detect-enco
mixed	literalize($value, $quoted = false)	Finds the type of the passed value, returns the value a
	pregtr($search, $replacePairs)	Returns subject replaced with regular expression mat
string	replaceConstants($value)	Replaces constant identifiers in a scalar value.
	stringToArray($string)	
	stripComments($source)	
	stripslashesDeep($value)	

Method details

- ### arrayDeepMerge ()

- ### clearDirectory ($directory)

$directory	An absolute filesystem path to a directory.

Clear all files in a given directory.

- ### clearGlob ($pattern)

$pattern	An absolute filesystem pattern.

Clear all files and directories corresponding to a glob pattern.

- ### (string) extractClassName ($filename)

$filename	A filename.

  Extract the class or interface name from filename.

  returns A class or interface name, if one can be extracted, otherwise null.

- ### getArrayValueForPath ($values, $name, $default = null)

- ### getPhpCli ()

- ### getTmpDir ()

  From PEAR System.php

  LICENSE: This source file is subject to version 3.0 of the PHP license that is available through the world-wide-web at the following URI: http://www.php.net/license/3_0.txt. If you did not receive a copy of the PHP License and are unable to obtain it through the web, please send a note to license@php.net so we can mail you a copy immediately.

- ### (bool) hasLockFile ($lockFile, $maxLockFileLifeTime)

$lockFile	A max amount of life time for the lock file.

  Determine if a lock file is present.

  returns true, if the lock file is present, otherwise false.

- ### isArrayValuesEmpty ($array)

- ### (bool) isPathAbsolute ((path) $path)

$path	A filesystem path.

  Determine if a filesystem path is absolute.

  returns true, if the path is absolute, otherwise false.

- ### (bool) isUTF8 ($string)

  Check if a string is an utf8 using a W3C regular expression http://fr3.php.net/manual/en/function.mb-detect-encoding.php#50087

  returns true if $string is valid UTF-8 and false otherwise.

- ### (mixed) literalize ($value, $quoted = false)

  Finds the type of the passed value, returns the value as the new type.

- ### pregtr ($search, $replacePairs)

$search	subject to search
$replacePairs	array of search => replace pairs

  Returns subject replaced with regular expression matchs

- ### (string) replaceConstants ($value)

  $value the value to perform the replacement on

  Replaces constant identifiers in a scalar value.

  returns the value with substitutions made

- ### stringToArray ($string)

- ### stripComments ($source)

- ### stripslashesDeep ($value)

[more]

Class sfYaml

sfYaml class.

Method summary

string	dump($array)	Dump YAML from PHP array statically
	getIncludeContents($input)	
array	load($input)	Load YAML into a PHP array statically

Method details

- ### (string) dump ($array)

$array	PHP array

  Dump YAML from PHP array statically

  The dump method, when supplied with an array, will do its best to convert the array into friendly YAML.

- ### getIncludeContents ($input)

- ### (array) load ($input)

$input	Path of YAML file or string containing YAML

  Load YAML into a PHP array statically

  The load method, when supplied with a YAML stream (string or file), will do its best to convert YAML in a file into a PHP array.

  Usage:

[more]

Class sfCallbackValidator

sfCallbackValidator allows you to use a custom callback function or method to validate the input. The function should return true on valid and false on invalid and should be callable using is_callable().

When using class/method specify it as an array in yml file as [class, method]

Inheritence

sfCallbackValidator < sfValidator

Method summary

boolean	execute([[[amp]]]$value, [[[amp]]]$error)	Executes this validator.
boolean	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (boolean) execute ([[[amp]]]$value, [[[amp]]]$error)

[[[amp]]]$value	A parameter value
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (boolean) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfCompareValidator

sfCompareValidator checks the equality of two different request parameters.

passwordValidator: class: sfCompareValidator param: check: password2 compare_error: The passwords you entered do not match. Please try again.

Inheritence

sfCompareValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
	initialize($context, $parameters = null)	

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### initialize ($context, $parameters = null)

[more]

Class sfDateValidator

sfDateValidator verifies a parameter is of a date format.

Inheritence

sfDateValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Execute this validator.
	getValidDate(($value) $value, ($culture) $culture)	Converts the given date into a Unix timestamp.
bool	initialize((sfContext) $context, $parameters = null)	Initializes the validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

  Execute this validator.

  returns true, if this validator executes successfully, otherwise false

- ### getValidDate (($value) $value, ($culture) $culture)

$value	Date to convert
$culture	Language culture to use

  Converts the given date into a Unix timestamp.

  Returns null if the date is invalid

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

Initializes the validator.

returns true, if initialization completes successfully, otherwise false

[more]

Class sfEmailValidator

sfEmailValidator verifies a parameter contains a value that qualifies as an email address.

Inheritence

sfEmailValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfFileValidator

sfFileValidator allows you to apply constraints to file upload.

Inheritence

sfFileValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfHtmlValidator

Inheritence

sfHtmlValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfNumberValidator

sfNumberValidator verifies a parameter is a number and allows you to apply size constraints.

Inheritence

sfNumberValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

Executes this validator.

returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

Initializes this validator.

returns true, if initialization completes successfully, otherwise false

[more]

Class sfPropelUniqueValidator

sfPropelUniqueValidator validates that the uniqueness of a column.

This validator only works for single column primary key.

Inheritence

sfPropelUniqueValidator < sfValidator

Method summary

	execute([[[amp]]]$value, [[[amp]]]$error)	
bool	initialize((sfContext) $context, $parameters = null)	Initialize this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### execute ([[[amp]]]$value, [[[amp]]]$error)
- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context.
$parameters	An associative array of initialization parameters.

Initialize this validator.

returns true, if initialization completes successfully, otherwise false.

[more]

Class sfRegexValidator

sfRegexValidator allows you to match a value against a regular expression pattern.

Inheritence

sfRegexValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, [[[amp]]]$error)

[[[amp]]]$value	A parameter value
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfStringValidator

sfStringValidator allows you to apply string-related constraints to a parameter.

Inheritence

sfStringValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A parameter value
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfUrlValidator

sfUrlValidator verifies a parameter contains a value that qualifies as a valid URL.

Inheritence

sfUrlValidator < sfValidator

Method summary

bool	execute([[[amp]]]$value, (error) [[[amp]]]$error)	Executes this validator.
bool	initialize((sfContext) $context, $parameters = null)	Initializes this validator.

Methods inherited from sfValidator

execute, getContext, getParameter, getParameterHolder, hasParameter, initialize, setParameter

Method details

- ### (bool) execute ([[[amp]]]$value, (error) [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

  Executes this validator.

  returns true, if this validator executes successfully, otherwise false

- ### (bool) initialize ((sfContext) $context, $parameters = null)

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

[more]

Class sfValidator

sfValidator allows you to apply constraints to user entered parameters.

Direct sub-classes

- sfCallbackValidator: sfCallbackValidator allows you to use a custom callback function or method to validate the input. The function should return true on valid and false on invalid and should be callable using is_callable().
- sfUrlValidator: sfUrlValidator verifies a parameter contains a value that qualifies as a valid URL.
- sfHtmlValidator:
- sfCompareValidator: sfCompareValidator checks the equality of two different request parameters.
- sfFileValidator: sfFileValidator allows you to apply constraints to file upload.
- sfRegexValidator: sfRegexValidator allows you to match a value against a regular expression pattern.
- sfStringValidator: sfStringValidator allows you to apply string-related constraints to a parameter.
- sfNumberValidator: sfNumberValidator verifies a parameter is a number and allows you to apply size constraints.
- sfEmailValidator: sfEmailValidator verifies a parameter contains a value that qualifies as an email address.
- sfDateValidator: sfDateValidator verifies a parameter is of a date format.
- sfPropelUniqueValidator: sfPropelUniqueValidator validates that the uniqueness of a column.

Method summary

bool	execute([[[amp]]]$value, [[[amp]]]$error)	Executes this validator.
	getContext()	Retrieves the current application context.
mixed	getParameter($name, $default = null, $ns = null)	Retrieves a parameter from the validator.
	getParameterHolder()	Retrieves the parameters from the validator.
boolean	hasParameter($name, $ns = null)	Indicates whether or not a parameter exist for the validator.
bool	initialize((sfContext) $context, $parameters = array())	Initializes this validator.
	setParameter($name, $value, $ns = null)	Sets a parameter for the validator.

Method details

- ### (bool) execute ([[[amp]]]$value, [[[amp]]]$error)

[[[amp]]]$value	A file or parameter value/array
[[[amp]]]$error	An error message reference

Executes this validator.

returns true, if this validator executes successfully, otherwise false

- ## getContext ()

  Retrieves the current application context.

  returns The current sfContext instance

- ## (mixed) getParameter ($name, $default = null, $ns = null)

$name	Parameter name
$default	A default parameter value
$ns	A parameter namespace

  Retrieves a parameter from the validator.

  returns A parameter value

- ## getParameterHolder ()

  Retrieves the parameters from the validator.

  returns List of parameters

- ## (boolean) hasParameter ($name, $ns = null)

$name	A parameter name
$ns	A parameter namespace

  Indicates whether or not a parameter exist for the validator.

  returns true, if parameter exists, otherwise false

- ## (bool) initialize ((sfContext) $context, $parameters = array())

$context	The current application context
$parameters	An associative array of initialization parameters

  Initializes this validator.

  returns true, if initialization completes successfully, otherwise false

- **setParameter ($name, $value, $ns = null)**

$name	A parameter name
$value	A parameter value
$ns	A parameter namespace

Sets a parameter for the validator.

[more]

Class sfValidatorManager

sfValidatorManager provides management for request parameters and their associated validators.

Method summary

	clear()
bool	execute()
	initialize((sfContext) $context)
	registerName($name, $required = true, $message = &apos;Required&apos;, $parent = null, $group = null, $isFil
	registerValidator($name, (Validator) $validator, $parent = null)
bool	validate([[[amp]]]$name, [[[amp]]]$data, $parent)

Method details

- ### clear ()
  Clears this validator manager so it can be reused.
- ### (bool) execute ()
  Executes all validators and determine the validation status.

returns true, if validation completed successfully, otherwise false

- ## initialize ((sfContext) $context)

$context	A sfContext instance

Initializes this validator manager.

- ## registerName ($name, $required = true, $message = &apos;Required&apos;, $parent = null, $group = null, $isFile = false)

$name	A file or parameter name
$required	The required status
$message	A required error message
$parent	A group name
$group	A parent array

Registers a file or parameter.

- ## registerValidator ($name, (Validator) $validator, $parent = null)

$name	A file or parameter name
$validator	A validator implementation instance
$parent	A parent array name

Registers a validator for a file or parameter.

- ## (bool) validate ([[[amp]]]$name, [[[amp]]]$data, $parent)

[[[amp]]]$name	A file or parameter name
[[[amp]]]$data	Data associated with the file or parameter
$parent	A parent name

Validates a file or parameter.

returns true, if validation completes successfully, otherwise false

[more]

Class sfMailView

Inheritence

sfMailView < sfView < sfPHPView

Method summary

	configure()	Configures template for this view.
string	getEngine()	Retrieves the template engine associated with this view.
mixed	render($templateVars = null)	Renders the presentation and send the email to the client.

Methods inherited from sfPHPView

configure, decorate, execute, getEngine, getGlobalVars, loadCoreAndStandardHelpers, render, renderFile

Methods inherited from sfView

__call, configure, execute, getAttribute, getAttributeHolder, getComponentSlot, getContext, getDecoratorDirectory, getDecoratorTemplate, getDirectory, getEngine, getEscaping, getEscapingMethod, getExtension, getParameter, getParameterHolder, getTemplate, hasAttribute, hasComponentSlot, hasParameter, importAttributes, initialize, isDecorator, preRenderCheck, render, setAttribute, setComponentSlot, setDecorator, setDecoratorDirectory, setDecoratorTemplate, setDirectory, setEscaping, setEscapingMethod, setExtension, setParameter, setTemplate

Method details

- ### configure ()
  Configures template for this view.

  throws sfActionException If the configure fails
- ### (string) getEngine ()
  Retrieves the template engine associated with this view.

  returns sfMail
- ### (mixed) render ($templateVars = null)
  Renders the presentation and send the email to the client.

  returns Raw data of the mail

[more]

Class sfOutputEscaper

Abstract class that provides an interface for escaping of output.

Direct sub-classes

- sfOutputEscaperGetterDecorator: Abstract output escaping decorator class for "getter" objects.

Method summary

	__construct($escapingMethod, $value)	Constructor stores the escaping method and value.
mixed	__get($var)	Gets a value from the escaper.
mixed	escape($escapingMethod, $value, 2)	Decorates a PHP variable with something that will escape any data obtained from it.
mixed	getRawValue()	Returns the raw value associated with this instance.

Method details

- ### __construct ($escapingMethod, $value)

$escapingMethod	Escaping method
$value	Escaping value

  Constructor stores the escaping method and value.

  Since sfOutputEscaper is an abstract class, instances cannot be created directly but the constructor will be inherited by sub-classes.

- ### (mixed) __get ($var)

$var	Value to get

  Gets a value from the escaper.

  returns Value

- ### (mixed) escape ($escapingMethod, $value, 2)

$escapingMethod	the escaping method (a PHP function) to apply to the value

$value	the value to escape
2	the escaped value

Decorates a PHP variable with something that will escape any data obtained from it.

The following cases are dealt with:

- The value is null or false: null or false is returned.
- The value is scalar: the result of applying the escaping method is returned.
- The value is an array or an object that implements the ArrayAccess interface: the array is decorated such that accesses to elements yield an escaped value.
- The value implements the Traversable interface (either an Iterator, an IteratorAggregate or an internal PHP class that implements Traversable): decorated much like the array.
- The value is another type of object: decorated such that the result of method calls is escaped.

The escaping method is actually the name of a PHP function. There are a set of standard escaping methods listed in the escaping helper (sfEscapingHelper.php).

returns Escaping value

throws sfException If the escaping fails

(mixed) getRawValue ()

Returns the raw value associated with this instance.

Concrete instances of sfOutputEscaper classes decorate a value which is stored by the constructor. This returns that original, unescaped, value.

returns The original value used to construct the decorator

[more]

Class sfOutputEscaperArrayDecorator

Output escaping decorator class for arrays.

Inheritence

sfOutputEscaperArrayDecorator < sfOutputEscaper < sfOutputEscaperGetterDecorator

Method summary

int	count()	Returns the size of the array (are required by the Countable interface).
mixed	current()	Escapes and return the current value (as required by the Iterator interface).
mixed	getRaw($key)	Returns the (unescaped) value from the array associated with the key supplied.
string	key()	Get the key associated with the current value (as required by the Iterator interface).
	next()	Moves to the next element (as required by the Iterator interface).
boolean	offsetExists($offset)	Returns true if the supplied offset is set in the array (as required by the ArrayAccess interface).
mixed	offsetGet($offset)	Returns the element associated with the offset supplied (as required by the ArrayAccess interface).
	offsetSet($offset, $value)	Throws an exception saying that values cannot be set (this method is required for the ArrayAccess interface).
	offsetUnset($offset)	Throws an exception saying that values cannot be unset (this method is required for the ArrayAccess interface).
	rewind()	Reset the array to the beginning (as required for the Iterator interface).
boolean	valid()	Returns true if the current element is valid (as required by the Iterator interface).

Methods inherited from sfOutputEscaperGetterDecorator

get, getRaw

Methods inherited from sfOutputEscaper

__construct, __get, escape, getRawValue

Method details

- ### (int) count ()
  Returns the size of the array (are required by the Countable interface).

  returns The size of the array
- ### (mixed) current ()
  Escapes and return the current value (as required by the Iterator interface).

  This escapes the value using sfOutputEscaper::escape with whatever escaping method is set for this instance.

  returns The escaped value
- ### (mixed) getRaw ($key)

$key	The key into the array to use

  Returns the (unescaped) value from the array associated with the key supplied.

  returns The value
- ### (string) key ()
  Get the key associated with the current value (as required by the Iterator interface).

  returns The key
- ### next ()
  Moves to the next element (as required by the Iterator interface).
- ### (boolean) offsetExists ($offset)

$offset	The offset of the value to check existance of

  Returns true if the supplied offset is set in the array (as required by the ArrayAccess interface).

  returns true if the offset exists; false otherwise
- ### (mixed) offsetGet ($offset)

$offset	The offset of the value to get

  Returns the element associated with the offset supplied (as required by the ArrayAccess interface).

  returns The escaped value

- ## offsetSet ($offset, $value)

  $offset (ignored)

  $value (ignored)

  Throws an exception saying that values cannot be set (this method is required for the ArrayAccess interface).

  This (and the other sfOutputEscaper classes) are designed to be read only so this is an illegal operation.

  throws sfException

- ## offsetUnset ($offset)

  $offset (ignored)

  Throws an exception saying that values cannot be unset (this method is required for the ArrayAccess interface).

  This (and the other sfOutputEscaper classes) are designed to be read only so this is an illegal operation.

  throws sfException

- ## rewind ()

  Reset the array to the beginning (as required for the Iterator interface).

- ## (boolean) valid ()

  Returns true if the current element is valid (as required by the Iterator interface).

  The current element will not be valid if next has fallen off the end of the array or if there are no elements in the array and rewind was called.

  returns The validity of the current element; true if it is valid

[more]

Class sfOutputEscaperGetterDecorator

Abstract output escaping decorator class for "getter" objects.

Inheritence

sfOutputEscaperGetterDecorator < sfOutputEscaper

Direct sub-classes

- sfOutputEscaperArrayDecorator: Output escaping decorator class for arrays.
- sfOutputEscaperObjectDecorator: Output escaping object decorator that intercepts all method calls and escapes their return values.

Method summary

mixed	get($key, $escapingMethod = null)	Returns the escaped value associated with the key supplied.
mixed	getRaw($key)	Returns the raw, unescaped value associated with the key supplied.

Method details

- ### (mixed) get ($key, $escapingMethod = null)

$key	The key to retieve
$escapingMethod	The escaping method (a PHP function) to use

  Returns the escaped value associated with the key supplied.

  Typically (using this implementation) the raw value is obtained using the getRaw method, escaped and the result returned.

  returns The escaped value

- ### (mixed) getRaw ($key)

$key	The key to retrieve

  Returns the raw, unescaped value associated with the key supplied.

  The key might be an index into an array or a value to be passed to the decorated object&apos;s get() method.

  returns The value

[more]

Class sfOutputEscaperIteratorDecorator

Output escaping iterator decorator.

This takes an object that implements the Traversable interface and turns it into an iterator with each value escaped.

Note: Prior to PHP 5.1, the IteratorIterator class was not implemented in the core of PHP. This means that although it will still work with classes that implement Iterator or IteratorAggregate, internal PHP classes that only implement the Traversable interface will cause the constructor to throw an exception.

Inheritence

sfOutputEscaperIteratorDecorator < sfOutputEscaper < sfOutputEscaperGetterDecorator < sfOutputEscaperObjectDecorator

Method summary

	__construct($escapingMethod, $value)	Constructs a new escaping iteratoror using the escaping method and value supplied.
int	count()	Returns the size of the array (are required by the Countable interface).
mixed	current()	Escapes and gets the current element (as required by the Iterator interface).
string	key()	Gets the current key (as required by the Iterator interface).
	next()	Moves to the next element in the iterator (as required by the Iterator interface).
boolean	offsetExists($offset)	Returns true if the supplied offset is set in the array (as required by the ArrayAccess interface).
mixed	offsetGet($offset)	Returns the element associated with the offset supplied (as required by the ArrayAccess interface).
	offsetSet($offset, $value)	Throws an exception saying that values cannot be set (this method is required for the ArrayAccess interface).
	offsetUnset($offset)	Throws an exception saying that values cannot be unset (this method is required for the ArrayAccess interface).
boolean	rewind()	Resets the iterator (as required by the Iterator interface).
boolean	valid()	Returns whether the current element is valid or not (as required by the Iterator interface).

Methods inherited from sfOutputEscaperObjectDecorator

__call, __toString, getRaw

Methods inherited from sfOutputEscaperGetterDecorator

get, getRaw

Methods inherited from sfOutputEscaper

__construct, __get, escape, getRawValue

Method details

- ### __construct ($escapingMethod, $value)

$escapingMethod	The escaping method to use
$value	The iterator to escape

  Constructs a new escaping iteratoror using the escaping method and value supplied.

- ### (int) count ()

  Returns the size of the array (are required by the Countable interface).

  returns The size of the array

- ### (mixed) current ()

  Escapes and gets the current element (as required by the Iterator interface).

  returns The escaped value

- ### (string) key ()

  Gets the current key (as required by the Iterator interface).

  returns Iterator key

- ### next ()

  Moves to the next element in the iterator (as required by the Iterator interface).

- ### (boolean) offsetExists ($offset)

$offset	The offset of the value to check existance of

  Returns true if the supplied offset is set in the array (as required by the ArrayAccess interface).

  returns true if the offset exists; false otherwise

- ## (mixed) offsetGet ($offset)

$offset	The offset of the value to get

  Returns the element associated with the offset supplied (as required by the ArrayAccess interface).

  returns The escaped value

- ## offsetSet ($offset, $value)

$offset	(ignored)
$value	(ignored)

  Throws an exception saying that values cannot be set (this method is required for the ArrayAccess interface).

  This (and the other sfOutputEscaper classes) are designed to be read only so this is an illegal operation.

  throws sfException

- ## offsetUnset ($offset)

$offset	(ignored)

  Throws an exception saying that values cannot be unset (this method is required for the ArrayAccess interface).

  This (and the other sfOutputEscaper classes) are designed to be read only so this is an illegal operation.

  throws sfException

- ## (boolean) rewind ()

  Resets the iterator (as required by the Iterator interface).

  returns true, if the iterator rewinds successfully otherwise false

- ## (boolean) valid ()

  Returns whether the current element is valid or not (as required by the Iterator interface).

  returns true if the current element is valid; false otherwise

[more]

Class sfOutputEscaperObjectDecorator

Output escaping object decorator that intercepts all method calls and escapes their return values.

Inheritence

sfOutputEscaperObjectDecorator < sfOutputEscaper < sfOutputEscaperGetterDecorator

Direct sub-classes

- sfOutputEscaperIteratorDecorator: Output escaping iterator decorator.

Method summary

mixed	__call($method, $args)	Magic PHP method that intercepts method calls, calls them on the objects that is being escaped and escapes the result.
string	__toString()	Try to call decorated object __toString() method if exists.
mixed	getRaw($key)	Returns the result of calling the get() method on the object, bypassing any escaping, if that method exists.

Method details

- ### (mixed) __call ($method, $args)

$method	The method on the object to be called
$args	An array of arguments to be passed to the method

  Magic PHP method that intercepts method calls, calls them on the objects that is being escaped and escapes the result.

  The calling of the method is changed slightly to accommodate passing a specific escaping strategy. An additional parameter is appended to the argument list which is the escaping strategy. The decorator will remove and use this parameter as the escaping strategy if it begins with &apos;esc_&apos; (the prefix all escaping helper functions have).

  For example if an object, $o, implements methods a() and b($arg):

  $o->a() // Escapes the return value of a() $o->a(ESC_RAW) // Uses the escaping method ESC_RAW with a() $o->b(&apos;a&apos;) // Escapes the return value of b(&apos;a&apos;) $o->b(&apos;a&apos;, ESC_RAW); // Uses the escaping method ESC_RAW with b(&apos;a&apos;)

  returns The escaped value returned by the method

- ### (string) __toString ()
  Try to call decorated object __toString() method if exists.

  throws sfException

- ### (mixed) getRaw ($key)

  $key The parameter to be passed to the get() get method

  Returns the result of calling the get() method on the object, bypassing any escaping, if that method exists.

  If there is not a callable get() method this will throw an exception.

  returns The unescaped value returned

  throws sfException if the object does not have a callable get() method

[more]

Class sfPartialView

Inheritence

sfPartialView < sfView < sfPHPView

Method summary

	configure()	Configures template for this view.
	execute()	Executes any presentation logic for this view.
string	render($templateVars = array())	Renders the presentation.

Methods inherited from sfPHPView

configure, decorate, execute, getEngine, getGlobalVars, loadCoreAndStandardHelpers, render, renderFile

Methods inherited from sfView

__call, configure, execute, getAttribute, getAttributeHolder, getComponentSlot, getContext, getDecoratorDirectory, getDecoratorTemplate, getDirectory, getEngine, getEscaping, getEscapingMethod, getExtension, getParameter, getParameterHolder, getTemplate, hasAttribute, hasComponentSlot, hasParameter, importAttributes, initialize, isDecorator, preRenderCheck, render, setAttribute, setComponentSlot, setDecorator, setDecoratorDirectory, setDecoratorTemplate, setDirectory, setEscaping, setEscapingMethod, setExtension, setParameter, setTemplate

Method details

- ### configure ()
  Configures template for this view.
- ### execute ()
  Executes any presentation logic for this view.
- ### (string) render ($templateVars = array())

$templateVars	Template attributes

  Renders the presentation.

  returns Current template content

[more]

Class sfPHPView

Inheritence

sfPHPView < sfView

Direct sub-classes

- sfMailView:
- sfPartialView:

Method summary

	configure()	Configures template.
string	decorate($content)	Loop through all template slots and fill them in with the results of presentation data.
	execute()	Executes any presentation logic for this view.
	getEngine()	Retrieves the template engine associated with this view.
array	getGlobalVars()	Returns variables that will be accessible to the template.
	loadCoreAndStandardHelpers()	Load core and standard helpers to be use in the template.
string	render($templateVars = null)	Renders the presentation.
string	renderFile($_sfFile)	Renders the presentation.

Method details

- ### configure ()

  Configures template.

- ### (string) decorate ($content)

$content	A chunk of decorator content

  Loop through all template slots and fill them in with the results of presentation data.

  returns A decorated template

- ### execute ()

  Executes any presentation logic for this view.

- ### getEngine ()
  Retrieves the template engine associated with this view.

  Note: This will return null because PHP itself has no engine reference.
- ### (array) getGlobalVars ()
  Returns variables that will be accessible to the template.

  returns Attributes from the template
- ### loadCoreAndStandardHelpers ()
  Load core and standard helpers to be use in the template.
- ### (string) render ($templateVars = null)
  Renders the presentation.

  When the controller render mode is sfView::RENDER_CLIENT, this method will render the presentation directly to the client and null will be returned.

  returns A string representing the rendered presentation, if the controller render mode is sfView::RENDER_VAR, otherwise null
- ### (string) renderFile ($_sfFile)

  $_sfFile Filename

  Renders the presentation.

  returns File content

[more]

Class sfView

A view represents the presentation layer of an action. Output can be customized by supplying attributes, which a template can manipulate and display.

Direct sub-classes

- sfPHPView:

Method summary

mixed	__call($method, $arguments)	Overloads given meth
	configure()	Configures template.
	execute()	Executes a presentatio logic and s template attributes.
mixed	getAttribute($name, $default = null, $ns = null)	Retrieves a attribute fo the current view.
	getAttributeHolder()	Retrieves attributes f the current view.
array	getComponentSlot($name)	Gets a componen slot
	getContext()	Retrieves t current applicatio context.
string	getDecoratorDirectory()	Retrieves t views decorator template directory.
string	getDecoratorTemplate()	Retrieves t views decorator

		template.
string	getDirectory()	Retrieves t view template directory.
mixed	getEngine()	Retrieves t template engine associated with this view.
string	getEscaping()	Gets the default escaping strategy associated with this view.
string	getEscapingMethod()	Returns th name of th function th is to be us as the escaping method.
string	getExtension()	Retrieves t current vie extension.
mixed	getParameter($name, $default = null, $ns = null)	Retrieves a parameter from the current vie
	getParameterHolder()	Retrieves t parameters for the current vie
string	getTemplate()	Retrieves t views template.
mixed	hasAttribute($name, $ns = null)	Returns tru if the view have attributes.
boolean	hasComponentSlot($name)	Indicates whether or not a

		componen slot exists.
boolean	hasParameter($name, $ns = null)	Indicates whether or not a parameter exist for th current vie
	importAttributes($names, $files = false, $errors = true, $stripTags = true, $specialChars = true)	Imports parameter values and error messages from the request directly as view attributes.
boolean	initialize((sfContext) $context, $moduleName, $actionName, $viewName)	Initializes this view.
boolean	isDecorator()	Indicates t this view i decorating view.
	preRenderCheck()	Executes a basic pre-render check to verify all required variables exist and t the templa is readable
string	render($templateVars = null)	Renders th presentatio
	setAttribute($name, $value, $ns = null)	Sets an attribute o the view.
	setComponentSlot($attributeName, $moduleName, $componentName)	Sets the module an action to b executed i place of a particular template

	attribute.
setDecorator($boolean)	Sets the decorating mode for t current vie
setDecoratorDirectory($directory)	Sets the decorator template directory f this view.
setDecoratorTemplate($template)	Sets the decorator template fo this view.
setDirectory($directory)	Sets the template directory f this view.
setEscaping($escaping)	Sets the escape caracter mode.
setEscapingMethod($method)	Sets the escaping method fo the current view.
setExtension($ext)	Sets an extension f the current view.
setParameter($name, $value, $ns = null)	Sets a parameter the view.
setTemplate($template)	Sets the template fo this view.

Method details

- ### (mixed) __call ($method, $arguments)

$method	Method name
$arguments	Method arguments

Overloads a given method

returns User function callback

throws sfException If the call fails

- ## configure ()
  Configures template.
- ## execute ()
  Executes any presentation logic and set template attributes.
- ## (mixed) getAttribute ($name, $default = null, $ns = null)

$name	Name of the attribute
$default	Value of the attribute
$ns	The current namespace

  Retrieves an attribute for the current view.

  returns Attribute
- ## getAttributeHolder ()
  Retrieves attributes for the current view.

  returns The attribute parameter holder
- ## (array) getComponentSlot ($name)

$name	The component slot name

  Gets a component slot

  returns The component slot
- ## getContext ()
  Retrieves the current application context.

  returns The current sfContext instance
- ## (string) getDecoratorDirectory ()
  Retrieves this views decorator template directory.

  returns An absolute filesystem path to this views decorator template directory

- ### (string) getDecoratorTemplate ()
  Retrieves this views decorator template.

  returns A template filename, if a template has been set, otherwise null
- ### (string) getDirectory ()
  Retrieves this view template directory.

  returns An absolute filesystem path to this views template directory
- ### (mixed) getEngine ()
  Retrieves the template engine associated with this view.

  Note: This will return null for PHPView instances.

  returns A template engine instance
- ### (string) getEscaping ()
  Gets the default escaping strategy associated with this view.

  The escaping strategy specifies how the variables get passed to the view.

  returns the escaping strategy
- ### (string) getEscapingMethod ()
  Returns the name of the function that is to be used as the escaping method.

  If the escaping method is empty, then that is returned. The default value specified by the sub-class will be used. If the method does not exist (in the sense there is no define associated with the method) and exception is thrown.

  returns The escaping method as the name of the function to use

  throws sfException If the method does not exist
- ### (string) getExtension ()
  Retrieves the current view extension.

  returns The extension for current view.
- ### (mixed) getParameter ($name, $default = null, $ns = null)

$name	Parameter name
$default	Default parameter value
$ns	Namespace for the current view

Retrieves a parameter from the current view.

returns A parameter value

-

getParameterHolder ()

Retrieves the parameters for the current view.

returns The parameter holder

-

(string) getTemplate ()

Retrieves this views template.

returns A template filename, if a template has been set, otherwise null

-

(mixed) hasAttribute ($name, $ns = null)

$name	Name of the attribute
$ns	Namespace for the current view

Returns true if the view have attributes.

returns Attribute of the view

-

(boolean) hasComponentSlot ($name)

$name	The component slot name

Indicates whether or not a component slot exists.

returns true, if the component slot exists, otherwise false

-

(boolean) hasParameter ($name, $ns = null)

$name	Name of the paramater
$ns	Namespace for the current view

Indicates whether or not a parameter exist for the current view.

returns true, if the parameter exists otherwise false

-

importAttributes ($names, $files = false, $errors = true, $stripTags = true, $specialChars = true)

$names	An indexed array of file/parameter names
$files	Is this a list of files?
$errors	Import error messages too?
$stripTags	Run strip_tags() on attribute value?

$specialChars	Run htmlspecialchars() on attribute value?

Imports parameter values and error messages from the request directly as view attributes.

- ## (boolean) initialize ((sfContext) $context, $moduleName, $actionName, $viewName)

$context	The current application context
$moduleName	The module name for this view
$actionName	The action name for this view
$viewName	The view name

Initializes this view.

returns true, if initialization completes successfully, otherwise false

- ## (boolean) isDecorator ()

Indicates that this view is a decorating view.

returns true, if this view is a decorating view, otherwise false

- ## preRenderCheck ()

Executes a basic pre-render check to verify all required variables exist and that the template is readable.

throws sfRenderException If the pre-render check fails

- ## (string) render ($templateVars = null)

$templateVars	An array with variables that will be extracted for the template If empty, the current actions var holder will be extracted

Renders the presentation.

When the controller render mode is sfView::RENDER_CLIENT, this method will render the presentation directly to the client and null will be returned.

returns A string representing the rendered presentation, if the controller render mode is sfView::RENDER_VAR, otherwise null

- ## setAttribute ($name, $value, $ns = null)

$name	Attribute name
$value	Value for the attribute

$ns	Namespace for the current

Sets an attribute of the view.

- ## setComponentSlot ($attributeName, $moduleName, $componentName)

$attributeName	A template attribute name
$moduleName	A module name
$componentName	A component name

Sets the module and action to be executed in place of a particular template attribute.

- ## setDecorator ($boolean)

$boolean	Set the decorating mode for the view

Sets the decorating mode for the current view.

- ## setDecoratorDirectory ($directory)

$directory	An absolute filesystem path to a template directory

Sets the decorator template directory for this view.

- ## setDecoratorTemplate ($template)

$template	An absolute or relative filesystem path to a template

Sets the decorator template for this view.

If the template path is relative, it will be based on the currently executing module&apos;s template sub-directory.

- ## setDirectory ($directory)

$directory	An absolute filesystem path to a template directory

Sets the template directory for this view.

- ## setEscaping ($escaping)

$escaping	Escape code

Sets the escape caracter mode.

- ## setEscapingMethod ($method)

$method	Method for escaping

  Sets the escaping method for the current view.

- ## setExtension ($ext)

$ext	The extension name.

  Sets an extension for the current view.

- ## setParameter ($name, $value, $ns = null)

$name	Name of the parameter
$value	The parameter value
$ns	Namespace for the current view

  Sets a parameter for the view.

- ## setTemplate ($template)

$template	An absolute or relative filesystem path to a template

  Sets the template for this view.

  If the template path is relative, it will be based on the currently executing module&apos;s template sub-directory.

[more]

Class sfViewCacheManager

Cache class to cache the HTML results for actions and templates.

This class uses $cacheClass class to store cache. All cache files are stored in files in the [sf_root_dir].&apos;/cache/&apos;.[sf_app].&apos;/html&apos; directory. To disable all caching, you can set to false [sf_cache] constant.

Method summary

	addCache($moduleName, $actionName, $options = array())	Adds a cache to the manager.
array	generateNamespace($internalUri)	Generates namespaces for the cache manager
string	get($internalUri)	Retrieves namespace for the current cache.
	getCache()	Retrieves the current cache type.
mixed	getCacheConfig($internalUri, $key, $defaultValue = null)	Gets a config option from the cache.
int	getClientLifeTime($internalUri)	Retrieves client lifetime from the cache option list
	getContext()	Retrieves the current cache context.
int	getLifeTime($internalUri)	Retrieves lifetime from the cache option list.
array	getVary($internalUri)	Retrieves vary option from the cache option list.

boolean	has($internalUri)	Returns true if there is a cache.
boolean	ignore()	Ignores the cache functionality.
	initialize((sfContext) $context, (sfCache) $cacheClass, $cacheParameters = array())	Initializes the cache manager.
boolean	isCacheable($internalUri)	Returns true if the current content is cacheable.
boolean	isContextual($internalUri)	Retrieves contextual option from the cache option list.
string	lastModified($internalUri)	Retrieves the last modified time.
	registerConfiguration($moduleName)	Registers configuration options for the cache.
boolean	remove($internalUri)	Removes the cache for the current namespace.
boolean	set($data, $internalUri)	Sets the cache content
	shutdown()	Executes the shutdown procedure.
boolean	start($name, $lifeTime, $clientLifeTime = null, $vary = array())	Starts the fragment cache.
boolean	stop($name)	Stops the fragment cache.
boolean	withLayout($internalUri)	Retrieves the cache layout from the cache option list.

Method details

- ### addCache ($moduleName, $actionName, $options = array())

$moduleName	Module name
$actionName	Action name
$options	Options for the cache

  Adds a cache to the manager.

- ### (array) generateNamespace ($internalUri)

$internalUri	Internal unified resource identifier.

  Generates namespaces for the cache manager

  returns Path and filename for the current namespace

  throws sfException if the generation fails

- ### (string) get ($internalUri)

$internalUri	Internal uniform resource identifier

  Retrieves namespace for the current cache.

  returns The data of the cache

- ### getCache ()

  Retrieves the current cache type.

  returns The current cache type

- ### (mixed) getCacheConfig ($internalUri, $key, $defaultValue = null)

$internalUri	Internal uniform resource identifier
$key	Option name
$defaultValue	Default value of the option

  Gets a config option from the cache.

  returns Value of the option

(int) getClientLifeTime ($internalUri)

$internalUri	Internal uniform resource identifier

Retrieves client lifetime from the cache option list

returns Client lifetime

getContext ()

Retrieves the current cache context.

returns The sfContext instance

(int) getLifeTime ($internalUri)

$internalUri	Internal uniform resource identifier

Retrieves lifetime from the cache option list.

returns LifeTime

(array) getVary ($internalUri)

$internalUri	Internal uniform resource identifier

Retrieves vary option from the cache option list.

returns Vary options for the cache

(boolean) has ($internalUri)

$internalUri	Internal uniform resource identifier

Returns true if there is a cache.

returns true, if there is a cache otherwise false

(boolean) ignore ()

Ignores the cache functionality.

returns true, if the cache is ignore otherwise false

initialize ((sfContext) $context, (sfCache) $cacheClass, $cacheParameters = array())

$context	Current application context
$cacheClass	Type of the cache
$cacheParameters	Cache parameters

Initializes the cache manager.

- ### (boolean) isCacheable ($internalUri)

$internalUri	Internal uniform resource identifier

Returns true if the current content is cacheable.

returns true, if the content is cacheable otherwise false

- ### (boolean) isContextual ($internalUri)

$internalUri	Internal uniform resource identifier

Retrieves contextual option from the cache option list.

returns true, if is contextual otherwise false

- ### (string) lastModified ($internalUri)

$internalUri	Internal uniform resource identifier

Retrieves the last modified time.

returns Last modified datetime for the current namespace

- ### registerConfiguration ($moduleName)

$moduleName	Module name

Registers configuration options for the cache.

- ### (boolean) remove ($internalUri)

$internalUri	Internal uniform resource identifier

Removes the cache for the current namespace.

returns true, if the remove happend otherwise false

- ### (boolean) set ($data, $internalUri)

$data	Data to put in the cache
$internalUri	Internal uniform resource identifier

Sets the cache content

returns true, if the data get set successfully otherwise false

- ### shutdown ()

  Executes the shutdown procedure.

- ### (boolean) start ($name, $lifeTime, $clientLifeTime = null, $vary = array())

$name	Unique fragment name
$lifeTime	Life time for the cache
$clientLifeTime	Client life time for the cache
$vary	Vary options for the cache

  Starts the fragment cache.

  returns true, if success otherwise false

- ### (boolean) stop ($name)

$name	Unique fragment name

  Stops the fragment cache.

  returns true, if success otherwise false

- ### (boolean) withLayout ($internalUri)

$internalUri	Internal uniform resource identifier

  Retrieves the layout from the cache option list.

  returns true, if have layout otherwise false