

leaf framework Documentation



Contents

Package leaf Procedural Elements	2
index.php	2
Define LEAF APPS	2
Define LEAF BASE	2
Define LEAF REL STATUS	2
Define LEAF REL VERSION	2
Define LEAF VAR	2
Define LEAF WORKING DIR	3
Base.php	4
Benchmark.php	5
Config.php	6
Controller.php	7
Dispatcher.php	8
EndorsementManager.php	9
Exception.php	10
Hash.php	11
Input.php	12
Loader.php	13
Logger.php	14
Model.php	15
Registry.php	16
Request.php	17
Response.php	18
Router.php	19
View.php	20
Package leaf Classes	21
Class leaf Base	21
Class Constant LEAF CLASS_ID	21
Class Constant LEAF REG_KEY	22
Var \$Registry	22
Constructor __construct	22
Method clone	22
Method get	23
Method toString	23
Class leaf Benchmark	23
Class Constant LEAF CLASS_ID	24
Class Constant LEAF REG_KEY	24
Var \$indexTable	24
Constructor __construct	24
Method addIndex	24
Class leaf Config	25
Class Constant LEAF CLASS_ID	25
Class Constant LEAF REG_KEY	25

Var \$options	26
Var \$optionsTable	26
Constructor __construct	26
Method getByHashKey	26
Method offsetExists	27
Method offsetGet	27
Method offsetSet	27
Method offsetUnset	28
Method __toString	28
Class leaf_Controller	28
Class Constant LEAF_CLASS_ID	29
Class Constant LEAF_REG_KEY	29
Constructor __construct	29
Method __toString	29
Class leaf_Dispatcher	30
Class Constant LEAF_CLASS_ID	30
Class Constant LEAF_REG_KEY	30
Var \$controller	30
Var \$controllerName	30
Var \$target	31
Constructor __construct	31
Method dispatchController	31
Method __toString	31
Class leaf_EndorsementManager	32
Class Constant LEAF_CLASS_ID	32
Class Constant LEAF_REG_KEY	32
Var \$allowEndorsement	32
Var \$currentClass	33
Var \$endorsed	33
Var \$registeredEndorsed	33
Constructor __construct	33
Method getEndorsedClasses	34
Method introspectEndorsedClass	34
Method isEndorsed	34
Method loadEndorsedClass	35
Class leaf_Exception	35
Constructor __construct	36
Class leaf_Hash	36
Var \$algorithms	37
Var \$defaultAlgorithm	37
Var \$useAltMethod	37
Constructor __construct	37
Method digestFile	38
Method digestMsg	38
Method setAlgorithm	39
Class leaf_Input	39
Class Constant LEAF_CLASS_ID	39
Class Constant LEAF_REG_KEY	40
Var \$input	40

Constructor construct	40
Method offsetExists	40
Method offsetGet	40
Method offsetSet	41
Method offsetUnset	41
Class leaf Loader	41
Class Constant LEAF_CLASS_ID	42
Class Constant LEAF_REG_KEY	42
Constructor construct	42
Method endorsed	42
Method extension	42
Method library	43
Method model	43
Method plugin	43
Method toString	43
Class leaf Logger	44
Class Constant LEAF_CLASS_ID	44
Class Constant LEAF_REG_KEY	44
Var \$buffer	44
Var \$filename	45
Var \$fp	45
Var \$levels	45
Var \$stampPat	45
Var \$threshold	45
Constructor construct	46
Method end_flush	46
Method flush	46
Method setLevel	47
Method setTimeStampPattern	47
Method call	47
Class leaf Model	48
Class Constant LEAF_CLASS_ID	48
Class Constant LEAF_REG_KEY	48
Constructor construct	48
Class leaf Registry	49
Var \$instance	49
Var \$registered	49
Constructor construct	50
Method getInstance	50
Method instance	50
Method isRegistered	50
Method register	51
Method toArray	51
Method unregister	51
Method get	52
Method set	52
Class leaf Request	53
Class Constant LEAF_CLASS_ID	53
Class Constant LEAF_REG_KEY	53

Var \$action	53
Var \$controller	54
Var \$controllerFile	54
Var \$queryElems	54
Var \$segments	55
Constructor __construct	55
Method getControllerFileName	55
Method getControllerName	55
Method getQueryString	55
Method getQueryStringAsString	56
Method reconstructUrl	56
Method redirect	57
Method segment	57
Method segmentsToArray	57
Method totalSegments	58
Method toString	58
Class leaf Response	58
Class Constant LEAF_CLASS_ID	59
Class Constant LEAF_REG_KEY	59
Var \$buffer	59
Var \$useGzip	59
Var \$useTidy	59
Constructor __construct	59
Method clearHeaders	60
Method flushOutputBuffer	60
Method getOutputBuffer	60
Method outputBufferingStart	60
Method outputBufferingEnd	61
Method sendResponse	61
Method setHeader	61
Method toString	61
Class leaf Router	62
Class Constant LEAF_CLASS_ID	62
Class Constant LEAF_REG_KEY	62
Var \$queryString	62
Var \$queryStringElements	63
Var \$requestClass	63
Var \$requestMethod	63
Var \$requestUri	63
Var \$segments	63
Constructor __construct	64
Method chopSegment	64
Method getClassName	64
Method getMethodName	65
Method queryStringElements	65
Method segments	65
Method segmentsSize	65
Method toString	66
Class leaf View	66

Class Constant LEAF CLASS ID	66
Class Constant LEAF REG KEY	66
Constructor <code>__construct</code>	66
Method <code>call</code>	67
Method <code>toString</code>	67
Benchmark.php	68
Request.php	69
Hook.php	70
Hook Conditional.php	71
Class leaf Hook	71
Class Constant LEAF CLASS ID	72
Class Constant LEAF REG KEY	72
Constructor <code>__construct</code>	72
Method <code>toString</code>	72
Class leaf Hook Conditional	72
Class Constant LEAF CLASS ID	73
Class Constant LEAF REG KEY	73
Constructor <code>__construct</code>	73
Method <code>condition</code>	73
Hooks.php	74
Function <code>getHooks</code>	74
Define HOOK POST CONTROLLER DISPATCH	74
Define HOOK POST FRONT CONTROLLER	74
Define HOOK PRE CONTROLLER DISPATCH	74
Function <code>introspectHooks</code>	75
Function <code>runHook</code>	75
Function <code>runHooks</code>	75
Class leaf Log	76
Debug.php	77
Front_Controller.php	78
Debug.php	80
Function <code>inspect var</code>	80
Function <code>i var</code>	80
Function <code>inspect var Array</code>	81
Function <code>inspect var Boolean</code>	81
Function <code>inspect var Generic</code>	81
Function <code>inspect var Number</code>	81
Function <code>inspect var Object</code>	82
Function <code>inspect var Object getSubclasses</code>	82
Function <code>inspect var Resource</code>	82
Function <code>inspect var String</code>	83
Dependancies.php	84
Function <code>dependsOn</code>	84
Function <code>dependsOnFunc</code>	84
Function <code>dependsOptionalOn</code>	85
Error.php	86
Function <code>showErrorPage404</code>	86
Function <code>showHtmlMessage</code>	87
Handlers.php	88

Function errorHandler	88
Function exceptionHandler	89
Main.php	90
Function frameworkForceTerminate	90
Function frameworkSanityCheck	90
Function autoload	90
Appendices	92
Appendix A - Class Trees	93
leaf	93
Appendix B - README/CHANGELOG/INSTALL	95
TODO	96
LICENSE	96
README	96
CHANGELOG	97
INSTALL	97
Appendix C - Source Code	98
Package leaf	99
source code: Benchmark.php	100
source code: Controller.php	102
source code: Dispatcher.php	103
source code: EndorsementManager.php	105
source code: Exception.php	108
source code: Hash.php	109
source code: Input.php	111
source code: Loader.php	113
source code: Logger.php	115
source code: Model.php	119
source code: Request.php	120
source code: Response.php	124
source code: Router.php	127
source code: View.php	132
source code: Benchmark.php	134
source code: Request.php	135
source code: Hook.php	136
source code: Hook Conditional.php	137
source code: Hooks.php	138
source code: Debug.php	140
source code: Front Controller.php	144
source code: Debug.php	147
source code: Dependancies.php	152
source code: Error.php	154
source code: Handlers.php	156
source code: Main.php	158
Appendix D - Todo List	162

Package leaf Procedural Elements

index.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Author** Avraam Marimpis < makism@users.sf.net>
- **Version** \$Id: index.php 23 2007-11-04 19:18:33Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **License** [New BSD License](#)

LEAF_APPS = LEAF_WORKING_DIR.'applications/' *[line 54]*

Subdirectory in which user`s applications are located.

LEAF_BASE = LEAF_WORKING_DIR.'leaf/' *[line 49]*

Subdirectory in which leaf`s files are located.

LEAF_REL_STATUS = 'DEV' *[line 38]*

leaf`s Status and Version information.

LEAF_REL_VERSION = '1.0' *[line 39]*

LEAF_VAR = LEAF_WORKING_DIR.'var/' *[line 59]*

Subdirectory in which cache and temporary files are stored.

LEAF_WORKING_DIR = dirname(realpath(__FILE__)).'/' [line 44]

Current directory where **this file is located.**

require_once LEAF_BASE.'front/Front_Controller.php' [line 65]

The [Front Controller](#) handles the startup sequence.

Base.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **License** [New BSD License](#)

Benchmark.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Framework PHP5. Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Framework PHP5.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Config.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **License** [New BSD License](#)

Controller.php

leaf Framework

PHP version 5

leaf is a simple and fast php framework. It is designed to be easy to use and to be a good starting point for your next project. It is a simple and fast php framework. It is designed to be easy to use and to be a good starting point for your next project.

- **Package** leaf
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Dispatcher.php

leaf Framework

PHP version 5

Dispatcher.php is a part of the leaf framework. It is a PHP script that handles the dispatching of requests to the appropriate controller. It is located in the `dispatcher` directory of the leaf framework.

- **Package** leaf
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

EndorsementManager.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Ένα ελληνικό open source PHP5 framework, γρήγορο, με μικρό footprint και εύκολα extensible.
Ένα ελληνικό open source PHP5 framework, γρήγορο, με μικρό footprint και εύκολα extensible.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Exception.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Framework PHP5. Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Framework PHP5.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Hash.php

leaf Framework

PHP version 5

leaf framework, is a small, fast, and easy to use framework, with a small footprint and easily extensible.

An open source framework, small, with small footprint and easily extensible.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Input.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Το πρώτο ελληνικό open source PHP5 framework, γρήγορο, με μικρό footprint και εύκολο στην επέκταση.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Loader.php

leaf Framework

PHP version 5

leaf framework, is a small, fast, and easy to use framework, with a small footprint and easily extensible.

An open source framework, small, with small footprint and easily extensible.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Logger.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Ένα γρήγορο, μικρό, εύκολο στην επέκταση, ανοιχτό κώδικα PHP5 framework. Το πρώτο ελληνικό framework PHP5. Έχει μικρό footprint, είναι γρήγορο και εύκολο στην επέκταση. Είναι ανοιχτό κώδικα.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Model.php

leaf Framework

PHP version 5

leaf framework, is a small, fast, and easy to use framework, with a small footprint and easily extensible.

An open source framework, small, with small footprint and easily extensible.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Registry.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Ένα γρήγορο, εύκολο στην χρήση, PHP5 framework. Έχει μικρό footprint, είναι εύκολο στην επέκταση και είναι το πρώτο ελληνικό open source PHP5 framework.

- **Package** leaf
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **License** -

Request.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Ένα γρήγορο, εύκολο στην χρήση, μικρό σε μέγεθος framework PHP5. Έχει μικρή υποδομή, είναι εύκολο να επεκταθεί και να ενσωματωθεί με άλλα frameworks. Είναι το πρώτο ελληνικό open source framework PHP5.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Response.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Αυτό είναι το πρώτο ελληνικό open source PHP5 framework, γρήγορο, με μικρό footprint και εύκολο στην επέκταση.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

Router.php

leaf Framework

PHP version 5

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Απλό, εύκολο στη χρήση, γρήγορο framework PHP5. Απλό, εύκολο στη χρήση, γρήγορο framework PHP5. Απλό, εύκολο στη χρήση, γρήγορο framework PHP5. Απλό, εύκολο στη χρήση, γρήγορο framework PHP5.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

View.php

leaf Framework

PHP version 5

leaf framework, is a small, fast, and easy to use framework, with a small footprint and easily extensible.

An open source framework, small, with small footprint and easily extensible.

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

leaf_Base::LEAF_REG_KEY

= "base" [line 42]

if (\$this->leaf_registry) {
 if (\$this->leaf_registry->get(\$key)) {
 return \$this->leaf_registry->get(\$key);
 }
}

leaf_Base::\$Registry

object leaf_Registry = NULL [line 59]

if (\$this->leaf_registry) {
 if (\$this->leaf_registry->get(\$key)) {
 return \$this->leaf_registry->get(\$key);
 }
}

- **Static**
- **Access** private

Constructor void function leaf_Base::__construct([\$descendant = NULL]) [line 69]

Function Parameters:

- *NULL*|string **\$descendant**

if (\$this->leaf_registry) {
 if (\$this->leaf_registry->get(\$key)) {
 return \$this->leaf_registry->get(\$key);
 }
}

- **Access** public

void function leaf_Base::__clone() [line 117]

- **Access** private

object function leaf_Base::__get(\$obj) [line 85]

Function Parameters:

- **string \$obj**

• **Access** protected

string function leaf_Base::__toString() [line 95]

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

Class leaf_Benchmark

[line 37]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **Link** <http://www.php.net/~helly/php/ext/spl/interfaceArrayAccess.html>
- **TODO**

1. $\hat{I} \in \mathbb{R}^{n \times n}$.

- **Since** 1.0-dev
- **License** -

leaf_Benchmark::LEAF_CLASS_ID

= "LEAF_BENCHMARK-1_0_dev" [[line 41](#)]

leaf_Benchmark::LEAF_REG_KEY

= "benchmark" [[line 39](#)]

leaf_Benchmark::\$indexTable

array = array() [[line 49](#)]

- **Access** private

Constructor *void* function leaf_Benchmark::__construct() [[line 57](#)]

- **Access** public

void function leaf_Benchmark::addIndex(\$name, \$type) [[line 71](#)]

Function Parameters:

- *string* **\$name**
- *integer* **\$type**

- **Access** public

Class leaf_Config

[line 42]

Provides access to the configuration files.

This class encapsulates all the configuration files.

All declared parameters in the files, are accessible via methods.

The interface ArrayAccess is implemented which allows easy array-like access in one class` properties.

A simple example follows:

```
1 $conf = new leaf_Config();
2 echo $conf['base_uri'];
```

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@users.sf.net>
- **Version** \$Id: Config.php 13 2007-11-01 11:10:37Z makism \$
- **Copyright** -
- **Link** <http://www.php.net/~helly/php/ext/spl/interfaceArrayAccess.html>
- **TODO**
 1. Maybe we should unset the global variable \$GLOBALS somewhere else.
 2. Possible implementation of the interface *Iterator*.
 3. Possible implementation of a subclass that will read separate configuration files for each Controller, thus declaring different parameters and maybe unique for each Controller.

leaf_Config::LEAF_CLASS_ID

= "LEAF_CONFIG-1_0_dev" [line 46]

leaf_Config::LEAF_REG_KEY

= "config" [line 44]

leaf_Config::\$options

`array = array()` [line 53]

All configuration parameters will be stored in this array.

- **Access** private

leaf_Config::\$optionsTable

`array = array()` [line 61]

All configuration parameters are stored in this array, using their parent configuration file as index.

- **Access** private

Constructor *void* function leaf_Config::__construct() [line 69]

Encapsulates all the configuration files, and stores the paremeters in an array.

- **Access** public

array|NULL function leaf_Config::getByHashKey(\$key, \$str) [line 94]

Function Parameters:

- *string* **\$str**
- **\$key**

Returns all parameters related with the specified key.

- **Access** public

boolean function leaf_Config::offsetExists(\$offset) [*line 108*]

Function Parameters:

- *string* **\$offset**

Checks if the request parameter-name exists.

- **Access** public

mixed function leaf_Config::offsetGet(\$offset) [*line 122*]

Function Parameters:

- *string* **\$offset**

Returns the value of the specified parameter.

- **Access** public

void function leaf_Config::offsetSet(\$offset, \$value) [*line 134*]

Function Parameters:

- *string* **\$offset**
- *mixed* **\$value**

Sets a value in the specific parameter.

- **Access** public

```
void function leaf_Config::offsetUnset($offset) [line 149]
```

Function Parameters:

- *string* \$offset

Unsets a parameter.

Unsets a parameter while the script is executed, that is, it doesn't mean it updates the configuration file itself.

- **Access** public

```
void function leaf_Config::__toString() [line 154]
```

- **Access** public

Class leaf Controller

[line 40]

[illegible][illegible]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**
 1. **Dependencies Injection (DI)** - **Leaf** Controller
 2. **Leaf** Controller
- **Abstract Element**
- **Since** 1.0-dev

leaf_Controller::LEAF_CLASS_ID

= "LEAF_CONTROLLER-1_0_dev" [[line 44](#)]

leaf_Controller::LEAF_REG_KEY

= "controller" [[line 42](#)]

Constructor *void* function leaf_Controller::__construct() [[line 52](#)]

Leaf Controller

- **Access** public

void function leaf_Controller::__toString() [[line 57](#)]

- **Access** public

Class leaf_Dispatcher

[line 31]

[illegible]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0
- **Copyright** -
- **Since** 1.0-dev

leaf_Dispatcher::LEAF_CLASS_ID

```
= "LEAF_DISPATCHER-1_0_dev" [line 35]
```

leaf_Dispatcher::LEAF_REG_KEY

```
= "dispatcher" [line 33]
```

leaf_Dispatcher::\$controller

```
object leaf_Controller = NULL [line 58]
```

$$\hat{\Gamma}^{\dagger 1/2} \hat{\Gamma} \pm \hat{f} \hat{\Gamma}, \hat{\Gamma}^{\dagger 3/4} \hat{\Gamma}^{\dagger} \hat{\Gamma} \hat{\Gamma}^{\dagger}, \dots \hat{\Gamma} \hat{\Gamma}^{\dagger} \hat{\Gamma} \dots \hat{\Gamma} \hat{\Gamma}^{\dagger} \hat{\Gamma}, \hat{\Gamma} \hat{\Gamma}^{\dagger} \hat{\Gamma}^{\dagger} \hat{\Gamma} \hat{\Gamma}^{\dagger} \hat{\Gamma} \hat{\Gamma}^{\dagger} \hat{\Gamma} \dots \text{Controller.}$$

- **Access** private

leaf_Dispatcher::\$controllerName

```
string = NULL [line 51]
```

$\hat{\alpha} \in \mathbb{R}^{n_x}$, $\hat{\beta} \in \mathbb{R}^{n_y}$, $\hat{\gamma} \in \mathbb{R}^{n_z}$. Controller $K(s) = \frac{N(s)}{D(s)}$.

- **Access** private

leaf_Dispatcher::\$target

```
string = NULL [line 44]
```

[illegible]

- **Access** private

Constructor *void* function leaf_Dispatcher::__construct() [*line 68*]

[illegible]

- **Access** public

```
void function leaf_Dispatcher::dispatchController() [line 122]
```

[illegible]

- **Access** public

```
void function leaf_Dispatcher::__toString() [line 112]
```

- **Access** public

[line [36](#)]

- $$1. \quad \hat{1} \neq \hat{1} \gg \hat{1}_\zeta \mid \in \hat{1}_\zeta \hat{1}^{-1} \cdot \hat{1} f \hat{1}.$$

```
= "LEAF_ENDORSEMENTMANAGER-1_0_dev" [line 40]
```

```
= "endorse_man" [line 38]
```

```
array = array (
    "leaf_Locale", "leaf_Logger"
) [line 65]
```

[illegible]

- Generated by phpDocumentor v1.4.0 <http://www.phpdoc.org> - <http://pear.php.net/package/PhpDocumentor> - <http://www.sourceforge.net/projects/phpdocu> Page 32 of 169

leaf_EndorsementManager::\$currentClass

string = NULL [*line 74*]

if (\$this->currentClass == null) {
 \$this->currentClass = \$this->getEndorsedClass();
}

- **Access** private

leaf_EndorsementManager::\$endorsed

array = array() [*line 49*]

if (\$this->endorsed == null) {
 \$this->endorsed = \$this->getEndorsed();
}

- **Access** private

leaf_EndorsementManager::\$registeredEndorsed

array = array() [*line 57*]

if (\$this->registeredEndorsed == null) {
 \$this->registeredEndorsed = \$this->getRegisteredEndorsed();
}

- **Access** private

Constructor *void* function leaf_EndorsementManager::__construct() [*line 83*]

\$this->currentClass = null;
\$this->endorsed = null;
\$this->registeredEndorsed = null;

[illegible]

Function Parameters:

İbî¼âîµî,,î-î¶îĸî...î¼îµî,,îî, îîîî»İžîfîµîî,î,,îĸî...î±îî±îµîîĸî...î,,îî, îî»î-fîîî.

1. |¥|

- A** _____

Function Parameters:

$$\hat{\Gamma}^1 \gg |\mu|^{3/4} \pm \hat{\Gamma}^2, \quad |\hat{\Gamma}^3| \hat{\Gamma}^1 \pm \hat{\Gamma}^{-1/2} \hat{\Gamma}^1, \quad |\hat{\Gamma}^4| \gg |\hat{\Gamma}^1 f|, \quad |\hat{\Gamma}^5| \hat{\Gamma}^2 \dots |\hat{\Gamma}^6| \hat{\Gamma}^1, |\hat{\Gamma}^7| \hat{\Gamma}^1, |\hat{\Gamma}^8| \hat{\Gamma}^1, |\hat{\Gamma}^9| \mu \dots |\hat{\Gamma}^{10}| \hat{\Gamma}^1 f, \hat{\Gamma}^1 \hat{\Gamma}^1 \hat{\Gamma}^1 \mu, \hat{\Gamma}^1 \pm \hat{\Gamma}^1.$$

- **Access** public

void function leaf_EndorsementManager::loadEndorsedClass(\$className) [*line 120*]

Function Parameters:

- **\$className**

Î:ÏœÏ•Ï,,Ï%oÏ¼Ï± Ï,,Ï•Ï, Ï¶Ï•Ï,,Ï¸Ï•Ï¼ÏµÏ½Ï•Ï, Ï°Ï»Ï-ÏÏÏ•Ï,.

- **TODO**

1. Ï•Ï€Ï±Ï½Ï-Ï»ÏµÏ³Ï±Ï¸Ï, Ï,,Ï•Ï, Ï»ÏµÏ¹Ï,,Ï¸Ï...Ï³Ï-Ï±Ï, Ï°Ï±Ï,,Ï±Ï±Ï¸Ï•Ï.ÏÏÏ•Ï, Ï,,Ï%oÏ½ Ï°Ï»Ï-ÏÏÏµÏ%oÏ½ Ï°Ï±Ï, Ï¸Ï, Ï°Ï±Ï¹ Ï,,Ï%oÏ½ "Ï°Ï»ÏµÏ¹Ï¹Ï¸Ï½" Ï,,Ï¸Ï...Ï, Ï€Ï¸Ï... Ï±Ï•ÏµÏ¹ÏÏÏ¼Ï¸Ï€Ï¸Ï¹Ï¸Ï½ ÏÏÏ½, Ï±Ï½Ï,,Ï¹Ï°ÏµÏ¹Ï¼ÏµÏ½Ï¸
[leaf_Registry](#).

- **Access** public

Class leaf_Exception

[*line 39*]

Exception Class

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev

- [illegible]

- Constructor *void* function leaf_Exception::__construct(\$message, [\$code = 0]) [*line 41*]
- Function Parameters:**

- **Access** public

[line 35]

$$\hat{I} \circ \hat{I}^{\epsilon} \hat{I}^{\eta}, \hat{J} \mu \rangle \hat{I} \mu \hat{I}^{-1} \hat{I} \rangle \hat{I} \mu \hat{I}^{\eta}, \hat{I} \chi \dots \hat{I} \circ \hat{I}^3 \hat{I} \mu \hat{I}, \hat{I} \hat{I}^{-1} \frac{1}{4} \hat{I} \hat{I} \chi \dots \hat{I} \circ \hat{I}^3 \hat{I} \pm \hat{I}, \hat{I} f \hat{I} \dots \hat{I} \frac{1}{2} \hat{I} \hat{O} \hat{E} \hat{I} \hat{I} \mu \hat{I}^0 \hat{I} \frac{1}{2}.$$

- Generated by phpDocumentor v1.4.0 <http://www.phpdoc.org> - <http://pear.php.net/package/PhpDocumentor> - <http://www.sourceforge.net/projects/phpdocu> Page 36 of 169

- **Copyright -**

- **TODO**

1. `interface`.

- **Since** 1.0-dev

- **License -**

leaf_Hash::\$algorithms

`array = array ("md5", "sha1")` [[line 49](#)]

.

- **Access** private

leaf_Hash::\$defaultAlgorithm

`string = "md5"` [[line 42](#)]

.

- **Access** private

leaf_Hash::\$useAltMethod

`boolean = FALSE` [[line 56](#)]

- **Access** private

Constructor `void` function `leaf_Hash::__construct()` [[line 65](#)]

1&1-1/4!±!,,1&1!1&1,,1µ1-!,,1-1/2 1±!•1®1f1- 1® 1&11±1! 1,,1-1, 1/4!µ1,1&11!1&1... hash, 1-1/21±1/2!,,1! 1,,1%001/2 1®1»1±1f1f
 1f1...1/21±1•1,,1®1f1µ1%001/2 md5 1®1±1! sha1.

- **Access** public

```
string function leaf_Hash::digestFile($filename, [$raw = FALSE]) [line 112]
```

Function Parameters:

- *string* **\$filename**
- *boolean* **\$raw**

[illegible]

- **Access** public

```
string function leaf_Hash::digestMsg($data, [$raw = FALSE]) [line 95]
```

Function Parameters:

- *string* **\$data**
- *boolean* **\$raw**

[illegible]

- **Access** public

```
boolean function leaf_Hash::setAlgorithm($algorithm) [line 77]
```

Function Parameters:

- *string* \$algorithm

[illegible]

- **Access** public

Class leaf_Input

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**
 1. $\hat{\mathbb{M}} \times \hat{\mathbb{L}} \in \hat{\mathbb{L}} \times \hat{\mathbb{L}}$.
- **Since** 1.0-dev

leaf_Input::LEAF_CLASS_ID

```
= "LEAF_INPUT-1_0_dev" [line 50]
```

A unique id for each class.

$$\hat{\Gamma}^{\dagger}\hat{I}_{\frac{1}{2}}\hat{I}_{\pm}\hat{I}_{\frac{1}{4}}\hat{I}_{\pm}\hat{I}_{\frac{1}{2}}\hat{I}_{\pm}\hat{I}^{\dagger}|0\rangle_{CE} = \pm\hat{I}_{\frac{1}{2}}\hat{I}_{\pm}\hat{I}^{\dagger}\hat{I}_{\frac{1}{2}}|0\rangle_{CE} = \hat{I}^{\dagger}\hat{I}_{\frac{1}{2}}|\mu\rangle_{\mu} = -\hat{I}^{\dagger}\hat{f}\hat{I}.$$

leaf_Input::LEAF_REG_KEY

= "input" [[line 42](#)]

leaf_Input::\$input

array = array (
"get" => array(),"post"=>array()) [[line 58](#)]

- **Access private**

Constructor *void* function leaf_Input::__construct() [[line 63](#)]

- **Access public**

void function leaf_Input::offsetExists(\$offset) [[line 68](#)]

Function Parameters:

- **\$offset**

- **Access public**

void function leaf_Input::offsetGet(\$offset) [[line 73](#)]

Function Parameters:

- **\$offset**

- **Access public**

void function leaf_Input::offsetSet(\$offset, \$value) [*line* [78](#)]

Function Parameters:

- **\$offset**
- **\$value**

- **Access** public

void function leaf_Input::offsetUnset(\$offset) [*line* [83](#)]

Function Parameters:

- **\$offset**

- **Access** public

Class leaf_Loader

[*line* [33](#)]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -

- **TODO**

1. [Leaf Loader](#).

- **Since** 1.0-dev

leaf_Loader::LEAF_CLASS_ID

= "LEAF_LOADER-1_0_dev" [[line 37](#)]

leaf_Loader::LEAF_REG_KEY

= "loader" [[line 35](#)]

Constructor *void* function leaf_Loader::__construct() [[line 45](#)]

- **Access** public

void function leaf_Loader::endorsed(\$class) [[line 96](#)]

Function Parameters:

- *string* **\$class**

- **Access** public

void function leaf_Loader::extension(\$ext, [\$namespace = NULL]) [[line 85](#)]

Function Parameters:

- *string* **\$ext**
- *string* **\$namespace**

- **Access** public

void function leaf_Loader::library(\$name) [*line* [73](#)]

Function Parameters:

- *string* **\$name**

- **Access** public

void function leaf_Loader::model(\$modelName, [\$opts = NULL]) [*line* [63](#)]

Function Parameters:

- *string* **\$modelName**
- *array* **\$opts**

- **Access** public

void function leaf_Loader::plugin(\$plugin) [*line* [106](#)]

Function Parameters:

- *string* **\$plugin**

- **Access** public

void function leaf_Loader::__toString() [*line* [51](#)]

- **Access** public

Class leaf_Logger

[line 39]

- **Package** leaf
- **Sub-Package** core
- **Method** void log(): log(string message) log(string level, string message) log(string level, string message, string fileName) log(string level, string message, string fileName, string className)
Handles to log a message.
- **Author** Avraam Marimpis < makism@venus.cs.teicrete.gr>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**
 1. **leaf_Logger::LEAF_CLASS_ID**
= "LEAF_LOGGER-1_0_dev" [line 43]
- **Since** 1.0-dev
- **License** -

leaf_Logger::LEAF_CLASS_ID

= "LEAF_LOGGER-1_0_dev" [line 43]

leaf_Logger::LEAF_REG_KEY

= "logger" [line 41]

leaf_Logger::\$buffer

array = NULL [line 65]

- **Access** private

leaf_Logger::\$filename

string = NULL [[line 51](#)]

- **Access** private

leaf_Logger::\$fp

resource = NULL [[line 86](#)]

- **Access** private

leaf_Logger::\$levels

array = array("All", "Debug", "Warning", "Info", "None") [[line 79](#)]

- **Access** private

leaf_Logger::\$stampPat

string = NULL [[line 58](#)]

- **Access** private

leaf_Logger::\$threshold

int = NULL [[line 72](#)]

- **Access** private

Constructor *void* function leaf_Logger::__construct([\$buffer = false]) [*line* [96](#)]

Function Parameters:

- *boolean* **\$buffer**

Class constructor

- **Access** public

void function leaf_Logger::end_flush() [*line* [157](#)]

end_flush

- **Access** public

array|NULL function leaf_Logger::flush([\$return = false]) [*line* [138](#)]

Function Parameters:

- *boolean* **\$return**

flush

- **Access** public

boolean function leaf_Logger::setLevel(\$filterLevel) [*line* [186](#)]

Function Parameters:

- *string* **\$filterLevel**

setLevel

Set the level of errors to capture.

~~~~~

- **Access** public

*void* function leaf\_Logger::setTimeStampPattern(\$pat) [*line* [171](#)]

**Function Parameters:**

- *string* **\$pat**

## setTimeStampPattern

- **Access** public

*void* function leaf\_Logger::\_\_call(\$method, \$args) [*line* [204](#)]

**Function Parameters:**

- *string* **\$method**
- *array* **\$args**

## \_\_call

- **Access** public

## Class leaf\_Model

[line [33](#)]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**
  1.  $\hat{\mathbb{N}} \times \hat{\mathbb{I}}_0 \in \hat{\mathbb{I}}_0 \hat{\mathbb{I}} \cdot \hat{\mathbb{I}} f \hat{\mathbb{I}}.$
- **Abstract Element**
- **Since** 1.0-dev

**leaf\_Model::LEAF\_CLASS\_ID**

= "LEAF\_MODEL-1\_0\_dev" [line [37](#)]

**leaf\_Model::LEAF\_REG\_KEY**

= "model" [line [35](#)]

Constructor *void* function leaf\_Model::\_\_construct() [line [45](#)]

- **Access** public

[line 42]

İl...İl,® İ. İl»İ-İfİ. İ...İ»İçİ€İçİİİµİ- İl,İ İ€İ•İœİ,,İ...İ€İç İfİİİ-İİİİç Registry. İœİµ İ±İ...İ,İœ İ,,İ ±İİ½İ,,İİİµİİİ¼İµİ½İ± İ±İ€İçİ İ-İİµİ•İçİ½İ,,İ±İİ İfİİİµİfİİİ±İfİİİ±İİ- İ¼İ-İfİ± İfİ,,İ-İ½ İİ»İ-İfİ- İ±İ...İ,İ®, İ±İ½İ±İİİµİ•İœİ¼İ±İfİ,,İµ İ¼İ-İfİ% İ±İ...İ,İ®,

- [illegible]

- **Since 1.0-dev**

```
object leaf_Registry = NULL [line 58]
```

- **Static**
- **Access** private

```
array = array() [line 50]
```



- **Access** private

Constructor *void* function leaf\_Registry::\_\_construct() [*line 68*]

### **Class constructor**

en: Does nothing...

- **Access** private

*object leaf\_Registry* function leaf\_Registry::getInstance() [*line 90*]

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

*object Register* function leaf\_Registry::instance() [*line 79*]

### **instance**

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

*boolean* function leaf\_Registry::isRegistered(\$key) [*line 176*]

### **Function Parameters:**

- *string* **\$key**

### **isRegistered**

- **Access** public

```
void function leaf_Registry::register($obj) [line 149]
```

**Function Parameters:**

- *object* **\$obj**

**register**

- **TODO**

- [illegible]

- **Access** public

array function leaf\_Registry::toArray() [line 190]

- **Access** public

```
void function leaf_Registry::unregister($class) [line 164]
```

**Function Parameters:**

- *string* \$class

- **Access** public

*mixed* function leaf\_Registry::\_\_get(\$key) [line 110]

**Function Parameters:**

- *string* **\$key**

### **\_\_get**

It is a magic method. Allows external access to this object's properties. We control the properties' visibility.

```
1      $instance-> exampleProperty
```

- **Access** public

*void* function leaf\_Registry::\_\_set(\$key, \$obj) [line 129]

**Function Parameters:**

- *string* **\$key**
- *object* **\$obj**

### **\_\_set**

This "magic method" completes the actions that performs the other "**\_\_get**" method. It is used to set a value to a property.

- **Access** private

# Class leaf\_Request

[line 45]

Î Î±•Î±Î±Î±<sup>1</sup> Î€•ÎœÎ±Î±Î±<sup>2</sup>Î±Î±Î±. Î±Î± ÎœÎ±»Î± Î±,Î± Î±Î±Î±Î±Î±Î±Î±Î±,Î± Î±Î±,Î±Î±Î±Î±Î±Î±Î± Î±,Î±Î±Î±... Uri.

[illegible][illegible]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**

[illegible]

- **Since** 1.0-dev
- **License** -

leaf\_Request::LEAF\_CLASS\_ID

```
= "LEAF REQUEST-1 0 dev" [line 49]
```

leaf\_Request::LEAF\_REG\_KEY

```
= "request" [line 47]
```

leaf\_Request::\$action

```
string = NULL [line 91]
```

[illegible]

- **Access** private

leaf\_Request::\$controller

```
string = NULL [line 72]
```

[illegible]

- **Access** private

## leaf\_Request::\$controllerFile

```
string = NULL [line 84]
```

**Controller.**

12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697989910010110210310410510610710810911011111211311411511611711811912012112212312412512612712812913013113213313413513613713813914014114214314414514614714814915015115215315415515615715815916016116216316416516616716816917017117217317417517617717817918018118218318418518618718818919019119219319419519619719819920020120220320420520620720820921021121221321421521621721821922022122222322422522622722822923023123223323423523623723823924024124224324424524624724824925025125225325425525625725825926026126226326426526626726826927027127227327427527627727827928028128228328428528628728828929029129229329429529629729829930030130230330430530630730830931031131231331431531631731831932032132232332432532632732832933033133233333433533633733833934034134234334434534634734834935035135235335435535635735835936036136236336436536636736836937037137237337437537637737837938038138238338438538638738838939039139239339439539639739839940040140240340440540640740840941041141241341441541641741841942042142242342442542642742842943043143243343443543643743843944044144244344444544644744844945045145245345445545645745845946046146246346446546646746846947047147247347447547647747847948048148248348448548648748848949049149249349449549649749849950050150250350450550650750850951051151251351451551651751851952052152252352452552652752852953053153253353453553653753853954054154254354454554654754854955055155255355455555655755855956056156256356456556656756856957057157257357457557657757857958058158258358458558658758858959059159259359459559659759859960060160260360460560660760860961061161261361461561661761861962062162262362462562662762862963063163263363463563663763863964064164264364464564664764864965065165265365465565665765865966066166266366466566666766866967067167267367467567667767867968068168268368468568668768868969069169269369469569669769869970070170270370470570670770870971071171271371471571671771871972072172272372472572672772872973073173273373473573673773873974074174274374474574674774874975075175275375475575675775875976076176276376476576676776876977077177277377477577677777877978078178278378478578678778878979079179279379479579679779879980080180280380480580680780880981081181281381481581681781881982082182282382482582682782882983083183283383483583683783883984084184284384484584684784884985085185285385485585685785885986086186286386486586686786886987087187287387487587687787887988088188288388488588688788888989089189289389489589689789889990090190290390490590690790890991091191291391491591691791891992092192292392492592692792892993093193293393493593693793893994094194294394494594694794894995095195295395495595695795895996096196296396496596696796896997097197297397497597697797897998098198298398498598698798898999099199299399499599699799899910001001100210031004100510061007100810091010101110121013101410151016101710181019102010211022102310241025102610271028102910301031103210331034103510361037103810391040104110421043104410451046104710481049105010511052105310541055105610571058105910601061106210631064106510661067106810691070107110721073107410751076107710781079108010811082108310841085108610871088108910901091109210931094109510961097109810991100110111021103110411051106110711081109111011111112111311141115111611171118111911201121112211231124112511261127112811291130113111321133113411351136113711381139114011411142114311441145114611471148114911501151115211531154115511561157115811591160116111621163116411651166116711681169117011711172117311741175117611771178117911801181118211831184118511861187118811891190119111921193119411951196119711981199120012011202120312041205120612071208120912101211121212131214121512161217121812191220122112221223122412251226122712281229123012311232123312341235123612371238123912401241124212431244124512461247124812491250125112521253125412551256125712581259126012611262126312641265126612671268126912701271127212731274127512761277127812791280128112821283128412851286128712881289129012911292129312941295129612971298129913

- **Access** private

leaf\_Request::\$queryElems

```
array = NULL [line 99]
```

[illegible]

- **Access** private

**leaf\_Request::\$segments**

*array* = NULL [*line 64*]

if (\$this->isAjaxRequest()) {  
 \$this->segments = array();  
 \$this->segments[] = \$this->getControllerFileName();  
 \$this->segments[] = \$this->getControllerName();  
 \$this->segments[] = \$this->getActionName();  
 \$this->segments[] = \$this->getQueryString();  
}

- **Access** private

Constructor *void* function leaf\_Request::\_\_construct() [*line 110*]

if (\$this->isAjaxRequest()) {  
 \$this->segments = array();  
 \$this->segments[] = \$this->getControllerFileName();  
 \$this->segments[] = \$this->getControllerName();  
 \$this->segments[] = \$this->getActionName();  
 \$this->segments[] = \$this->getQueryString();  
}

- **Access** public

*string* function leaf\_Request::getControllerFileName() [*line 168*]

- **Access** public

*string* function leaf\_Request::getControllerName() [*line 159*]

- **Access** public

*mixed* function leaf\_Request::getQueryString(\$offset) [*line 265*]

**Function Parameters:**

- **string \$offset**

- **TODO**

1. `if ($offset > 0) {`

- **Access public**

*string* function leaf\_Request::getQueryStringAsString() [*line 284*]

- **TODO**

1. `if ($offset > 0) {`

2. `if ($offset < 0) {`

- **Access public**

*string* function leaf\_Request::reconstructUrl([\$className = NULL], [\$methodName = NULL], [\$segments = NULL],  
[\$queryString = NULL]) [*line 201*]

**Function Parameters:**

- **string \$className**
- **string \$methodName**
- **array \$segments**
- **array \$queryString**

- **TODO**

1. `if ($className) {`

- **Access** public

```
void function leaf_Request::redirect($target, [$isExternal = FALSE]) [line 183]
```

### Function Parameters:

- *string* **\$target**
- *boolean* **\$isExternal**

- **TODO**

1.  $\forall i \in I, i \in I \rightarrow f(i) \in I$ .

- **Access** public

```
string|NULL function leaf_Request::segment($n) [line 235]
```

### Function Parameters:

- *integer \$n*

- **TODO**

1.  $\forall i \in I, i \in I \rightarrow f(i) \in I$ .

- **Access** public

array function leaf\_Request::segmentsToArray() [line 250]

- **TODO**

$$1. \quad \hat{\mathbb{M}} \gg \hat{\mathbb{L}} \in \hat{\mathbb{L}}^{-1} \cdot \hat{f}.$$



- **Access** public

*integer* function leaf\_Request::totalSegments() [*line* [220](#)]

- **TODO**

1.  $\hat{I} \neq \hat{I} \gg \hat{I} \in \hat{I} \in \hat{I} \cdot \hat{I} f \hat{I} \cdot$

- **Access** public

*void* function leaf\_Request::\_\_toString() [*line* [150](#)]

- **Access** public

## Class leaf\_Response

[*line* [37](#)]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**
- 1.  $\hat{I} \neq \hat{I} \gg \hat{I} \in \hat{I} \in \hat{I} \cdot \hat{I} f \hat{I} \cdot$
- **Since** 1.0-dev



- **Access** public

*void* function leaf\_Response::clearHeaders() [*line* [146](#)]

- **Access** public

*void* function leaf\_Response::flushOutputBuffer() [*line* [207](#)]

- **Access** public

*string* function leaf\_Response::getOutputBuffer([\$flush = FALSE]) [*line* [194](#)]

**Function Parameters:**

- **\$flush**

**Gets output buffer.**

- **Access** public

*void* function leaf\_Response::ouputBufferingStart() [*line* [168](#)]

**Start output buffering**

- **Access** public

*void/string* function leaf\_Response::outputBufferingEnd([\$returnBuffer = FALSE]) [*line* [183](#)]

**Function Parameters:**

- *boolean* **\$returnBuffer**

## End output buffering

- **Access** public

*void* function leaf\_Response::sendResponse() [*line* [157](#)]

- **Access** public

*void* function leaf\_Response::setHeader(\$name, \$value) [*line* [142](#)]

**Function Parameters:**

- *string* **\$name**
- *string* **\$value**

- **Access** public

*void* function leaf\_Response::\_\_toString() [*line* [130](#)]

- **Access** public

## Class leaf\_Router

[line 45]

[illegible]

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**

1.  $\text{leaf\_Request};$
2.  $\text{leaf\_Request};$

- **Since 1.0-dev**

leaf\_Router::LEAF\_CLASS\_ID

```
= "LEAF_ROUTER-1_0_dev" [line 49]
```

leaf\_Router::LEAF\_REG\_KEY

```
= "router" [line 47]
```

## leaf\_Router::\$queryString

```
string = NULL [line 85]
```

- **Access** private

### **leaf\_Router::\$queryStringElements**

*array* = NULL [[line 92](#)]

- **Access** private

### **leaf\_Router::\$requestClass**

*string* = NULL [[line 64](#)]

- **Access** private

### **leaf\_Router::\$requestMethod**

*string* = NULL [[line 71](#)]

- **Access** private

### **leaf\_Router::\$requestUri**

*string* = NULL [[line 57](#)]

- **Access** private

### **leaf\_Router::\$segments**

*array* = NULL [[line 78](#)]

- **Access private**

Constructor *void* function `leaf_Router::__construct()` [line 108]

**Leaf Router Constructor**

- See [leaf\\_Config](#)
- See [leaf\\_Request](#)
- **TODO**

1. "Leaf Router Constructor" is a class constructor. It is a function that is called when a new instance of the class is created. It is used to initialize the class's properties and methods.
2. The constructor function is defined in the class file. It is a function that is called when a new instance of the class is created. It is used to initialize the class's properties and methods.

- **Access public**

*void* function `leaf_Router::chopSegment(&$seg, $seg)` [line 329]

**Function Parameters:**

- *string* `$seg`
- `&$seg`

**Leaf Router chopSegment()** This function is used to chop a segment from a URI. It takes two parameters: `&$seg` and `$seg`. The function returns the chopped segment.

- **Access private**

*string* function `leaf_Router::getClassName()` [line 345]

**Leaf Router getClassName()** This function is used to get the class name of a controller. It returns the class name as a string.

- **Access** public

*string* function leaf\_Router::getMethodName() [*line* [356](#)]

[illegible]

- **Access** public

array/NULL function leaf\_Router::queryStringElements() [line 295]

[illegible]

- **Access** public

array/NULL function leaf\_Router::segments() [line 317]

$\{ \hat{0} \in \hat{1}^i f_i, \hat{0} \in \hat{1}^i \mu_i \mid \hat{1}^{-1/2} \hat{1}^{\pm 1/2} \in \hat{1}^{-1/2} \hat{1}^{\pm 1/2} \mid \hat{1}^4 \mu_i, \hat{1}_{\pm} \mid \mu_i \in \hat{1}^i \in \rangle \hat{1} \hat{1} \hat{1}^{1/2} \mid f_i, \hat{1} \hat{1}^i \hat{1}^{\pm 1} \mu_i \hat{1}^{\pm 1} \mid f_i, \hat{1} \hat{1} \mid \text{Uri} \mid \mu_i \hat{0} \mid \hat{0} \rangle \hat{1} \hat{1} f_i \hat{1}, \hat{1}_{\pm} \hat{1}, \hat{1}^4 \mu_i \mid \hat{0} \hat{1} \hat{1} \hat{1} \hat{1} \dots \hat{0} \hat{1}^{\pm 1} \hat{1}_{\pm} \hat{1} \hat{1} \dots \text{query string} \}.$

- **Access** public

*integer* function leaf\_Router::segmentsSize() [*line* [305](#)]

[illegible]



- **Access** public

```
void function leaf_Router::__toString() [line 284]
```

- **Access** public

## Class leaf\_View

- **Package** leaf
- **Sub-Package** core
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**

$$1. \quad \hat{A} \neq \hat{B} \Rightarrow \hat{A} \in \hat{A}^{-1} \cdot \hat{B}.$$

- **Since 1.0-dev**

leaf\_View::LEAF\_CLASS\_ID

```
= "LEAF_VIEW-1_0_dev" [line 37]
```

leaf View::LEAF REG KEY

```
= "view" [line 35]
```

Constructor *void* function leaf\_View::\_\_construct() [line 45]

- **Access** public

*void* function leaf\_View::\_\_call(\$method, \$args) [*line* [61](#)]

***Function Parameters:***

- **\$method**
- **\$args**

- **Access** public

*void* function leaf\_View::\_\_toString() [*line* [51](#)]

- **Access** public

# Benchmark.php

## leaf Framework

*PHP version 5*

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Framework PHP5. Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση.

- **Package** leaf
- **Sub-Package** core-helpers
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

# Request.php

## leaf Framework

*PHP version 5*

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

Απλό, εύκολο, γρήγορο, με μικρό footprint και εύκολο στην επέκταση. Το πρώτο ελληνικό open source PHP5 framework, γρήγορο, με μικρό footprint και εύκολο στην επέκταση.

- **Package** leaf
- **Sub-Package** core-helpers
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

# Hook.php

## leaf Framework

PHP version 5

leaf framework, is a small, easy to use, and easily extensible framework, with small footprint and easily extensible.

An open source framework, small, with small footprint and easily extensible.

- **Package** leaf
- **Sub-Package** core-hook
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

# Hook\_Conditional.php

## leaf Framework

PHP version 5

leaf framework, a small, with small footprint and easily extensible.

An open source framework, small, with small footprint and easily extensible.

- **Package** leaf
- **Sub-Package** core-hook
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

## Class leaf\_Hook

[line [33](#)]

- **Package** leaf
- **Sub-Package** core-hook
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**

1. leaf\_Hook class.

- **Abstract Element**
- **Since** 1.0-dev

**leaf\_Hook::LEAF\_CLASS\_ID**

= "LEAF\_HOOK-1\_0\_dev" [[line 37](#)]

**leaf\_Hook::LEAF\_REG\_KEY**

= "hook" [[line 35](#)]

Constructor *void* function leaf\_Hook::\_\_construct() [[line 45](#)]

- **Access** public

*string* function leaf\_Hook::\_\_toString() [[line 55](#)]

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

## Class leaf\_Hook\_Conditional

[[line 33](#)]

- **Package** leaf
- **Sub-Package** core-hook
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **TODO**

$$1. \quad \hat{\mathbb{Y}} \hat{\mathbb{I}} \gg \hat{\mathbb{I}}_{\mathbb{Z}} \hat{\mathbb{I}} \in \hat{\mathbb{I}}_{\mathbb{Z}} \hat{\mathbb{I}}^{-1} \hat{\mathbb{I}} \hat{f} \hat{\mathbb{I}}.$$

- **Abstract Element**
- **Since** 1.0-dev

leaf\_Hook\_Conditional::LEAF\_CLASS\_ID

```
= "LEAF_HOOK_CONDITIONAL-1_0_dev" [line 37]
```

leaf\_Hook\_Conditional::LEAF\_REG\_KEY

```
= "conditional_hook" [line 35]
```

Constructor *void* function leaf\_Hook\_Conditional::\_\_construct() [[line 45](#)]

- **Access** public

```
boolean function leaf_Hook_Conditional::condition() [line 55]
```

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



## Hooks.php

## leaf Framework

*PHP version 5*

The first greek open source PHP5 framework, fast, with small footprint and easily extensible.

[illegible]

- **Package** leaf
- **Sub-Package** core-hook-helpers
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **Filesource** [Source Code for this file](#)
- **License** -

```
array function getHooks([$level = NULL], [$controller = NULL], [$method = NULL]) [line 109]
```

### Function Parameters:

- *string|NULL \$level*
- *string|NULL \$controller*
- *string|NULL \$method*

**Returns all registered hooks for one -or all- level. Also, the hooks can be filtered using the parameters \$controller and \$method.**

[illegible]

```
HOOK_POST_CONTROLLER_DISPATCH = 'post_controller_dispatch' [line 34]
```

```
HOOK_POST_FRONT_CONTROLLER = 'post front controller' [line 40]
```

```
HOOK PRE CONTROLLER DISPATCH = 'pre controller dispatch' [line 28]
```

array function introspectHooks([\$level = NULL]) [[line 51](#)]

**Function Parameters:**

- *string* **\$level**

- **Access private**

void function runHook(\$controller, \$method) [[line 91](#)]

**Function Parameters:**

- *string* **\$controller**
- *string* **\$method**

**Executes a specific hook.**

Î•Î¶Î„ÎµÎ»ÎµÎ¬ Î¬Î½Î±Î½ Î¶Î…Î³Î¶Î¶Î¶Î¶Î¶Î½Î¬Î½Î½ hook.

- **Access private**

void function runHooks(\$level) [[line 77](#)]

**Function Parameters:**

- *string* **\$level**

**Runs all registered hooks in one level.**

Î•Î¶Î„ÎµÎ»ÎµÎ¬ ÎœÎ»Î½Î…Î½, Î½Î½Î½Î…Î½, hooks Î¶Îµ Î¬Î½Î± ÎµÎ¶Î¶ÎµÎ¶Î½Î½.

- **Access private**

[line 24]

PHP version 5

$\{ \hat{\mu}_i \}_{i \in \mathbb{I}} \subseteq \mathbb{R}^n$  and  $\{ \hat{\sigma}_i \}_{i \in \mathbb{I}} \subseteq \mathbb{R}^n$  are the mean and standard deviation of the random variables  $\hat{\mu}_i$  and  $\hat{\sigma}_i$  respectively. The framework PHP5 [14] is a framework for the analysis of the data. The framework PHP5 is a framework for the analysis of the data. The framework PHP5 is a framework for the analysis of the data.

- **Package** leaf
- **Sub-Package** core-log
- **Author** Avraam Marimpis < [makism@venus.cs.teicrete.gr](mailto:makism@venus.cs.teicrete.gr)>
- **Version** 1.0-dev
- **Copyright** -
- **License** -

# Debug.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Debug.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **TODO**
  1. Document.
  2. Recheck the information displayed under the tab "Config Settings".
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

# Front\_Controller.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Front\_Controller.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **TODO**
  1. Add functionality for the 'hooks' subsystem.
  2. Add functionality for the benchmarking subsystem.
  3. Completion of the documentation.
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

`require_once LEAF_BASE.'core/hook/helpers/Hooks.php'` [*line 64*]

## Helper functions handling hooks.

- See [leaf\\_Hook\\_Conditional](#)
- See [leaf\\_Hook](#)

`require_once LEAF_BASE.'front/Debug.php'` [*line 168*]

`require_once LEAF_BASE.'front/helpers/Debug.php'` [[line 55](#)]

### **Custom debug functions.**

`require_once LEAF_BASE.'front/helpers/Error.php'` [[line 49](#)]

### **Functions used to present errors.**

`require_once LEAF_BASE.'front/helpers/Dependancies.php'` [[line 37](#)]

### **Helper functions that are used to declare dependancies on specific extensions, functions or leaf's Classes.**

`require_once LEAF_BASE.'front/helpers/Handlers.php'` [[line 43](#)]

### **Custom error and exception handling functions.**

`require_once LEAF_BASE.'front/helpers/Main.php'` [[line 30](#)]

### **Helper functions needed to start-up the framework and to load its classes.**

# Debug.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front-helpers
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Debug.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **TODO**
  1. Document.
  2. Possible removal.
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

*void function inspect\_var(\$var, [\$depth = NULL]) [line 50]*

**Function Parameters:**

- *mixed* **\$var**
- *boolean|integer* **\$depth**

*void function i\_var(\$var, [\$depth = NULL]) [line 38]*

**Function Parameters:**

- *mixed* **\$var**
- *boolean|integer* **\$depth**

*mixed* function \_inspect\_var\_Array(\$arr, [\$recursionLevel = 5]) [*line* [173](#)]

**Function Parameters:**

- *array* **\$arr**
- *integer* **\$recursionLevel**

- **Access private**

*string* function \_inspect\_var\_Boolean(\$flag) [*line* [156](#)]

**Function Parameters:**

- *boolean* **\$flag**

- **Access private**

*void* function \_inspect\_var\_Generic(\$var) [*line* [112](#)]

**Function Parameters:**

- *mixed* **\$var**

- **Access private**

*void* function \_inspect\_var\_Number(\$num) [*line* [124](#)]

**Function Parameters:**

- *integer|double* **\$num**



- **Access** private

*void* function `_inspect_var_Object($obj, [$inDepth = FALSE])` [*line* [187](#)]

**Function Parameters:**

- *object* **\$obj**
- *boolean* **\$inDepth**

- **Access** private

*string* function `_inspect_var_Object_getSubclasses($obj)` [*line* [333](#)]

**Function Parameters:**

- *object* **\$obj**

- **Access** private

*void* function `_inspect_var_Resource($var)` [*line* [95](#)]

**Function Parameters:**

- *resource* **\$var**

- **Access** private

`void function _inspect_var_String($str) [line 139]`

**Function Parameters:**

- *string* **\$str**
- **Access** private

# Dependancies.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front-helpers
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Dependancies.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **TODO**
  1. Implement a function that will handle dependancies as optional.
  2. Recheck **all** functions.
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

*boolean* function dependsOn(\$deps) [[line 32](#)]

**Function Parameters:**

- *string|array* **\$deps**

**Handles dependacies on PHP extensions.**

*boolean* function dependsOnFunc(\$func, \$funcs) [[line 75](#)]

**Function Parameters:**

- *string* **\$funcs**

- **\$func**

**Handles dependencies on specific functions.**

*boolean* function dependsOptionalOn(\$deps) [[line 64](#)]

**Function Parameters:**

- *string* **\$deps**

**Handles optional dependencies.**

- **TODO**

1. Implementd.

# Error.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front-helpers
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Error.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **TODO**
  1. Implement a function that will handle dependencies as optional.
  2. Recheck **all** functions.
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

*void* function showErrorPage404([\$str = NULL]) [*line* [36](#)]

**Function Parameters:**

- *string* \$str

**Presents a "Page Not Found" message.**

- **TODO**
  1. Implement.

*void* function showHtmlMessage(\$title, [\$str = NULL], [\$die = FALSE]) [*line* [49](#)]

***Function Parameters:***

- *string* **\$title**
- *string* **\$str**
- *boolean* **\$die**

**Prints styled text. Used in printing debug messages and errors.**

# Handlers.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front-helpers
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Handlers.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **TODO**
  1. Implement a function that will handle dependencies as optional.
  2. Recheck **all** functions.
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

*void* function errorHandler(\$errno, \$errstr, \$errfile, \$errline) [*line* [35](#)]

## **Function Parameters:**

- *integer* **\$errno**
- *string* **\$errstr**
- *string* **\$errfile**
- *integer* **\$errline**

**Custom error handler.**

*void* function `exceptionHandler($ex)` [*line* [70](#)]

***Function Parameters:***

- *object Exception* **\$ex**

**Custom exception handler.**



# Main.php

This source file is part of the leaf framework and is licensed under the New BSD license.

For the full copyright and license information, please view the LICENSE file that was distributed with this source code.

- **Package** leaf
- **Sub-Package** front-helpers
- **Author** Avraam Marimpis < [makism@users.sf.net](mailto:makism@users.sf.net)>
- **Version** \$Id: Main.php 24 2007-11-04 19:20:06Z makism \$
- **Copyright** Copyright (c) 2007 Avraam Marimpis
- **Link** <http://leaf-framework.sourceforge.net>
- **Filesource** [Source Code for this file](#)
- **License** [New BSD License](#)

*void function frameworkForceTerminate() [line [124](#)]*

**Terminates the execution of the framework.**

*void function frameworkSanityCheck([\$checkAll = false]) [line [136](#)]*

**Function Parameters:**

- *boolean* **\$checkAll**

**Performs some basic tests to check whether the framework is usable or not.**

*void function \_\_autoload(\$className) [line [33](#)]*

**Function Parameters:**

- *string* **\$className**

**Includes all necessary files related to leaf's classes.**

We use the "magic" function `__autoload` in order to have leaf's classes loaded automatically.

Also, `leaf_EndorsedManager` is instantiated if needed.

- **See** `leaf_EndorsedManager`
- **Link** <http://www.php.net/manual/en/language.oop5.autoload.html>

# Appendices

# Appendix A - Class Trees

## Package leaf

### leaf\_Base

- [leaf\\_Base](#)
  - [leaf\\_Benchmark](#)
  - [leaf\\_Config](#)
  - [leaf\\_Controller](#)
  - [leaf\\_Dispatcher](#)
  - [leaf\\_EndorsementManager](#)
  - [leaf\\_Input](#)
  - [leaf\\_Loader](#)
    - [leaf\\_Hook](#)
      - [leaf\\_Hook\\_Conditional](#)
- [leaf\\_Logger](#)
- [leaf\\_Model](#)
- [leaf\\_Request](#)
- [leaf\\_Response](#)
- [leaf\\_Router](#)
- [leaf\\_View](#)

### leaf\_Exception

- Exception
  - [leaf\\_Exception](#)

### leaf\_Hash

- [leaf\\_Hash](#)

## leaf\_Log

- [leaf\\_Log](#)

## leaf\_Registry

- [leaf\\_Registry](#)

# Appendix B - README/CHANGELOG/INSTALL

# TODO

## LICENSE

Copyright (c) 2007, Avraam Marimpis  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of leaf nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
4. Products derived from this software may not be called "leaf" nor may "leaf" appear in their names without specific prior written permission from the author.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## README

leaf framework

\*\*\*\*\*

- \* Supported PHP versions, 5.1 - or newer.
- \* This is a testing release, which means that.
  - 1) The API is not final, and maybe change.
  - &
  - 2) Since this is a test release, it is not wise to be used in production.

## CHANGELOG

## INSTALL

There are no available information.



# Appendix C - Source Code

## Package leaf

# File Source for Benchmark.php

Documentation for this file is available at [Benchmark.php](#)

```
1  <?php
2  /**
3   * leaf Framework
4   *
5   * <i>PHP version 5</i>
6   *
7   *
8   * The first greek open source PHP5 framework, fast, with small footprint and
9   * easily extensible.<br>
10  * ίνίε ίεί.ίζί.ίί ίμί»ί»ί.ί%ί¹ί°ίε framework PHP5 ί±ί%ίίί°ί°ί.ί. ί°ίζί.ί¹ί°ί±, ί³ί.ί@ί³ίίί.ίί, ί%ί¹ί°ί.ίε ίf
11  * ί°ί±ί¹ ίμί.ί°ίίί»ί± ίμίείμί°ί.ί-ίfί¹ί%ίί.<br>
12  *
13  *
14  * @package      leaf
15  * @subpackage    core
16  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
17  * @copyright     -
18  * @license       -
19  * @version       1.0-dev
20  * @filesource
21  */
22
23
24  /**
25  *
26  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
27  * @copyright     -
28  * @license       -
29  * @version       1.0-dev
30  * @since         1.0-dev
31  * @link          http://www.php.net/~helly/php/ext/spl/interfaceArrayAccess.html
32  * @todo
33  * <ol>
34  * <li>ί%ί»ίίίείί-ί.ίfί¹.</li>
35  * </ol>
36  */
37  final class leaf_Benchmark extends leaf_Base {
38
39      const LEAF_REG_KEY = "benchmark" ;
40
41      const LEAF_CLASS_ID = "LEAF_BENCHMARK-1_0_dev" ;
42
43      /**
44       *
45       *
46       *
47       * @var array $indexTable
48       */
49      private $indexTable = array();
50
51
52      /**
53       *
54       *
55       * @return void
56       */
57      public function __construct()
58      {
59          parent::__construct(self::LEAF_REG_KEY);
60
61          dependsOptOn("xdebug" );
62      }
63
64      /**
65       *
66       *
```

```
67      * @param string $name
68      * @param integer $type
69      * @return void
70      */
71      public function addIndex($name, $type)
72      {
73
74      }
75
76  }
77
78  ?>
```

## File Source for Controller.php

Documentation for this file is available at [Controller.php](#)

[illegible]

# File Source for Dispatcher.php

Documentation for this file is available at [Dispatcher.php](#)

[illegible]



# File Source for EndorsementManager.php

Documentation for this file is available at [EndorsementManager.php](#)

[illegible]





```

144      *      <li>î^î»îµî³î+î¿î, î³î¹î± î„î¿ î¬î¼ î· î°î»î¬îîî· î€î¿î… îµî»î-î³î+î¿î…î¼îµ, î-î+îµî¹
î~î·î»î%î¼î-î¼î± î„î¹î,
145      *      îîî„î±î,îµî·î-î, LEAF_REG_KEY î@/î°î±î¹ LEAF_CLASS_ID.</li>
146      *      </ol>
147      */
148      private function introspectEndorsedClass($className, $fileName)
149      {
150
151      }
152
153      /**
154      * î·î€î¹îîî„î·î¬îµî¹ î„î¹î, î°î»î¬îîµî¹, î€î¿î… î-î+î¿î…î¼ î…î€îµî·î°î±î»î·î^îµî¹ î„î¹î, î±î¼î„î¬îî„î
î€î…î·î@î¼î±.
155      *
156      * @return array
157      */
158      public function getEndorsedClasses()
159      {
160          return $this-> endorsed;
161      }
162
163      }
164
165      ?>

```

# File Source for Exception.php

Documentation for this file is available at [Exception.php](http://www.phpdoc.org)

```
1  <?php
2  /**
3   * leaf Framework
4   *
5   * <i>PHP version 5</i>
6   *
7   *
8   * The first greek open source PHP5 framework, fast, with small footprint and
9   * easily extensible.<br>
10  * ίνίζ ίεί.ίζί.ίζ ίμί»ί»ί.ί%ί°ί@ framework PHP5 ί±ί%ίζίί°ίζί. ί°ίζί.ί°ί±, ί°ί.ί@ί°ίζί.ίζ, ί%ί°ί.ί@ ίf
11  ί%ί-ί°ίμí.ίζί,
12  * ί°ί±ί°ί ίμí.ί°ίζί»ί± ίμíείμí°ίζί.ί-ίfί°ί%ίζ.<br>
13  *
14  * @package      leaf
15  * @subpackage    core
16  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
17  * @copyright     -
18  * @license       -
19  * @version       1.0-dev
20  * @filesource
21  */
22
23
24  /**
25   * Exception Class
26   *
27   *
28   * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
29   * @copyright     -
30   * @license       -
31   * @version       1.0-dev
32   * @since         1.0-dev
33   * @link          http://php.net/manual/en/language.exceptions.html
34   * @todo
35   * <ol>
36   * <li>ί ίί.ί±ί%ί@ ίμífί%ί.ίμí.ί°ί@ ί...ίείζίfίζί.ί@ί.ί°ί%ί. ί°ίί.ί±ί°ί.ί±ί@ί, ί.ί%ί%
37  ίμí%ί±ί°ί.ί-ίfίμí%ί.</li>
38   * </ol>
39   */
40  final class leaf_Exception extends Exception {
41      public function __construct($message, $code = 0)
42      {
43          parent::__construct($message, $code);
44      }
45  }
46
47
48  ?>
```

# File Source for Hash.php

Documentation for this file is available at [Hash.php](http://www.phpdoc.org/package/PhpDocumentor)

```
1  i»¿<?php
2  /**
3   * leaf Framework<br>
4   *
5   * PHP version 5<br>
6   *
7   * f^f%î± î±f%î¿î¹î°î„î¿î• î°îžî´î¹î°î± framework, î³î•î®î³î¿î•î¿, î%îµ î%î¹î°î•î¿ î%î-î³îµî,î¿î,
8   î°î±î¹<br>
9   * îµî•î°î¿î»î± îµîîîµî°î„î-îfî¹î%î¿.<br>
10  * An open source framework, small, with small footprint and<br>
11  * easily extensible.<br>
12  *
13  * @package      leaf
14  * @subpackage    core
15  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
16  * @copyright      -
17  * @license        -
18  * @filesource
19  */
20
21 /**
22  * î•î¿î¹î„îµî»îµî- î»îµî¹î„î¿î...î³î-îµî, î´î•î%î¹î¿î...î³î-î±î, îfî...î%î¿î¹îµîî%.
23  *
24  *
25  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
26  * @copyright      -
27  * @license        -
28  * @version        1.0-dev
29  * @since          1.0-dev
30  * @todo
31  * <ol>
32  * <li>î î¹î,î±î%î® î¿î±î•î¿î±î® îµîî¿î, interface.</li>
33  * </ol>
34  */
35 final class leaf_Hash {
36
37     /**
38      * îŸ î¿î•î¿îµîî¿î¹î»îµî¹î%î-î%î¿î, î±î»î³î¿î•î¹î,î%î¿î,.
39      *
40      * @var string
41      */
42     private $defaultAlgorithm = "md5" ;
43
44     /**
45      * î¿î»î¿î¹ î¿î¹ î´î¹î±î,î-îfî¹î%î¿î¹ î±î»î³î¿î•î¹î,î%î¿î¹.
46      *
47      * @var array
48      */
49     private $algorithms = array ("md5" , "sha1" );
50
51     /**
52      *
53      *
54      * @var boolean
55      */
56     private $useAltMethod = FALSE;
57
58
59     /**
60      * îfî•î%î±î„î¿î¹î¿îîµî- î„î•î% î±î•î®îfî• î® î¿î±î¹ î„î•î, î%îµî,î¿î¹î¿î... hash,
61      * î-î%î±î»î„î¹ î„î%î% î°î»î±îfîfî¹î°îžî% îfî...î%î±î•î„î®îfîµîî% md5 î°î±î¹ sha1.
62      *
63      * @return void
64      */
65     public function __construct()
66     {
```

```

67         if (dependsOptOn('hash'))
68             $this-> useAltMethod = TRUE;
69     }
70
71     /**
72      * f~f-i„fufi f„fçf% fufçf¹f,f…f%fi„fiç f±fi»f³fiçfi„fi¹f,f%fiç fçfi„fiçfi, fi¹fiçfiffi..
73      *
74      * @param string $algorithm
75      * @return boolean
76      */
77     public function setAlgorithm($algorithm)
78     {
79         if (in_array($algorithm, $this-> available)) {
80             $this-> algorithm = $algorithm;
81             return TRUE;
82         } else {
83             return FALSE;
84         }
85     }
86
87     /**
88      * fi„fiçfi„fiffi„fi„fi„fiufi¹ (fufi~fi„fufi fži, fi°fufi~fi%fiufi%fiç fufi~fi„fufi fži, fi%fi~fi± fi±fi°fiçfi»fiçfi…fi„fi~fi± bytes)
89      * fi„fi„fi% fiffi„fi%fiçfi~fi„ fi³fi¹fi± fi„fiç fi°fufi~fi%fiufi%fiç fçfiçfi… fi~fi~fi%fiufi„fi±fi¹.
90      *
91      * @param string $data
92      * @param boolean $raw
93      * @return string
94      */
95     public function digestMsg($data, $raw=FALSE)
96     {
97         if ($this-> useAltMethod) {
98             return hash($this-> algorithm, $data, $raw);
99         } else {
100             return call_user_func($this-> algorithm, $data, $raw);
101         }
102     }
103
104     /**
105      * fi„fiçfi„fiffi„fi„fi„fiufi¹ (fufi~fi„fufi fži, fi°fufi~fi%fiufi%fiç fufi~fi„fufi fži, fi%fi~fi±fi, fi±fi°fiçfi»fiçfi…fi„fi~fi± bytes)
106      * fi„fi„fi% fiffi„fi%fiçfi~fi„ fi„fi%fi% fiçfiufi„fi¹fufi±fiçfi%fi~fi%fi% fiufi%fiçfi, fi±fi„fi±fufi~fiçfi…
107      *
108      * @param string $filename
109      * @param boolean $raw
110      * @return string
111      */
112     public function digestFile($filename, $raw=FALSE)
113     {
114         if (file_exists($filename) && is_readable($filename)) {
115             if ($this-> useAltMethod) {
116                 return hash_file($this-> algorithm, $filename, $raw);
117             } else {
118                 return call_user_func($this-> algorithm . "_file" , $filename, $raw);
119             }
120         } else {
121             throw new leaf_Exception('file access failed');
122         }
123     }
124 }
125
126
127 ?>

```

## File Source for Input.php

Documentation for this file is available at [Input.php](#)

```

1 <?php
2 /**
3  * leaf Framework
4  *
5  * <i>PHP version 5</i>
6  *
7  *
8  * The first greek open source PHP5 framework, fast, with small footprint and
9  * easily extensible.<br>
10  * ίεί·ίζί„ίς ίμí»í»í·ί%í·í°í framework PHP5 í±í%ίςí·í°í„ίςí· í°íζí·í·í°í±, í·í·í°í·ίςí·ίς, ί%í·í°í·í° íf
11  * í°í±í· ίμí·í°íςí»í± ίμíέίμí°í„í-ííí·ί%ίς.<br>
12  *
13  *
14  * @package leaf
15  * @subpackage core
16  * @author Avraam Marimpis <makism@venus.cs.teicrete.gr>
17  * @copyright -
18  * @license -
19  * @version 1.0-dev
20  * @filesource
21  */
22
23
24 /**
25  *
26  * @package leaf
27  * @subpackage core
28  * @author Avraam Marimpis <makism@venus.cs.teicrete.gr>
29  * @version 1.0-dev
30  * @since 1.0-dev
31  * @todo
32  * <ol>
33  * <li>ί%í»ίςίέίςí·í·ííí..</li>
34  * </ol>
35  */
36 final class leaf_Input extends leaf_Base implements ArrayAccess {
37
38  /**
39  *
40  *
41  */
42  const LEAF_REG_KEY = "input" ;
43
44  /**
45  * A unique id for each class.<br>
46  * í·ί%í± ί%ίςί%í±í·í·í°í framework í±í%í±í·ί%í&í·í·ííí„í·í°í framework í·í·í± í°í-í,ίμ í°í»í-ííí·.
47  *
48  * @var string
49  */
50  const LEAF_CLASS_ID = "LEAF_INPUT-1_0_dev" ;
51
52
53  /**
54  *
55  *
56  * @var array
57  */
58  private $input = array (
59  "get" => array(),
60  "post" => array()
61 );
62
63 public function __construct()
64 {
65  parent::__construct(self::LEAF_REG_KEY);
66 }

```

```
67
68     public function offsetExists($offset)
69     {
70     }
71
72     public function offsetGet($offset)
73     {
74     }
75
76     public function offsetSet($offset, $value)
77     {
78     }
79
80     public function offsetUnset($offset)
81     {
82     }
83
84 }
85
86 ?>
```

# File Source for Loader.php

Documentation for this file is available at [Loader.php](#)

```
1  i»¿<?php
2  /**
3   * leaf Framework<br>
4   *
5   * PHP version 5<br>
6   *
7   * f^f%î± î±f%î¿î'f°î„î¿î• f°îžî'f'f°î± framework, î'î•î@î'î¿î•î¿, î%îµ î%î'f°î•î¿ î%î-î'îµî,î¿î,
f°î±f'±<br>
8   * îµî•f°î¿î»î± îµî€îµf°î„î-îfî'î%î¿.<br>
9   * An open source framework, small, with small footprint and<br>
10  * easily extensible.<br>
11  *
12  * @package      leaf
13  * @subpackage    core
14  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
15  * @copyright      -
16  * @license        -
17  * @filesource
18  */
19
20
21 /**
22  *
23  * @package      leaf
24  * @subpackage    core
25  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
26  * @version        1.0-dev
27  * @since          1.0-dev
28  * @todo
29  * <ol>
30  * <li>î%î»î¿î¿î¿î-î•îfî•.</li>
31  * </ol>
32  */
33 class leaf_Loader extends leaf_Base {
34
35     const LEAF_REG_KEY = "loader" ;
36
37     const LEAF_CLASS_ID = "LEAF_LOADER-1_0_dev" ;
38
39
40     /**
41     *
42     *
43     * @return void
44     */
45     public function __construct()
46     {
47         parent::__construct(self::LEAF_REG_KEY);
48     }
49
50
51     public function __toString()
52     {
53         return __CLASS__ . " " . self::LEAF_CLASS_ID;
54     }
55
56     /**
57     *
58     *
59     * @param string $modelName
60     * @param array $opts
61     * @return void
62     */
63     public function model($modelName, $opts=NULL)
64     {
65     }
66 }
```



```

67      /**
68      *
69      *
70      * @param string $name
71      * @return void
72      */
73      public function library($name)
74      {
75      }
76
77
78      /**
79      *
80      *
81      * @param string $ext
82      * @param string $namespace
83      * @return void
84      */
85      public function extension($ext, $namespace=NULL)
86      {
87      }
88
89
90      /**
91      *
92      *
93      * @param string $class
94      * @return void
95      */
96      public function endorsed($class)
97      {
98      }
99
100
101      /**
102      *
103      * @param string $plugin
104      * @return void
105      */
106      public function plugin($plugin)
107      {
108      }
109
110
111 }
112
113 ?>

```

# File Source for Logger.php

Documentation for this file is available at [Logger.php](#)

```
1  i»¿<?php
2  /**
3   * leaf Framework
4   *
5   * <i>PHP version 5</i>
6   *
7   *
8   * The first greek open source PHP5 framework, fast, with small footprint and
9   * easily extensible.<br>
10  * ίπί¿ ίεί.ίζί„ί¿ ίμί»ί»ί.ί%ί¹ί°ίε framework PHP5 ί±ί%ί¿ί¹ί°ί„ί¿ί. ί°ίζί·ί¹ί°ί±, ί³ί.ί°ί³ί¿ί.ί¿, ί%ί¹ί°ί.ίε ίf
11  ί%ί-ί³ίμí,ί¿ί,
12  * ί°ί±ί¹ ίμí.ί°ί¿ί»ί± ίμíείμí°ί„ί-ίfί¹ί%ί¿.<br>
13  *
14  * @author      Avraam Marimpis <makism@venus.cs.teicrete.gr>
15  * @copyright    -
16  * @license      -
17  * @package     leaf
18  * @subpackage   core
19  * @version      1.0-dev
20  * @filesource
21  */
22
23
24  /**
25  *
26  *
27  * @author      Avraam Marimpis <makism@venus.cs.teicrete.gr>
28  * @copyright    -
29  * @license      -
30  * @version      1.0-dev
31  * @since        1.0-dev
32  * @method       void log() log(string message) log(string level, string message) log(string
33  level, string message, string fileName) log(string level, string message, string fileName, string
34  className) Handles to log a message. ί`ί%ί±ί»ί±ί%ί²ί-ί%ίμí¹ ί%ί± ί³ί.ί-ί¹ίμí¹ ί-ί%ί± ί%ί°ί%ί„ί%ί±.
35  * @todo
36  * <ol>
37  * <li>ί ί¹ί.ί±ί%ί° ί„ίεί¿ίfί„ί°ί.ί¹ί%ί. ί³ί¹ί± ί°ί±ί„ί±ί±ίζί·ί-ίfί. ί„ί%ί% ίfί±ί±ί»ί%ί-ί„ί%ί% ί„ί¿ί„ level
38  "Error"
39  * ίfίμ ί²ί-ίfί. ί¹ίμí¹ί¿ί%ί-ί%ί% ί° ί°ί-ίεί¿ί¹ί± e-mail ί¹ί¹μí·ί,ί„ί%ίfί.·.</li>
40  * </ol>
41  */
42  final class leaf_Logger extends leaf_Base implements leaf_Log {
43
44      const LEAF_REG_KEY = "logger" ;
45
46      const LEAF_CLASS_ID = "LEAF_LOGGER-1_0_dev" ;
47
48      /**
49       *
50       * @var string
51       */
52      private $filename= NULL;
53
54      /**
55       *
56       * @var string
57       */
58      private $stampPat= NULL;
59
60      /**
61       *
62       *
63       * @var array
```



```

141         $tmp = $this-> buffer;
142         unset($this-> buffer);
143
144         if ($return)
145             return $tmp;
146
147     }
148
149     return NULL;
150 }
151
152 /**
153  * end_flush
154  *
155  * @return void
156  */
157 public function end_flush()
158 {
159     $this-> flush();
160     fwrite($this-> fp, "\n" );
161     flock($this-> fp, LOCK_UN);
162     fclose($this-> fp);
163 }
164
165 /**
166  * setTimeStampPattern
167  *
168  * @param string $pat
169  * @return void
170  */
171 public function setTimeStampPattern($pat)
172 {
173     $this-> stampPat = $pat;
174 }
175
176 /**
177  * setLevel
178  *
179  * Set the level of errors to capture.<br>
180  * f~f-i„fµf¹ f„f¿ fµfēf-iefµf¹f¿ f„f¿f% ifi+i+i»f%fi-i„f¿f% fēf¿f... f,f± f°fi+i„fi+i+f¿f•f@ifµf¹¹.
181  *
182  *
183  * @param string $filterLevel
184  * @return boolean
185  */
186 public function setLevel($filterLevel)
187 {
188     if (array_key_exists($filterLevel, $this-> levels)) {
189         $this-> threshold = $filterLevel;
190         return true;
191     } else {
192         return false;
193     }
194 }
195
196 /**
197  * __call
198  *
199  *
200  * @param string $method
201  * @param array $args
202  * @return void
203  */
204 public function __call($method, $args)
205 {
206     if ($method=="log" ) {
207         $s = sizeof($args);
208
209         if ($s> 0) {
210             if ($s==1) {
211                 $msg = $args[0];
212                 fwrite($this-> fp, $msg . "\n" );
213             } else {
214
215                 if (in_array($args[0], $this-> levels)==FALSE)
216                     ;
217
218                 $level = strtoupper($args[0]);
219
220                 if ($this-> threshold==$level || $this-> threshold=="All" ) {

```

```

221         $msg    = $args[1];
222         $file    = ($s==3)
223             ? "(file: " . $args[2] . ")"
224             : NULL;
225
226         fwrite($this-> fp, "    [{ $level}] { $msg} { $file}" .
227 "\n"    );
228     }
229 }
230 }
231 }
232
233 }
234
235 ?>

```

# File Source for Model.php

Documentation for this file is available at [Model.php](#)

```
1  i»¿<?php
2  /**
3   * leaf Framework<br>
4   *
5   * PHP version 5<br>
6   *
7   * f^f%î± î±f%î¿î¹f°î„î¿î• f°îžî¹f¹f°î± framework, î¹î•î®î¹î¿î•î¿, î¼îµ î¼î¹î°î¹î¿ î¼î¹î¹îµî¹,î¿î,
8   f°î±f¹<br>
9   * îµî¹•f°î¿î»î± îµî¹îµf°î„î¹îfî¹f%î¿.<br>
10  * An open source framework, small, with small footprint and<br>
11  * easily extensible.<br>
12  *
13  * @package      leaf
14  * @subpackage    core
15  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
16  * @copyright      -
17  * @license        -
18  * @filesource
19  */
20
21 /**
22  *
23  * @package      leaf
24  * @subpackage    core
25  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
26  * @version        1.0-dev
27  * @since          1.0-dev
28  * @todo
29  * <ol>
30  * <li>î¼î¹»î¿î¿î¿î¹î¹îfî¹.</li>
31  * </ol>
32  */
33 abstract class leaf_Model extends leaf_Base {
34
35     const LEAF_REG_KEY = "model" ;
36
37     const LEAF_CLASS_ID = "LEAF_MODEL-1_0_dev" ;
38
39
40     /**
41     *
42     * @return void
43     */
44     public function __construct()
45     {
46
47     }
48 }
49
50 }
51
52 ?>
```

# File Source for Request.php

Documentation for this file is available at [Request.php](#)

```

1 >?<?php
2 /**
3  * leaf Framework
4  *
5  * <i>PHP version 5</i>
6  *
7  *
8  * The first greek open source PHP5 framework, fast, with small footprint and
9  * easily extensible.<br>
10 * ίοίε ίεί.ίζί„ίε ίμλ»ί»ί.ί%ί'ί°ίε framework PHP5 ί±ί%ίεί'ί°ί„ίεί• ί°ίζί'ί'ί°ί±, ί'ί.ί@ί³ίεί•ίε, ί%ί'ί°ί.ίε ίf
11 * ί°ί±ί' ίμλ.ί°ίεί»ί± ίμίέίμί°ί„ί-ίfί'ί%ίε.<br>
12 *
13 *
14 * @package         leaf
15 * @subpackage       core
16 * @author           Avraam Marimpis <makism@venus.cs.teicrete.gr>
17 * @copyright        -
18 * @license          -
19 * @version          1.0-dev
20 * @filesource
21 */
22
23
24 /**
25 * ί ί±ί.ί-ί+ίμλ¹ έεί.ίείfί²ί±ίfί. ίfίμ ίεί»ί± ί„ί± ί'ί'ί±ί°ί.ί'ί„ί~ ίfί„ίείί'ί±ίμλ-ί± ί„ίεί... Uri.
26 *
27 * ί"ί·ί>ί±ί'ί@ ί-ί+ίεί...ί%ίμ ί„ί·ί% ί'ί...ί%ί±ί„ίεί„ί·ί„ί± ί%ί± ί±ί%ί±ί'ίμλ·ί,ίεί·ί%ίμ ί°ί±ί¹ ί%ί± ί%ί·ί„ί@ίfίεί...
28 * ί„ίε ί±ί.ί+ίμλ-ίε ί±ίείε ί„ίε ίείέίί-ίε ί,ί± ί'ίεί·ί„ίζίfίεί...ί%ίμ ί„ίεί% Controller, ί„ίε Action,
29 * ί„ί± ίμίέí·έεί»ί-ίεί% ίfί,ίεί·ί+ίμλ-ί± (segments), ί°ί±ί¹ ί„ί-ί»ίεί, ί„ίε query string ίfί„ί.
30 * ί°ί»ί±ίfίfί'ί°ί@ ί„ίεί... ί%ίεί·ί'ί@ ίείέί&ί, ί. super-global ί%ίμλ„ί±ί²ί»ί·ί„ί@ $_GET.<br>
31 * ίοί-ίείεί, ί„ί»ί»ίείείί'ίμλ- ί'ίμλ%ί'ί°ί-ί, ί%ίμλ,ίεί'ίεί...ί, ίfί+ίμλ„ί·ί°ί-ί, ί%ίμ ί„ί·ί% ί±ί%ί±ί°ί±ί„ίμλ·ί.
32 * ίfίμ ί%ί-ί± ί'ί'ίμλ·ί...ί%ίfί. ί°ί±ί¹ ί±ί%ί±ί'ί·ί·ί%ί'ίεί...ί·ί³ί-ί± ίμλ%ίεί, Uri.
33 *
34 *
35 * @author           Avraam Marimpis <makism@venus.cs.teicrete.gr>
36 * @copyright        -
37 * @license          -
38 * @version          1.0-dev
39 * @since            1.0-dev
40 * @todo
41 * <ol>
42 * <li>ί%ί»ίείέίί-ί·ίfί. έεί'ί,ί±ί%ίζί% facade ίfί...ί%ί±ί·ί„ί@ίfίμλ&ί%.</li>
43 * </ol>
44 */
45 final class leaf_Request extends leaf_Base {
46     const LEAF_REG_KEY = "request" ;
47
48     const LEAF_CLASS_ID = "LEAF_REQUEST-1_0_dev" ;
49
50
51
52 /**
53 * ίοί± ίμίέí·έεί»ί-ίεί% ί°ίεί%ίλ-ί„ί·ί± ί„ίεί... Uri έείéί... ί'ίμλ% ί±ί%ί@ί°ίéί...ί% ίεί·ί„ίμ
54 * ίfί„ίεί% Controller ίεί·ί„ίμ ίfί„ίε Action.
55 *
56 * έεί±ί·ί-ί'ίμλ¹ί³ί%ί±:
57 * <pre>http://localhost/Blog/view/200X-XX-XX/authorName/postTitle</pre>
58 * ίεί„ί'ί'ί°ίέέίéί„ίμ ί%ίμλ„ί~ ί„ίε "view", ί,ί± ί,ίμλ&ί·ί·ί,ίμλ- ίμίέí·έεί»ί-ίεί%
59 * ίfί„ίείί'ί+ίμλ-ίε.
60 *
61 *
62 * @var array
63 */
64 private $segments = NULL;
65
66 /**

```

[illegible]





```

226 *
227 *
228 * @param integer $n
229 * @return string|NULL
230 * @todo
231 * <ol>
232 * <li>f¶f»f¿f¶f¿f~f·f¶f·.</li>
233 * </ol>
234 */
235 public function segment($n)
236 {
237
238 }
239
240 /**
241 *
242 *
243 *
244 * @return array
245 * @todo
246 * <ol>
247 * <li>f¶f»f¿f¶f¿f~f·f¶f·.</li>
248 * </ol>
249 */
250 public function segmentsAsArray()
251 {
252
253 }
254
255 /**
256 *
257 *
258 * @param string $offset
259 * @return mixed
260 * @todo
261 * <ol>
262 * <li>f f¹f,f±f%fi@ f±f»f±f³fi@ f¿f%fi¶fi±fi„fi¿fi„.</li>
263 * </ol>
264 */
265 public function getQueryString($offset)
266 {
267     if ($this-> queryElems!=NULL)
268         if (array_key_exists($offset, $this-> queryElems))
269             return $this-> queryElems[$offset];
270         else
271             return NULL;
272 }
273
274 /**
275 *
276 *
277 * @return string
278 * @todo
279 * <ol>
280 * <li>f f¹f,f±f%fi@ f±f»f±f³fi@ f¿f%fi¶fi±fi„fi¿fi„.</li>
281 * <li>f·f¶f±f%fi~f»f¶f³fi¿fi, f¶f·fi%fi,f¶f¶f·fi, fi„fi·fi, f¶fi„fi%fi²fi¿fi»fi¿fif¶fi¹fi·fi~fi„.</li>
282 * </ol>
283 */
284 public function getQueryStringAsString()
285 {
286     if ($this-> queryElems!=NULL) {
287         $str = NULL;
288
289         foreach($this-> queryElems as $Var => $Val)
290             $str .= $Var . " = " . $Val . " , " ;
291
292         if ($str{strlen($str)-1}==" , " )
293             $str{strlen($str)-1} = " " ;
294
295         return $str;
296     } else
297         return NULL;
298 }
299
300 }
301
302 ?>

```

# File Source for Response.php

*Documentation for this file is available at [Response.php](#)*

```

1 i»¿<?php
2 /**
3  * leaf Framework
4  *
5  * <i>PHP version 5</i>
6  *
7  *
8  * The first greek open source PHP5 framework, fast, with small
9  * footprint and easily extensible.<br>
10 * ïðíç îēī·îžĭ„îç ĩµŃ»ï»ï·ĩ¼ï·ï°İĞ framework PHP5 İ±İ½İçİ·İ°İ„İçİ· İ°İŽİ·İ¹İ°İ±, İ³İ·İ@İ³İçİ·İç,
11 * İ¾İ¹İ°İ·İĞ İřİµ İ¼İ-İ³İµİ,İçİ, İ°İ±İ¹ İµİ·İ°İçİ»İ± İµİēİµİ°İ„İ~İřİ¹İ¾İç.<br>
12 *
13 *
14 * @package         leaf
15 * @subpackage      core
16 * @author           Avraam Marimpis <makism@venus.cs.teicrete.gr>
17 * @copyright        -
18 * @license          -
19 * @version          1.0-dev
20 * @filesource
21 */
22
23
24 /**
25  *
26  *
27 * @author           Avraam Marimpis <makism@venus.cs.teicrete.gr>
28 * @copyright        -
29 * @license          -
30 * @version          1.0-dev
31 * @since            1.0-dev
32 * @todo
33 *     <ol>
34 *       <li>Îşİ»İçİēİçİ·İ·İřİ·.</li>
35 *     </ol>
36 */
37 final class leaf_Response extends leaf_Base {
38
39     /**
40      *
41      *
42     */
43     const LEAF_REG_KEY = "response" ;
44
45     /**
46      * A unique id for each class.<br>
47      * Îşİ± İ¾İçİŃ»İ·İ¹İ°İĞ İ±İ½İ±İ¾İ&İ·İ¹İřİ„İ¹İ°İĞ İ³İ¹İ± İ°İ¬İ,İµ İ°İ»İ¬İřİ..
48      *
49      * @var string
50      */
51     const LEAF_CLASS_ID = "LEAF_RESPONSE-1_0_dev" ;
52
53
54     /**
55      *
56      *
57      * @var boolean
58      */
59     private $useTidy = FALSE;
60
61     /**
62      *
63      *
64      * @var boolean
65      */
66     private $useGzip = FALSE;
67 }

```

```

68  /**
69  *
70  *
71  * @var string
72  */
73  private $buffer = NULL;
74
75
76  /**
77  * Class constructor
78  *
79  *
80  * @return void
81  */
82  public function __construct()
83  {
84      parent::__construct(self::LEAF_REG_KEY);
85
86      $outputHandler = $this-> config['output_handler'];
87
88      /**
89      *
90      */
91      if (!empty($outputHandler)) {
92
93          /**
94          *
95          *
96          */
97          if (in_array($outputHandler, ob_list_handlers())) {
98
99              switch ($outputHandler) {
100
101                  /**
102                  *
103                  *
104                  */
105                  case "gz" :
106                      $zlibCompressionStatus = ini_get('zlib.output_compression');
107
108                      if ($zlibCompressionStatus==FALSE)
109                          $this-> useGzip = TRUE;
110                      break;
111
112                  /**
113                  *
114                  *
115                  */
116                  case "tidy" :
117                      if (extension_loaded('tidy') && function_exists('ob_tidyhandler'))
118                          $this-> useTidy = TRUE;
119                      break;
120
121                  default:
122                      break;
123              }
124
125          }
126      }
127  }
128
129
130  public function __toString()
131  {
132      return __CLASS__ . " " . self::LEAF_CLASS_ID;
133  }
134
135  /**
136  *
137  *
138  * @param string $name
139  * @param string $value
140  * @return void
141  */
142  public function setHeader($name, $value)
143  {
144  }
145
146  public function clearHeaders()
147  {

```

```

148     }
149 }
150
151 /**
152  *
153  *
154  * @return void
155  */
156 public function sendResponse()
157 {
158 }
159
160 /**
161  * Start output buffering
162  *
163  * @return void
164  */
165 public function ouputBufferingStart()
166 {
167     if ($this-> useGzip)
168         ob_start('ob_gzhandler');
169     else if ($this-> useTidy)
170         ob_start('ob_tidyhandler');
171 }
172
173 /**
174  * End output buffering
175  *
176  * @param boolean $returnBuffer
177  * @return void|string
178  */
179 public function outputBufferingEnd($returnBuffer=FALSE)
180 {
181 }
182
183 /**
184  * Gets output buffer.
185  *
186  * @return string
187  */
188 public function getOutputBuffer($flush=FALSE)
189 {
190     if ($flush)
191         ob_get_flush();
192     else
193         return ob_get_contents();
194 }
195
196 /**
197  *
198  * @return void
199  */
200 public function flushOutputBuffer()
201 {
202     ob_end_flush();
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }

```

# File Source for Router.php

Documentation for this file is available at [Router.php](#)

```
1  i»¿<?php
2  /**
3   * leaf Framework
4   *
5   * <i>PHP version 5</i>
6   *
7   *
8   * The first greek open source PHP5 framework, fast, with small footprint and
9   * easily extensible.<br>
10  * ίρλζ ίεί.ίζί„ίζ ίρλ»ί»ί.ί%ίί°ίε framework PHP5 ί±ί%ίζίί°ί„ίζί. ί°ίζί.ίί°ί±, ί±ί.ί°ίζί.ίζ, ί%ίί°ί.ίε ίf
11  ί%ί-ί±ίρλ.ίζί,
12  * ί°ί±ίί ίρλ.ί°ίζίί»ί± ίρλείρλ°ί„ί-ίfίί%ίζ.<br>
13  * @package leaf
14  * @subpackage core
15  * @author Avraam Marimpis <makism@venus.cs.teicrete.gr>
16  * @copyright -
17  * @license -
18  * @version 1.0-dev
19  * @filesource
20  */
21
22
23  /**
24  * ί.ίείρλ%ίρλ.ί±ί-ί%ίρλ„ί±ί ί°ί±ίί ίfίί»ί„ί.ί-ί.ίρλί, ί„ίζ Uri ί±ίί± ί„ίζ ί„ίί ίεί.ί-ίείρλί ί%ί± ί±ί-ί%ίρλί
25  *
26  * ί.ί%ί±ί°ί±ί»ί.ίεί„ίρλί ίείζίίζί% Controller ί.ί± ίεί.ί-ίείρλί ί%ί± ίρλ°ί„ίρλί»ί-ίfίζί„ί%ίρλ ί°ί±ίί
27  * ίείζίί± ί%ί-ί.ίζίίζ (ί±ί% ί-ίίρλί ίζί.ίίfί„ίρλί). ί.ίείίεί»ί-ίζί% ίfίί»ί„ί.ί-ί.ίρλί ί„ίζ Uri
28  * ί±ίί± ί±ί±ί.ί±ί°ί„ί°ί.ίρλί, ίείζί„ί.ί± ίεί.ί-ίείρλί ί%ί± ί°ί.ί-ίfί°ίζί%ί„ί±ί ίρλ°ίρλί- ί°ί±ίί ί±ίείζίfίεί-
29  * ί„ίζ Query String ίεί„ί±ί% ί±ί.ίρλί-ί%ίρλ„ί±ί ί°ί±ίί ίρλ-ί% ί-ί±ίρλί ίρλ%ίρλ.ί±ίείζίί.ί.ί,ίρλί.
30  *
31  *
32  * @package leaf
33  * @subpackage core
34  * @author Avraam Marimpis <makism@venus.cs.teicrete.gr>
35  * @version 1.0-dev
36  * @since 1.0-dev
37  * @todo
38  * <ol>
39  * <li>ίœί°ίέί„ί, {<link leaf_Router ί±ί„ί„ί° ί. ί°ί»ί-ίfίί%ί± ί±ί%ί±ί»ί-ί-ίρλί ί°ί±ίί ί„ίί,
40  * ίρλίί„ίζί„ί.ί±ί-ίρλί, ί„ί.ί, ί°ί»ί-ίfί.ί,
41  * {<link leaf_Request>;</li>
42  * <li>ί ί±ί.ίζίί° ί%ί-ί±ί, ίρλ%ί±ί»ί»ί±ί°ί„ί.ί°ί°ί, ί°ί»ί-ίfί.ί, (ίείζί„ί.ίείί.ί±ί%ίœί%, ί.ί±
43  * ί°ί»ί.ί.ίζίί%ίζί%ίρλί- ί±ίείε ί±ί„ί.ί°ί%)
44  * ίείζί„ί.ί.ί± ί±ί.ί-ίfίί%ίζίείζίίίρλί- ίfί„ί±ί„ί.ί°ίζί.ί, ί°ί±ί%ίœί%ίρλί, ί.ί.ίζίί%ίζίί»ίœί±ί.ίfί.ί.</li>
45  * </ol>
46  */
47  final class leaf_Router extends leaf_Base {
48
49      const LEAF_REG_KEY = "router" ;
50
51      const LEAF_CLASS_ID = "LEAF_ROUTER-1_0_dev" ;
52
53      /**
54       *
55       * @var string
56       */
57      private $requestUri = NULL;
58
59      /**
60       *
61       *
62       * @var string
63       */
64      private $requestClass = NULL;
```

```

65
66 /**
67  *
68  *
69  * @var string
70  */
71 private $requestMethod = NULL;
72
73 /**
74  *
75  *
76  * @var array
77  */
78 private $segments = NULL;
79
80 /**
81  *
82  *
83  * @var string
84  */
85 private $queryString = NULL;
86
87 /**
88  *
89  *
90  * @var array
91  */
92 private $queryStringElements = NULL;
93
94
95 /**
96  * 
97  * @see      leaf_Config
98  * @see      leaf_Request
99  * @return    void
100  * @todo
101  * <ol>
102  * <li>"
103  *  constructor if 
104  * <li>"
105  *  Uri.</li>
106  * </ol>
107  */
108 public function __construct() {
109
110     parent::__construct(self::LEAF_REG_KEY);
111
112     $this-> requestUri    = $_SERVER['REQUEST_URI'];
113
114     /*
115     * 
116     */
117     $this-> requestUri = preg_replace("@/+@i" , "/" , $this-> requestUri);
118
119     /*
120     * We check if there are any illegal characters in
121     * the Uri that they shouldn't.
122     * Moreover, we take into consideration if the option
123     * $this->config['url_suffix'] is used which defines the file
124     * extensions that will be shown, so we ignore it.
125     *
126     */
127     /*
128     * $skipExt = (!empty($this->config['url_suffix']))
129     *           ? "(\.[^?=&]+)?"
130     *           : "";*/
131     $skipExt = "" ;
132
133     $skipQueryString = ($this-> config['allow_query_strings']=='Yes' )
134     ? "(\?[ "
135     . $this-> config['allow_query_string_chars']
136     . "]+(= "
137     . $this-> config['allow_query_string_chars']
138     . "*)?(&?)?"
139     : "" ;
140
141     $checkUri = preg_match_all(
142         "| "

```

```

143         . "^[" . preg_quote($this-> config['allow_uri_chars']) . "]" +
144         . " {"$skipExt}{$skipQueryString}$"
145         "|iu"
146         $this-> requestUri,
147         $hits
148     );
149
150     if ($checkUri==0) {
151         /*$this->logger->log(
152             "Error",
153             "Routing Error, Malicious characters found in Uri ("
154             . "{$this->requestUri})"
155         );*/
156
157         showHtmlMessage(
158             "Routing Error",
159             "Malicious characters found in Uri!"
160             true
161         );
162     }
163
164     /*
165     * We check if there is a query string in our uri.
166     * If query strings are enable in file
167     * (Etc/config.php - parameter $config['allow_query_strings']),
168     * we process it in an associative array way.
169     * Either way, we remove the query string from the
170     * request uri.
171     *
172     */
173
174     if (preg_match("@\?(.+?(=.)?)\&?$)+@" . "iu" , $this-> requestUri, $matches)) {
175
176         if ($this-> config['allow_query_strings']=="Yes" ) {
177
178             $this-> queryString = $matches[0];
179
180             $keysWithValues = explode("&" , $matches[1]);
181
182             foreach ($keysWithValues as $Elem) {
183
184                 if (strpos($Elem, "=") ) {
185                     list($Key, $Value) = explode("=", $Elem);
186                     $this-> queryStringElements[$Key] = $Value;
187                 } else {
188                     $this-> queryStringElements[$Elem] = NULL;
189                 }
190             }
191
192             $this-> requestUri = preg_replace(
193                 "| " . preg_quote($this-> queryString) .
194                 "|iu" ,
195                 " " .
196                 $this-> requestUri
197             );
198
199         } else {
200             $this-> requestUri = preg_replace(
201                 "| " . preg_quote($matches[0]) . "|i" ,
202                 " " .
203                 $this-> requestUri
204             );
205         }
206     }
207
208
209     /*
210     * Removal of leading base dir.
211     *
212     */
213
214     $this-> requestUri = preg_replace(
215         " " . @^{ $this-> config['base_dir'] }@" ,
216         " " .
217         $this-> requestUri
218     );
219
220
221     /*
222     * Removal of the url trailing if defined.

```



```

223 *
224 */
225 if (!empty($this-> config['url_suffix']))
226     $this-> requestUri = preg_replace(
227         "@\.{${this-> config['url_suffix']}}/?$@" ,
228         "" ,
229         $this-> requestUri
230     );
231
232 /*
233  * Removal of trailing '/' (if exists in Uri).
234  */
235 if (preg_match("@/$@" , $this-> requestUri))
236     $this-> requestUri = preg_replace("@/$@" , "" , $this-
> requestUri);
237
238 /*
239  * If no class is defined instantiate the default Controller.
240  */
241 if ($this-> requestUri=="/" ||
242     $this-> requestUri==NULL ||
243     ($this-> requestUri{0}=="?" && $this-
> config['allow_query_strings']=="Yes" ))
244     $this-> requestClass = $this-> config['default_controller'];
245
246 /*
247  * We extract the Controller class (if defined).
248  */
249 else
250     $this-> chopSegment($this-> requestClass);
251
252 /*
253  * We extract the method (if defined).
254  */
255 $this-> chopSegment($this-> requestMethod);
256 if ($this-> requestMethod==NULL)
257     $this-> requestMethod = "index" ;
258
259 /*
260  * If there are more segments in the Uri, except from the
261  * class name and the method name, we store those segments
262  * in an array and we provide them at the user's disposal.
263  */
264 $segments = explode("/", $this-> requestUri);
265 foreach ($segments as $s) {
266     if ($s!=null)
267         $this-> segments[] = $s;
268 }
269
270 public function __toString()
271 {
272     return __CLASS__ . " " . self::LEAF_CLASS_ID;
273 }
274
275 /**
276  * f·ief·ifî„î·î·îpî¹ î-fîîfî% îef-fîî+f°î± î%îp î„î± ifî„îçî¹î+îpî-fî± î„îçî... Query String.
277  * @see leaf_Request
278  * @return array=NULL
279  */
280 public function queryStringElements()
281 {
282     return $this-> queryStringElements;
283 }
284
285 /**

```



# File Source for View.php

Documentation for this file is available at [View.php](#)

[illegible]

```
67     }
68
69     }
70
71     ?>
```

# File Source for Benchmark.php

Documentation for this file is available at [Benchmark.php](#)

```
1  <?php
2  /**
3   * leaf Framework
4   *
5   * <i>PHP version 5</i>
6   *
7   *
8   * The first greek open source PHP5 framework, fast, with small footprint and
9   * easily extensible.<br>
10  * ίνίζ ίεί.ίζίι,ίζ ίμί»ί»ί.ί%ί¹ί°ίε framework PHP5 ί±ί%ίζίί°ίι,ίζί. ί°ίζίί.ί¹ί°ί±, ί³ί.ί@ί³ίζίί.ίζ, ί%ί¹ί°ί.ίε ίf
11  ί%ί-ί³ίμí,ίζί,
12  * ί°ί±ί¹ ίμí.ί°ίζίί»ί± ίμíείμí°ίι,ί-ίfί¹ί%ίζ.<br>
13  *
14  * @package    leaf
15  * @subpackage core.helpers
16  * @author      Avraam Marimpis <makism@venus.cs.teicrete.gr>
17  * @copyright   -
18  * @license     -
19  * @version     1.0-dev
20  * @filesource
21  */
22
23  ?>
```

# File Source for Request.php

Documentation for this file is available at [Request.php](#)

```
1  <?php
2  /**
3   * leaf Framework
4   *
5   * <i>PHP version 5</i>
6   *
7   *
8   * The first greek open source PHP5 framework, fast, with small footprint and
9   * easily extensible.<br>
10  * ίνίζ ίεί•ίζί„ίζ ίμί»ί»ί•ί%ί°ίε framework PHP5 ί±ί%ίζίί°ί„ίζί• ί°ίζίί°ίί±, ί°ί•ί@ί°ίζί•ίζ, ί%ί°ί•ί•ίε ίf
11  * ί°ί±ί°ί ίμί•ί°ίζί»ί± ίμίείμί°ί„ί-ίfί°ί%ίζ.<br>
12  *
13  *
14  * @package    leaf
15  * @subpackage core.helpers
16  * @author      Avraam Marimpis <makism@venus.cs.teicrete.gr>
17  * @copyright   -
18  * @license     -
19  * @version     1.0-dev
20  * @filesource
21  */
22
23  ?>
```

# File Source for Hook.php

Documentation for this file is available at [Hook.php](http://hook.php)

```
1  i»¿<?php
2  /**
3   * leaf Framework<br>
4   *
5   * PHP version 5<br>
6   *
7   * f^f%î± î±f%î¿î¹f°î„î¿î• f°îžî¹f¹f°î± framework, î¹î•î®î¹î¿î•î¿, î¼îµ î¼î¹î°î•î¿ î¼î¹-î¹îµî¹,î¿î,
8   f°î±f¹<br>
9   * îµî¹•î°î¿î»î± îµî¹îµf°î„î¹-îfî¹î¼î¿.<br>
10  * An open source framework, small, with small footprint and<br>
11  * easily extensible.<br>
12  *
13  * @package      leaf
14  * @subpackage    core.hook
15  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
16  * @copyright      -
17  * @license        -
18  * @filesource
19  */
20
21 /**
22  *
23  * @package      leaf
24  * @subpackage    core.hook
25  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
26  * @version        1.0-dev
27  * @since          1.0-dev
28  * @todo
29  * <ol>
30  * <li>î¼î»î¿î¿î¿î¹-î¹fî¹.</li>
31  * </ol>
32  */
33 abstract class leaf_Hook extends leaf_Loader {
34
35     const LEAF_REG_KEY = "hook" ;
36
37     const LEAF_CLASS_ID = "LEAF_HOOK-1_0_dev" ;
38
39
40     /**
41     *
42     *
43     * @return void
44     */
45     public function __construct()
46     {
47         parent::__construct();
48     }
49
50     /**
51     *
52     *
53     * @return string
54     */
55     abstract public function __toString();
56
57 }
58
59 ?>
```

# File Source for Hook\_Conditional.php

Documentation for this file is available at [Hook\\_Conditional.php](http://www.phpdoc.org/package/PhpDocumentor/Projects/PhpDocu)

```
1  <?php
2  /**
3   * leaf Framework<br>
4   *
5   * PHP version 5<br>
6   *
7   * leaf framework, a small, with small footprint and easily
8   * extensible.<br>
9   * An open source framework, small, with small footprint and<br>
10  * easily extensible.<br>
11  *
12  * @package      leaf
13  * @subpackage    core.hook
14  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
15  * @copyright      -
16  * @license        -
17  * @filesource
18  */
19
20
21 /**
22  *
23  * @package      leaf
24  * @subpackage    core.hook
25  * @author        Avraam Marimpis <makism@venus.cs.teicrete.gr>
26  * @version        1.0-dev
27  * @since          1.0-dev
28  * @todo
29  * <ol>
30  * <li>leaf framework, a small, with small footprint and easily
31  * </li>
32  * </ol>
33  */
34 abstract class leaf_Hook_Conditional extends leaf_Hook {
35     const LEAF_REG_KEY = "conditional_hook" ;
36
37     const LEAF_CLASS_ID = "LEAF_HOOK_CONDITIONAL-1_0_dev" ;
38
39
40     /**
41     *
42     * @return void
43     */
44     public function __construct()
45     {
46         parent::__construct();
47     }
48
49     /**
50     *
51     * @return boolean
52     */
53     abstract public function condition();
54 }
55
56 ?>
```



# File Source for Hooks.php

*Documentation for this file is available at [Hooks.php](#)*

```

1 <?php
2 /**
3  * leaf Framework
4  *
5  * <i>PHP version 5</i>
6  *
7  *
8  * The first greek open source PHP5 framework, fast, with small footprint and
9  * easily extensible.<br>
10 * ίαίι ίέί·ίΪί,ίι ίμί»ί»ί·ί%ί·ί°ί framework PHP5 ί±ί%ίιί·ί°ί, ί°ίΪί·ί·ί°ί±, ί·ί·ί@ί·ίι·ίι, ί%ί·ί°ί·ί° ίf
11 * ί°ί±ί· ίμί·ί°ίιί»ί± ίμίέίμί°ί,ί-ίfί·ί%ίιι.<br>
12 *
13 *
14 * @package leaf
15 * @subpackage core.hook.helpers
16 * @author Avraam Marimpis <makism@venus.cs.teicrete.gr>
17 * @copyright -
18 * @license -
19 * @version 1.0-dev
20 * @filesource
21 */
22
23
24 /**
25  *
26  *
27  */
28 define('HOOK_PRE_CONTROLLER_DISPATCH', 'pre_controller_dispatch');
29
30 /**
31  *
32  *
33  */
34 define('HOOK_POST_CONTROLLER_DISPATCH', 'post_controller_dispatch');
35
36 /**
37  *
38  *
39  */
40 define('HOOK_POST_FRONT_CONTROLLER', 'post_front_controller');
41
42
43 /**
44  *
45  *
46  *
47  * @access private
48  * @param string $level
49  * @return array
50  */
51 function introspectHooks($level=NULL)
52 {
53     if ($level!=NULL)
54         if (!defined('HOOK_' . strtoupper($level)))
55             throw new leaf_Exception("Unknown Hook Level!" );
56
57     static $allHooks;
58
59     if ($allHooks==NULL)
60         $allHooks = leaf_Registry::getInstance()-> config-> hooks;
61
62     if ($level!=NULL)
63         return $allHooks[$level];
64     else
65         return $allHooks;
66 }

```

```

67
68 /**
69  * Runs all registered hooks in one level.<br>
70  * f·f°i„îµî»îµî- î@î»îçî...î, î„îçî...î, hooks îfîµ î-fî%î± îµî€î-f€îµî`îç.
71  *
72  *
73  * @access private
74  * @param string $level
75  * @return void
76  */
77 function runHooks($level)
78 {
79     $hooks = introspectHooks($level);
80 }
81
82 /**
83  * Executes a specific hook.<br>
84  * f·f°i„îµî»îµî- î-fî%î±î% îfî...î:f°îµî°î·î·î%î-fî%îç hook.
85  *
86  * @access private
87  * @param string $controller
88  * @param string $method
89  * @return void
90  */
91 function runHook($controller, $method)
92 {
93 }
94
95
96 /**
97  * Returns all registered hooks for one -or all- level. Also, the hooks
98  * can be filtered using the parameters $controller and $method.<br>
99  * f·î€î:îfî„î·î-îîµî¹ î@î»îçî...î, î„îçî...î, î°î±î„î±î±î%î·î·î%î-fî%îçî...î, hooks î³î¹î± î-fî%î± î@ î@î»î»î± î„î± îµî
100  * î·î€î:î€î»î-fîçî% î„î± hooks î%î€îçî·îçî·î%î± î·î¹î±î%î·î¹îfî„îçî·î%, î±î·î·îfî¹î%îçîî€îî¹î%î„î±î, î„î¹î,
101  * î€î±î·î±î%î-fî„î·îçî...î,
102  * $controller î°î±î¹ $method.
103  *
104  * @param string/NULL $level
105  * @param string/NULL $controller
106  * @param string/NULL $method
107  * @return array
108  */
109 function getHooks($level=NULL, $controller=NULL, $method=NULL)
110 {
111 }
112 }
113
114 ?>

```

# File Source for Debug.php

Documentation for this file is available at [Debug.php](#)

```
1  <?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright Copyright (c) 2007 Avraam Marimpis
9   * @license http://leaf-framework.sourceforge.net/LICENSE/ New BSD License
10  * @link http://leaf-framework.sourceforge.net
11  *
12  * @package leaf
13  * @subpackage front
14  * @author Avraam Marimpis <makism@users.sf.net>
15  * @version $Id: Debug.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  * @todo
18  * <ol>
19  * <li>Document.</li>
20  * <li>Recheck the information displayed under the tab
21  * "Config Settings".</li>
22  * </ol>
23  */
24
25
26 /*
27  * The statistics will be produced only if leaf's version
28  * is tagged as "dev".
29  * This is most surely that will change.
30  */
31 if (LEAF_REL_STATUS=='DEV') {
32     dependsOn('xdebug');
33     dependsOnFunc('memory_get_usage');
34
35     $parseTime = xdebug_time_index();
36     $memoryUsage = memory_get_usage();
37     $memoryUsageKbs = sprintf("%1.2fkbs", $memoryUsage/1024);
38     $baseUrl = $reg->config['base_url'];
39
40     echo "<br /><br /><br /><br /><br /><br />";
41
42     echo <<<DEBUG_STYLES
43     < link rel="stylesheet" type="text/css"
href="{ $baseUrl}yui/tabview.css">
44     < link rel="stylesheet" type="text/css"
href="{ $baseUrl}yui/border_tabs.css">
45     < script type="text/javascript"
src="{ $baseUrl}yui/yahoo.js"></script>
46     < script type="text/javascript"
src="{ $baseUrl}yui/event.js"></script>
47     < script type="text/javascript" src="{ $baseUrl}yui/dom.js"></script>
48     < script type="text/javascript" src="{ $baseUrl}yui/element-
beta.js"></script>
49     < script type="text/javascript"
src="{ $baseUrl}yui/tabview.js"></script>
50     < style type="text/css">
51     #leaf_debugDiv .yui-content { padding:1em; }
52     .yui-nav li a:visited { color: #0000FF; }
53     </ style>
54     < script type="text/javascript">
55     YAHOO.example.init = function() {
56     var leafDebug = new YAHOO.widget.TabView('leaf_debugDiv');
57     };
58     YAHOO.example.init();
59     </ script>
60     DEBUG_STYLES;
61 }
```

```

62  /*
63  * Start Main Div
64  */
65      echo
66          "<div style=\"width: 600px;\">"
67          "<div id=\"leaf_debugDiv\" class=\"yui-navset yui-navset-
left\">"
68          "<ul class=\"yui-nav\">"
69
70          // tabs
71          echo
72              "<li title=\"active\" class=\"selected\"><a
href=\"#leaf_Debug_leaf_Statistics\"><em>leaf
Statistics</em></a></li>"
73              "<li><a href=\"#leaf_Debug_Global_Settings\"><em>Global
Settings</em></a></li>"
74              "<li><a
href=\"#leaf_Degug_Registry\"><em>Registry</em></a></li>"
75              "<li><a href=\"#leaf_Debug_Request\"><em>Request
Information</em></a></li>"
76
77          if ($reg->    config['allow_hooks']=="Yes"          ) {
78              echo
79                  "<li><a
href=\"#leaf_Debug_Hooks\"><em>Hooks</em></a></li>"
80          }
81
82          if ($reg->    config['log_level']!="None"            ) {
83              echo
84                  "<li><a href=\"#leaf_Debug_Log_Buffer\"><em>Log
Buffer</em></a></li>"
85          }
86
87          if ($reg->    config['allow_endorsed']=="Yes"        ) {
88              echo
89                  "<li><a href=\"#leaf_Debug_Endorsed\"><em>Endorsed
Classes</em></a></li>"
90          }
91
92          // tabs (end)
93          echo "</ul>"
94
95
96      /*
97      * Data
98      */
99      echo
100         "<div class=\"yui-content\">"
101
102         //
103         // Status
104         //
105         echo
106             "<div id=\"leaf_Debug_leaf_Statistics\" style=\"display:
block;\">"
107             . " <pre style=\"font-size: 12px; font-family: Verdana, Arial, helvetica,
sans-serif;\">"
108             . " <span style=\"color: #4e9a06;\"><b>leaf
Version</b></span>\n"
109             . " <span style=\"color: #ff0000;\"> " . LEAF_REL_VERSION .
" " . LEAF_REL_STATUS . "\n"
110             . " <span style=\"color: #4e9a06;\"><b>Script memory
usage</b></span>\n"
111             . " <span style=\"color: #ff0000;\"> { $memoryUsageKbs}
< small>< i>({ $memoryUsage} bytes)</ i></ small></ span>\n "
112             . " <span style=\"color: #4e9a06;\"><b>Total parsing
time</b></span>\n"
113             . " <span style=\"color: #ff0000;\"> { $parseTime}
seconds</ span> "
114             . " </pre>"
115             . "</div>"
116
117         //
118         // Settings
119         //
120         echo
121             "<div id=\"leaf_Debug_Global_Settings\" style=\"display:
none;\">"
122             . " <pre style=\"font-size: 12px; font-family: Verdana, Arial, helvetica,
sans-serif;\">"

```

```

123
124         echo
125         "<fieldset style=\"border: 0px solid
126         #ffffff;\"><legend><small><b>General</b></small></legend>&quot;
127         ;
128         foreach($reg->    config->    getByHashKey("general"                ) as $Var =>    $Val) {
129             echo
130             "
131             <span style=\"                color:
132             #4e9a06;\"><b>{    $Var}</b></span>\n                "
133             . "
134             <span style=\"                color:
135             #ff0000;\"><small>{    $Val}</small></span>\n                " ;
136         }
137         echo "</fieldset>" ;
138
139         echo "<br/>" ;
140
141         echo
142         "<fieldset style=\"border: 0px solid
143         #ffffff;\"><legend><small><b>Autoload</b></small></legend>&quot;
144         t;
145         foreach($reg->    config->    getByHashKey("autoload"                ) as $Var =>    $Val) {
146             echo
147             "
148             <span style=\"                color:
149             #4e9a06;\"><b>{    $Var}</b></span>\n                "
150             . "
151             <span style=\"                color:
152             #ff0000;\"><small>{    $Val}</small></span>\n                " ;
153         }
154         echo "</fieldset>" ;
155
156         echo
157         "<fieldset style=\"border: 0px solid
158         #ffffff;\"><legend><small><b>Endorsed</b></small></legend>&quot;
159         t;
160         echo "</fieldset>" ;
161
162         echo
163         "<fieldset style=\"border: 0px solid
164         #ffffff;\"><legend><small><b>Hooks</b></small></legend>" ;
165         echo "</fieldset>" ;
166
167         echo
168         " </pre>"
169         . " </div>" ;
170
171         //
172         // Registry
173         //
174         echo
175         "<div id=\"leaf_Degug_Registry\" style=\"display:
176         none;\">"
177         . " <pre style=\"font-size: 12px; font-family: Verdana, Arial, helvetica,
178         sans-serif;\">" ;
179
180         foreach($reg->    toArray() as $InstanceName =>    $ClassType)
181             echo
182             "
183             key: <span style=\"                color:
184             #4e9a06;\"><b>{    $InstanceName}</b></span>\n                "
185             . "
186             type: <span style=\"                color:
187             #ff0000;\"><b>{    $ClassType}</b></span>\n\n                " ;
188         }
189         echo
190         " </pre>"
191         . " </div>" ;
192
193         //
194         // Request
195         //
196         $className = $reg->    router->    getClassName();
197         $methodName = $reg->    router->    getMethodName();
198         $segments = implode(" " , (array)$reg->    router->    segments());
199         $queryString= $reg->    request->    getQueryStringAsString();
200
201         echo
202         "<div id=\"leaf_Debug_Request\" style=\"display:
203         none;\">"
204         . " <pre style=\"font-size: 12px; font-family: Verdana, Arial, helvetica,
205         sans-serif;\">"
206         . " <span style=\"color: #4e9a06;\"><b>Controller

```

```

(Class)</b></span>\n"
186 . " < span style=\" color: #ff0000;\">>
{$className}</ span>\n "
187 . " <span style=\"color: #4e9a06;\">><b>Action
(Method)</b></span>\n"
188 . " < span style=\" color: #ff0000;\">>
{$methodName}</ span>\n "
189 . " <span style=\"color: #4e9a06;\">><b>Extra
segments</b></span>\n"
190 . " < span style=\" color: #ff0000;\">>
{$segments}</ span>\n "
191 . " <span style=\"color: #4e9a06;\">><b>Query
String</b></span>\n"
192 . " < span style=\" color: #ff0000;\">>
{$queryString}</ span> "
193 . " </pre>"
194 . "</div>" ;
195
196 //
197 // Hooks
198 //
199 if ($reg-> config['allow_hooks']=="Yes" ) {
200 echo
201 "<div id=\"leaf_Debug_Hooks\" style=\"display:
none;\">>"
202 . "</div>" ;
203 }
204
205 //
206 // Log Buffer
207 //
208 if ($reg-> config['log_level']!="None" ) {
209 echo
210 "<div id=\"leaf_Debug_Log_Buffer\" style=\"display:
none;\">>"
211 . "</div>" ;
212 }
213
214 //
215 // Endorsed
216 //
217 if ($reg-> config['allow_endorsed']=="Yes" ) {
218 echo
219 "<div id=\"leaf_Debug_Endorsed\" style=\"display:
block;\">>"
220 . " <pre style=\"font-size: 14px; font-family: Verdana, Arial,
helvetica, sans-serif;\">>" ;
221
222 foreach($reg-> endorse_man-> getEndorsedClasses() as $Key => $Value)
223 {
224 $endorsedClass = key($Value);
225 $classObject = $Value[$endorsedClass]['reg_key'];
226 echo
227 "<span style=\"color: #4e9a06;\">>"
228 . " < b>{ $endorsedClass}</ b>\n "
229 . " < small style=\" color: #b91f49\">>{ $reg-
230 $classObject}</ small> "
231 . "</span>\n\n" ;
232 }
233
234 echo
235 " </pre>"
236 . "</div>" ;
237 }
238
239 /*
240 * End Main Div
241 */
242 echo
243 "</div>"
244 . "</div>" ;
245 }
246
247 ?>

```

# File Source for Front\_Controller.php

Documentation for this file is available at [Front\\_Controller.php](#)

```
1  i»¿<?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright  Copyright (c) 2007 Avraam Marimpis
9   * @license    http://leaf-framework.sourceforge.net/LICENSE/  New BSD License
10  * @link       http://leaf-framework.sourceforge.net
11  *
12  * @package    leaf
13  * @subpackage front
14  * @author     Avraam Marimpis <makism@users.sf.net>
15  * @version    $Id: Front_Controller.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  * @todo
18  * <ol>
19  * <li>Add functionality for the 'hooks' subsystem.</li>
20  * <li>Add functionality for the benchmarking subsystem.</li>
21  * <li>Completion of the documentation.</li>
22  * </ol>
23  */
24
25  /**
26   * Helper functions needed to start-up the framework
27   * and to load its classes.
28   */
29
30  require_once LEAF_BASE . 'front/helpers/Main.php';
31
32
33  /**
34   * Helper functions that are used to declare dependancies
35   * on specific extensions, functions or leaf's Classes.
36   */
37  require_once LEAF_BASE . 'front/helpers/Dependancies.php';
38
39
40  /**
41   * Custom error and exception handling functions.
42   */
43  require_once LEAF_BASE . 'front/helpers/Handlers.php';
44
45
46  /**
47   * Functions used to present errors.
48   */
49  require_once LEAF_BASE . 'front/helpers/Error.php';
50
51
52  /**
53   * Custom debug functions.
54   */
55  require_once LEAF_BASE . 'front/helpers/Debug.php';
56
57
58  /**
59   * Helper functions handling hooks.
60   *
61   * @see leaf_Hook
62   * @see leaf_Hook_Conditional
63   */
64  require_once LEAF_BASE . 'core/hook/helpers/Hooks.php';
65
66
67  /**
```

```

68      * Perform some basic checks to determine whether or not
69      * the system's sanity (and thus, all will be run normal).
70      */
71      #frameworkSanityCheck();
72
73
74      /*
75      * Startup leaf's Registry and register some basic objects.
76      */
77      $reg = leaf_Registry::getInstance();
78      $reg->register(new leaf_Config());
79      $reg->register(new leaf_Locale());
80      $reg->register(new leaf_Benchmark());
81
82
83      /*
84      * Configure timezone.
85      */
86      $timezoneSetting = (empty($reg->config['timezone']))
87          ? @date_default_timezone_get()
88          : $reg->config['timezone'];
89
90      date_default_timezone_set($timezoneSetting);
91
92
93      /*
94      * Handle errors and exceptions.
95      */
96      #set_error_handler("errorHandler");
97      #set_exception_handler("exceptionHandler");
98
99
100     /*
101     * Register the logger if it has been enabled.
102     */
103     #if ($reg->config['log_level']!="None" ||
104     #     empty($reg->config['log_level'])) {
105     #     $reg->register(new leaf_Logger());
106     #}
107
108
109     /*
110     * Register the "base" classes that are needed
111     * for leaf to work properly.
112     */
113     $reg->register(new leaf_Router());
114     $reg->register(new leaf_Request());
115     $reg->register(new leaf_Loader());
116     $reg->register(new leaf_Dispatcher());
117     $reg->register(new leaf_Response());
118
119
120     /*
121     * Begin output buffering.
122     */
123     $reg->response->outputBufferingStart();
124
125
126     /*
127     * Run all hooks for level: Pre-Controller-Dispatch
128     */
129     #runHooks(HOOK_PRE_CONTROLLER_DISPATCH);
130
131
132     /*
133     * Dispatch controller.
134     */
135     $reg->dispatcher->dispatchController();
136
137
138     /*
139     * Run all hooks for level: Post-Controller-Dispatch
140     */
141     #runHooks(HOOK_POST_CONTROLLER_DISPATCH);
142
143
144     /*
145     * Flush buffer, and turn off.
146     */
147     $reg->response->outputBufferingEnd();

```



```

148
149
150  /*
151   * Close the logger if it has been
152   * requested and registered.
153   */
154  #if ($reg->isRegistered("logger"))
155  #   $reg->logger->end_flush();
156
157
158  /*
159   * Run all hooks for level: Post-Front-Controller
160   */
161  #runHooks(HOOK_POST_FRONT_CONTROLLER);
162
163
164  /*
165   * General statistics like memory usage, parsing time and other.
166   */
167  if ($reg->    config['enable_debug_stats']=="Yes"          )
168      require_once LEAF_BASE . 'front/Debug.php';
169
170
171  ?>

```

# File Source for Debug.php

Documentation for this file is available at [Debug.php](#)

```
1  <?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright Copyright (c) 2007 Avraam Marimpis
9   * @license http://leaf-framework.sourceforge.net/LICENSE/ New BSD License
10  * @link http://leaf-framework.sourceforge.net
11  *
12  * @package leaf
13  * @subpackage front.helpers
14  * @author Avraam Marimpis <makism@users.sf.net>
15  * @version $Id: Debug.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  * @todo
18  * <ol>
19  * <li>Document.</li>
20  * <li>Possible removal.</li>
21  * </ol>
22  */
23
24
25  /*****
26   * It is suggested to used XDebug's functions since they are *
27   * natively implemented thus much faster. *
28   *****/
29
30
31  /**
32   *
33   *
34   * @param mixed $var
35   * @param boolean|integer$depth
36   * @return void
37   */
38  function i_var($var, $depth=NULL)
39  {
40      inspect_var($var, $depth);
41  }
42
43  /**
44   *
45   *
46   * @param mixed $var
47   * @param boolean|integer$depth
48   * @return void
49   */
50  function inspect_var($var, $depth=NULL)
51  {
52      $varType = gettype($var);
53
54      echo "<div>"
55          . "<pre style=\"font-family: Arial, sans; font-size: 14px;\">"
56
57      switch ($varType) {
58          case "integer" :
59          case "int" :
60          case "double" :
61          case "float" :
62              inspect_var_Number($var);
63              break;
64          case "string" :
65              inspect_var_String($var);
66              break;
67          case "object" :
```

```

68         _inspect_var_Object($var, (boolean)$depth);
69     break;
70     case "array" :
71         _inspect_var_Array($var, (integer)$depth);
72     break;
73     case "resource" :
74         _inspect_var_Resource($var);
75     break;
76     case "boolean" :
77         _inspect_var_Boolean($var);
78     break;
79     default:
80         _inspect_var_Generic($var);
81     break;
82 }
83
84 echo "</pre>"
85 . "</div>" ;
86 }
87
88 /**
89  *
90  *
91  * @access private
92  * @param resource $var
93  * @return void
94  */
95 function _inspect_var_Resource($var)
96 {
97     $resourceName = get_resource_type($var);
98
99     echo
100     "<span style=\"font-size: 11px; color:
101 #56a0ee\">(resource)</span> "
102     "< span style=\" color: #f06f00;\">{ $resourceName}</ span> "
103     "< span style=\" color: #3f6b5b; font-style: italic; font-size:
104 12px;\">({ $var})</ span> " ;
105 }
106
107 /**
108  *
109  *
110  * @access private
111  * @param mixed
112  * @return void
113  */
114 function _inspect_var_Generic($var)
115 {
116 }
117
118 /**
119  *
120  *
121  * @access private
122  * @param integer|double
123  * @return void
124  */
125 function _inspect_var_Number($num)
126 {
127     $varType = gettype($num);
128     echo
129     "< span style=\" font-size: 11px; color:
130 #000000\">({ $varType})</ span> "
131     "< span style=\" color: #ff0000;\">{ $num}</ span> " ;
132 }
133
134 /**
135  *
136  *
137  * @access private
138  * @param string
139  * @return void
140  */
141 function _inspect_var_String($str)
142 {
143     $len = strlen($str);
144     echo

```

```

144         "<span style=\"font-size: 11px; color: #4e9a06\">(string)</span>
145         "      < span style=\"      color: #0000ff;\">>'{$str}'</ span>      "
146         "      < span style=\"      font-style: italic; font-size:
12px;\">>({      length={$len})</ span>      "      ;
147     }
148
149 /**
150  *
151  *
152  * @access private
153  * @param boolean $flag
154  * @return string
155  */
156 function _inspect_var_Boolean($flag)
157 {
158     $str = ($flag===TRUE) ? "TRUE"          : "FALSE"          ;
159
160     echo
161     "<span style=\"font-size: 11px; color:
#999999\">(boolean)</span> "
162     "      < span style=\"      color: #75507b; font-weight:
bold;\">>{$str}</ span>      "      ;
163 }
164
165 /**
166  *
167  *
168  * @access private
169  * @param array $arr
170  * @param integer $recursionLevel
171  * @return mixed
172  */
173 function _inspect_var_Array(array $arr, $recursionLevel=5)
174 {
175
176 }
177
178
179 /**
180 /**
181  *
182  *
183  * @access private
184  * @param object $obj
185  * @param boolean $inDepth
186  * @return void
187  */
188 function _inspect_var_Object($obj, $inDepth=FALSE)
189 {
190     dependsOn('Reflection');
191
192     $reflect = new ReflectionClass($obj);
193
194     /*
195     *
196     */
197     $className = get_class($obj);
198     $parentName = get_parent_class($obj);
199     $subClasses = NULL;
200     $interfaces = NULL;
201     $interfacesStr = NULL;
202     $isUserDefined = $reflect-> isUserDefined();
203     $properties = $reflect-> getProperties();
204     $constants = $reflect-> getConstants();
205     $statics = $reflect-> getStaticProperties();
206
207
208     if ($parentName)
209         $subClasses = _inspect_var_Object_getSubClasses($parentName);
210
211     $interfaces = $reflect-> getInterfaces();
212     if (sizeof($interfaces)) {
213         foreach ($interfaces as $interface)
214             $interfacesStr .= $interface-> getName() . ", "      ;
215
216         $interfacesStr = substr($interfacesStr, 0, strlen($interfacesStr)-2);
217     }

```

```

218
219 /*
220 *
221 *
222 */
223 if ($inDepth==TRUE)
224 ;
225
226
227 /*
228 * Class name.
229 */
230 echo
231 "<fieldset style=\"border: 1px solid #c5c5c5;\">"
232 "    < legend> < i> object</ i> < b>{ $className}</ b>
</ legend> "
233 "    <div style=\"padding: 10px;\">
234 ;
235
236 /*
237 * Show information.
238 */
239 echo
240 "<div style=\"color: #3f6b5b; float: left; width: 150px; overflow:
hidden;\">Defined by</div>"
241 ;
242 echo
243 "<span style=\"font-weight: bold; font-style: italic; color:
#f06f00;\">"
244 "    (( $isUserDefined==TRUE ) ? "User" : "System" ) .
245 "</span>"
246 ;
247 echo "<hr />"
248 ;
249
250 echo
251 "<div style=\"color: #3f6b5b; float: left; width: 150px; overflow:
hidden;\">Is abstract</div>"
252 ;
253 echo
254 "<span style=\"font-weight: bold; font-style: italic; color:
#f06f00;\">"
255 "    (( $reflect-> isAbstract()==TRUE ) ? "Yes" : "No" ) .
256 "</span>"
257 ;
258 echo "<hr />"
259 ;
260
261 echo
262 "<div style=\"color: #3f6b5b; float: left; width: 150px; overflow:
hidden;\">Is interface</div>"
263 ;
264 echo
265 "<span style=\"font-weight: bold; font-style: italic; color:
#f06f00;\">"
266 "    (( $reflect-> isInterface()==TRUE ) ? "Yes" : "No" ) .
267 "</span>"
268 ;
269 echo "<hr />"
270 ;
271
272 /*
273 * Class filename.
274 */
275 echo
276 "<div style=\"color: #3f6b5b; float: left; width: 150px; overflow:
hidden;\">Filename</div>"
277 ;
278 echo
279 "<span style=\"font-weight: bold; font-style: italic; color:
#f06f00;\">"
280 "    $reflect-> getFileName() .
281 "</span>"
282 ;
283 echo "<hr />"
284 ;
285
286 /*
287 * The name of the classes this object extends.
288 */
289 echo
290 "<div style=\"color: #3f6b5b; float: left; width: 150px; overflow:
hidden;\">Extends</div>"
291 ;
292 echo
293 "<span style=\"font-weight: bold; font-style: italic; color:
#f06f00;\">"
294 "    (( $parentName!=NULL ) ? $subClasses : "none" ) .
295 "</span>"
296 ;
297 echo "<hr />"
298 ;
299
300 /*
301 * The list of the interfaces of class implements or inherits.

```

```

287     */
288     echo
289     "<div style=\"color: #3f6b5b; float: left; width: 150px; overflow:
hidden;\>Implements</div>"
290     echo
291     "<span style=\"font-weight: bold; font-style: italic; color:
#f06f00;\>"
292     (($interfacesStr!=NULL) ? $interfacesStr : "none"
293     "</span>"
294
295
296     echo "<br /><br /><br />"
297
298     /*
299     * Properties dumping.
300     */
301     echo "<ul><b>Properties</b>"
302     foreach ($properties as $prop) {
303         $status = ($prop-> isPublic())
304         ? "public"
305         : (($prop-> isPrivate()) ? "private" : "protected"
306         );
307
308         $value = "
309         if ($status=="public"
310         $value = " (value={$prop-> getValue($obj)})"
311
312     echo
313     "<li>"
314     " <span style=\" color: #4e9a06; font-size:
12px;\>{ $status}</span>"
315     " <span style=\" color: #56a0ee; font-style: italic;\>{ $prop-
> getName()}</span>"
316     "{ $value}"
317     "</li>"
318     }
319     echo "</ul>"
320
321     echo
322     "</div>"
323     "</legend>"
324 }
325
326 /**
327  *
328  *
329  * @access private
330  * @param object $obj
331  * @return string
332  */
333 function _inspect_var_Object_getSubclasses($obj)
334 {
335     $parent = get_parent_class($obj);
336
337     if ($parent!=NULL)
338         return $obj . ", " . _inspect_var_Object_getSubClasses($parent);
339     else
340         return $obj;
341 }
342
343 ?>

```

# File Source for Dependancies.php

Documentation for this file is available at [Dependancies.php](http://www.phpdoc.org)

```
1  <?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright Copyright (c) 2007 Avraam Marimpis
9   * @license http://leaf-framework.sourceforge.net/LICENSE/ New BSD License
10  * @link http://leaf-framework.sourceforge.net
11  *
12  * @package leaf
13  * @subpackage front.helpers
14  * @author Avraam Marimpis <makism@users.sf.net>
15  * @version $Id: Dependancies.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  * @todo
18  * <ol>
19  * <li>Implement a function that will handle dependencies as
20  * optional.</li>
21  * <li>Recheck <b>all</b> functions.</li>
22  * </ol>
23  */
24
25
26 /**
27  * Handles dependencies on PHP extensions.
28  *
29  * @param string/array $deps
30  * @return boolean
31  */
32 function dependsOn($deps)
33 {
34     if (is_array($deps)) {
35         foreach ($deps as $dependency)
36             if (!extension_loaded($dependency))
37                 showHtmlMessage(
38                     "Dependancy Failure"
39                     " Extension \"{$dependency}\" not found."
40                 );
41         die();
42     } else {
43         if (!extension_loaded($deps))
44             showHtmlMessage(
45                 "Dependancy Failure"
46                 " Extension \"{$deps}\" not found."
47                 TRUE
48             );
49     }
50
51     return TRUE;
52 }
53
54 /**
55  * Handles optional dependencies.
56  *
57  * @param string $deps
58  * @return boolean
59  * @todo
60  * <ol>
61  * <li>Implementd.</li>
62  * </ol>
63  */
64 function dependsOptionalOn($deps)
65 {
66     return TRUE;
67 }
```

```
68
69  /**
70   * Handles dependacies on specific functions.
71   *
72   * @param  string  $funcs
73   * @return boolean
74   */
75  function dependsOnFunc($func)
76  {
77      return function_exists($func);
78  }
79
80  ?>
```



# File Source for Error.php

Documentation for this file is available at [Error.php](#)

```
1  <?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright Copyright (c) 2007 Avraam Marimpis
9   * @license http://leaf-framework.sourceforge.net/LICENSE/ New BSD License
10  * @link http://leaf-framework.sourceforge.net
11  *
12  * @package leaf
13  * @subpackage front.helpers
14  * @author Avraam Marimpis <makism@users.sf.net>
15  * @version $Id: Error.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  * @todo
18  * <ol>
19  * <li>Implement a function that will handle dependancies as
20  * optional.</li>
21  * <li>Recheck <b>all</b> functions.</li>
22  * </ol>
23  */
24
25
26 /**
27  * Presents a "Page Not Found" message.
28  *
29  * @param string $str
30  * @return void
31  * @todo
32  * <ol>
33  * <li>Implement.</li>
34  * </ol>
35  */
36 function showErrorPage404($str=NULL)
37 {
38     die();
39 }
40
41 /**
42  * Prints styled text. Used in printing debug messages and errors.
43  *
44  * @param string $title
45  * @param string $str
46  * @param boolean $die
47  * @return void
48  */
49 function showHtmlMessage($title, $str=NULL, $die=FALSE)
50 {
51     static $hasOutputedHtml;
52
53     $errorMsg = (is_array($str))
54         ? implode("<br/>" , $str)
55         : $str;
56
57     if ($hasOutputedHtml==FALSE)
58     echo "
59         <style>
60         body {
61             padding: 10px;
62             font-family: Tahoma, sans-serif;
63             font-size: 12pt;
64         }
65         .topic {
66             width: 80px;
67             overflow: hidden;
```

```

68         float: left;
69         text-align: right;
70         padding: 0px 10px 0px 0px;
71     }
72     #msg {
73         color: red;
74         font-weight: bold;
75     }
76     #file {
77         color: green;
78         font: italic;
79     }
80     #line {
81         color: blue;
82         text-decoration: underline;
83     }
84 </style>"
85
86
87     echo "<div style=\"border: 1px solid #c5c5c5; padding: 10px;\">"
88     ."    < h2> $title</ h2>    "
89     ."        $errorMsg"
90     . "</div><br />"
91
92     $hasOutputedHtml = TRUE;
93
94     if ($die)
95         die();
96 }
97
98 ?>

```

# File Source for Handlers.php

Documentation for this file is available at [Handlers.php](#)

```
1  <?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright Copyright (c) 2007 Avraam Marimpis
9   * @license http://leaf-framework.sourceforge.net/LICENSE/ New BSD License
10  * @link http://leaf-framework.sourceforge.net
11  *
12  * @package leaf
13  * @subpackage front.helpers
14  * @author Avraam Marimpis <makism@users.sf.net>
15  * @version $Id: Handlers.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  * @todo
18  * <ol>
19  * <li>Implement a function that will handle dependancies as
20  * optional.</li>
21  * <li>Recheck <b>all</b> functions.</li>
22  * </ol>
23  */
24
25 /**
26  * Custom error handler.
27  *
28  * @param integer $errno
29  * @param string $errstr
30  * @param string $errfile
31  * @param integer $errline
32  * @return void
33  */
34
35 function errorHandler($errno, $errstr, $errfile, $errline)
36 {
37     static $errorTypes;
38
39     $errorTypes = array (
40         2 => "Warning" ,
41         8 => "Notice" ,
42         256 => "User Error" ,
43         512 => "User Warning" ,
44         1024 => "User Notice" ,
45         4096 => "Recoverable Error"
46     );
47
48     $dieStatus = ($errno==256) ? TRUE : FALSE;
49
50     showHtmlMessage(
51         $errorTypes[$errno],
52         "<div class=\"topic\">Message:</div> <span
53         id=\"msg\">" . $errstr
54         . " ({ $errno })</span><br/>\n "
55         "<div class=\"topic\">In file:</div> <span
56         id=\"file\">" . $errfile
57         . "</span><br/>\n"
58         "<div class=\"topic\">On line:</div> <span
59         id=\"line\">#" . $errline
60         . "</span>\n"
61         . $dieStatus
62     );
63
64     return true;
65 }
66 /**
```

```

65      * Custom exception handler.
66      *
67      * @param    object    Exception    $ex
68      * @return    void
69      */
70      function exceptionHandler(Exception $ex)
71      {
72          showHtmlMessage(
73              "Uncaught Exception"
74              , $ex->getMessage()
75              , true
76          );
77      }
78  }
79
80
81  ?>

```

# File Source for Main.php

Documentation for this file is available at [Main.php](#)

```
1  i»¿<?php
2  /**
3   * This source file is part of the leaf framework and
4   * is licensed under the New BSD license.
5   * For the full copyright and license information, please view the LICENSE
6   * file that was distributed with this source code.
7   *
8   * @copyright Copyright (c) 2007 Avraam Marimpis
9   * @license http://leaf-framework.sourceforge.net/LICENSE/ New BSD License
10  * @link http://leaf-framework.sourceforge.net
11  *
12  * @package leaf
13  * @subpackage front.helpers
14  * @author Avraam Marimpis <makism@users.sf.net>
15  * @version $Id: Main.php 24 2007-11-04 19:20:06Z makism $
16  * @filesource
17  */
18
19
20 /**
21  * Includes all necessary files related to leaf's classes.
22  *
23  * We use the "magic" function __autoload in order
24  * to have leaf's classes loaded automatically.<br>
25  * Also, {@link leaf_EndorsedManager Endorsed Manager} is
26  * instantiated if needed.
27  *
28  * @link http://www.php.net/manual/en/language.oop5.autoload.html
29  * @see leaf_EndorsedManager
30  * @param string $className
31  * @return void
32  */
33 function __autoload($className)
34 {
35     static $baseClasses;
36     static $hasInitd;
37     static $enableEndorsementManager;
38     static $endorsementManager;
39
40     /*
41      * Determine if EndorsedManager is enabled.
42      */
43     if ($hasInitd) {
44         if (leaf_Registry::getInstance()-> config['allow_endorsed']=="Yes" )
45             $enableEndorsementManager = TRUE;
46         $hasInitd=NULL;
47     }
48
49     /*
50      * If, EndorsedManager has been enabled, we
51      * instantiate it.
52      */
53     if ($endorsementManager==NULL && $enableEndorsementManager==TRUE) {
54         require_once LEAF_BASE . "core/EndorsementManager.php" ;
55         leaf_Registry::getInstance()-> register(new leaf_EndorsementManager());
56         $endorsementManager = leaf_Registry::getInstance()-> endorse_man;
57     }
58
59     /*
60      * Most of the leaf's classes.
61      */
62     if ($baseClasses==NULL)
63         $baseClasses = array (
64             'leaf_Base' => 'Base.php',
65             'leaf_Benchmark' => 'Benchmark.php',
66             'leaf_Config' => 'Config.php',
67             'leaf_Controller'=> 'Controller.php',
```

```

68         'leaf_Debug'      =>    'Debug.php',
69         'leaf_Dispatcher' =>    'Dispatcher.php',
70         'leaf_Exception' =>    'Exception.php',
71         'leaf_Hash'       =>    'Hash.php',
72         'leaf_Hook'       =>    'hook/Hook.php',
73         'leaf_Hook_Conditional' => 'hook/Hook_Conditional.php',
74         'leaf_Loader'     =>    'Loader.php',
75         'leaf_Locale'     =>    'Locale.php',
76         'leaf_Log'        =>    'log/Log.php',
77         'leaf_Logger'     =>    'log/Logger.php',
78         'leaf_Model'      =>    'Model.php',
79         'leaf_Registry'   =>    'Registry.php',
80         'leaf_Request'    =>    'Request.php',
81         'leaf_Response'   =>    'Response.php',
82         'leaf_Router'     =>    'Router.php',
83         'leaf_View'       =>    'View.php'
84     );
85
86     /*
87     * Check to dermine whether or not, the requested class
88     * is part of leaf`s core.
89     */
90     if (array_key_exists($className, $baseClasses)) {
91
92         /*
93         * We let the EndorsedManager to handle the external
94         * classes that will overlap the internal ones.
95         */
96         if ($enableEndorsementManager) {
97             if ($endorsementManager->isEndorsed($className))
98                 $endorsementManager->loadEndorsedClass($className);
99             else
100                 require_once LEAF_BASE
101                     . 'core/'
102                     . $baseClasses[$className];
103         } else {
104             require_once LEAF_BASE
105                 . 'core/'
106                 . $baseClasses[$className];
107         }
108     }
109
110     /*
111     * If the class {@link leaf_Config} has been called,
112     * we can access the configuration parameters and
113     * determine if EndorsedManager is enabled.
114     */
115     if ($className=="leaf_Config" )
116         $hasInitied = TRUE;
117 }
118
119 /**
120 * Terminates the execution of the framework.
121 *
122 * @return void
123 */
124 function frameworkForceTerminate()
125 {
126     showHtmlMessage("Forced Termination" , NULL, TRUE);
127 }
128
129 /**
130 * Performs some basic tests to check whether
131 * the framework is usable or not.
132 *
133 * @param    boolean    $checkAll
134 * @return    void
135 */
136 function frameworkSanityCheck($checkAll=false)
137 {
138     $msg = array();
139
140     if (extension_loaded('spl')==FALSE) {
141         $msg[] = "SPL functions are disabled." ;
142     }
143
144     /*
145     * LEAF_REL_STATUS
146     */
147     if (!defined('LEAF_REL_STATUS'))

```

```

148     $msg[] = "No releasing status information found - "
149         . "Missing \"LEAF_REL_STATUS\" declaration.\n"
150
151     /*
152     * LEAF_WORKING_DIR
153     */
154     if (!defined('LEAF_WORKING_DIR')) {
155         $msg[] = "No Working directory is defined - "
156             . "Missing \"LEAF_WORKING_DIR\" declaration.\n"
157     } else {
158         if (!is_readable(LEAF_WORKING_DIR)) {
159             $msg[] = "Working directory is NOT readable.\n"
160         } else {
161             if (is_writeable(LEAF_WORKING_DIR)) {
162                 $msg[] = "Working directory IS writeable."
163                 . "Security issues arise.\n"
164             }
165         }
166     }
167
168     /*
169     * LEAF_BASE
170     */
171     if (!defined('LEAF_BASE')) {
172         $msg[] = "No leaf Base directory is defined - "
173             . "Missing \"LEAF_BASE\" declaration.\n"
174     } else {
175         if (!file_exists(LEAF_BASE)) {
176             $msg[] = "leaf Base directory NOT found.\n"
177         } else {
178             if (!is_readable(LEAF_BASE)) {
179                 $msg[] = "leaf Base directory is NOT readable.\n"
180             } else {
181                 if (is_writeable(LEAF_BASE)) {
182                     $msg[] = "leaf Base directory IS writeable."
183                     . "Security issues arise.\n"
184                 }
185             }
186         }
187     }
188
189     /*
190     * LEAF_APPS
191     */
192     if (!defined('LEAF_APPS')) {
193         $msg[] = "No leaf Application directory is defined - "
194             . "Missing \"LEAF_APPS\" declaration.\n"
195     } else {
196         if (!file_exists(LEAF_APPS)) {
197             $msg[] = "leaf Application directory NOT found.\n"
198         } else {
199             if (!is_readable(LEAF_APPS)) {
200                 $msg[] = "leaf Application directory is NOT readable.\n"
201             }
202         }
203     }
204
205     /*
206     * LEAF_VAR
207     */
208     if (!defined('LEAF_VAR')) {
209         $msg[] = "No Variable directory is defined - "
210             . "Missing \"LEAF_VAR\" declaration.\n"
211     } else {
212         if (!is_readable(LEAF_VAR)) {
213             $msg[] = "Variable directory is NOT readable.\n"
214         } else {
215             if (is_writable(LEAF_VAR)) {
216                 $msg[] = "Variable directory IS writable."
217                 . "Security issues arise.\n"
218             } else {
219                 if (!is_readable(LEAF_VAR . 'cache/')) {
220                     $msg[] = "Cache directory is NOT readable.\n"
221                 } else {
222                     if (!is_writeable(LEAF_VAR . 'cache/')) {
223                         $msg[] = "Cache directory is NOT writable.\n"
224                     }
225                 }
226             }
227         }

```

```

228         if (!is_readable(LEAF_VAR . 'logs/')) {
229             $msg[] = "Logs directory is NOT readable.\n" ;
230         } else {
231             if (!is_writeable(LEAF_VAR . 'logs/')) {
232                 $msg[] = "Logs directory is NOT writable.\n" ;
233             }
234         }
235     }
236
237     if (!is_readable(LEAF_VAR . 'sigs/')) {
238         $msg[] = "Sigs directory is NOT readable.\n" ;
239     } else {
240         if (!is_writeable(LEAF_VAR . 'sigs/')) {
241             $msg[] = "Sigs directory is NOT writable.\n" ;
242         }
243     }
244
245     if (!is_readable(LEAF_VAR . 'tmp/')) {
246         $msg[] = "Tmp directory is NOT readable.\n" ;
247     } else {
248         if (!is_writeable(LEAF_VAR . 'tmp/')) {
249             $msg[] = "Tmp directory is NOT writable.\n" ;
250         }
251     }
252 }
253
254 }
255
256 if ($checkAll) {
257     /*
258     * ETC
259     */
260     if (!file_exists(LEAF_BASE . 'Etc/')) {
261         $msg[] = "Etc directory NOT found.\n" ;
262     } else {
263         if (!is_readable(LEAF_BASE . 'Etc/')) {
264             $msg[] = "Etc directory is NOT readable.\n" ;
265         }
266     }
267
268     /*
269     * CONFIGURATION FILES
270     */
271     $configFiles = array (
272         "autoload.php" , "config.php" , "database.php" ,
273         "hooks.php" , "routes.php"
274     );
275     foreach ($configFiles as $file) {
276         if (!file_exists(LEAF_BASE . 'Etc/' . $file)) {
277             $msg[] = "Configuration file \"{$file}\", does not exist.\n" ;
278         } else {
279             if (!is_readable(LEAF_BASE . 'Etc/' . $file)) {
280                 $msg[] = "Configuration file \"{$file}\", cannot be
281 read.\n" ;
282             }
283         }
284     }
285     unset($configFiles);
286 }
287
288
289 if (sizeof($msg)> 0)
290     showHtmlMessage("Sanity Check Failure" , $msg, true);
291 }
292
293 ?>

```



# Appendix D - Todo List

## In Package leaf

In [Debug.php](#)

- 1. Document.
  2. Recheck the information displayed under the tab "Config Settings".

In [Debug.php](#)

- 1. Document.
  2. Possible removal.

In [Dependancies.php](#)

- 1. Implement a function that will handle dependancies as optional.
  2. Recheck **all** functions.

In [dependsOptionalOn\(\)](#)

- 1. Implementd.

In Error.php

- 1. Implement a function that will handle dependancies as optional.
- 2. Recheck **all** functions.

In Front Controller.php

- 1. Add functionality for the 'hooks' subsystem.
  2. Add functionality for the benchmarking subsystem.
  3. Completion of the documentation.

In leaf Request::getQueryString()

1.  $\hat{I} \hat{I}^\dagger \hat{I}, \hat{I} \pm \hat{I}^2 \hat{I} \otimes \hat{I} \pm \hat{I} \gg \hat{I} \pm \hat{I}^3 \hat{I} \otimes \hat{I}, \hat{I} \hat{I}^2 \hat{I} \otimes \hat{I} \hat{I}^2 \hat{I} \otimes \hat{I} \hat{I}^2 \hat{I},$

In [leaf\\_Request::getQueryStringAsString\(\)](#)

- [illegible]

In Handlers.php

-

1. Implement a function that will handle dependencies as optional.
2. Recheck **all** functions.

In `leaf EndorsementManager::introspectEndorsedClass()`

- [illegible]

In leaf Benchmark:

1.  $\forall i \in I, i \in I^{-} \Rightarrow f(i) = 0$ .

In leaf Config

1. Maybe we should unset the global variable `$GLOBALS` somewhere else.
2. Possible implementation of the interface *Iterator*.
3. Possible implementation of a subclass that will read separate configuration files for each Controller, thus declaring different parameters and maybe unique for each Controller.

In **leaf Controller**:

- [illegible]

In leaf EndorsementManager

1.  $\exists x \in \mathbb{R} \text{ s.t. } x^2 = 2$

In leaf Exception

- [illegible]

In leaf\_Hash:

1.  $\hat{I} \hat{I}_1^{\dagger} \hat{J}_{\pm} \hat{I}_{\pm}^{\dagger} \hat{I} @ \hat{I} \in \pm i \bullet \hat{I}_{\pm} \hat{I}_{\pm}^{\dagger} @ \hat{I} \mu \hat{I}_1^{\dagger} \hat{I} \text{CEI}$ , interface.

In leaf Hook

1.  $\hat{M} \in \hat{\mathcal{M}}_f$ .

In leaf Hook Conditional

1.  $\hat{M} \hat{N} \hat{P} \hat{Q} \hat{R} \hat{S} \hat{T} \hat{U} \hat{V} \hat{W} \hat{X} \hat{Y} \hat{Z}$

In leaf Input

1.  $\hat{M} \in \mathcal{M}_n(\mathbb{C})$  is a normal matrix.

In leaf\_loader

1.  $\hat{M} \gg \hat{L} \in \hat{L}^{\perp} \cdot \hat{f}$ .

In leaf Logger

- [illegible]

In leaf Model

1.  $\hat{\mathbb{M}} \in \hat{\mathcal{L}}(\hat{f})$ .

In leaf Registry

- [illegible]

In leaf Request

- [illegible]

In leaf Response

1.  $\hat{M} \gg \hat{L} \in \hat{L}^{-1} \hat{f}$ .

In leaf Router

1.  $\text{leaf\_Request};$
2.  $\text{leaf\_Request};$

In leaf View:

1.  $\hat{A} \neq \hat{B} \Rightarrow \hat{A} \in \hat{A}^{-1} \cdot \hat{f} \hat{A}.$

In leaf EndorsementManager::loadEndorsedClass()

-

- leaf Registry.

## leaf Request::reconstructUrl()

1.  $\hat{A} \neq \hat{B} \Rightarrow \hat{A} \in \hat{C} \wedge \hat{B} \notin \hat{C} \wedge \hat{A} \in \hat{f}.$

## leaf Request::redirect()

- $$1. \quad \hat{A} \neq \hat{A} \gg \hat{A}_\zeta \in \hat{A}_\zeta \hat{A}^{-1} \cdot f \hat{A}.$$

## leaf Registry::register()

- [illegible]

## leaf Request::segment()

1.  $\hat{A} \neq \hat{B} \Rightarrow \hat{A} \in \hat{A}^{-1} \cdot f \hat{A}$ .

## leaf\_Request::segmentsAsArray()

- 

1. `if ($?) { $ErrorActionPreference = 'Continue' }`

In [showErrorPage404\(\)](#)

- 

1. Implement.

In [leaf Request::totalSegments\(\)](#)

- 

1. `if ($?) { $ErrorActionPreference = 'Continue' }`

In [leaf Router::construct\(\)](#)

- 

1. "if (\$?) { \$ErrorActionPreference = 'Continue' }" i.e. 1/2 of the code is in the constructor if the constructor is not called, the code is in the constructor.
2. if (\$?) { \$ErrorActionPreference = 'Continue' } i.e. 1/2 of the code is in the constructor if the constructor is not called, the code is in the constructor.



# Index

## B

|                                                                                           |     |
|-------------------------------------------------------------------------------------------|-----|
| <a href="#">Benchmark.php</a>                                                             | 134 |
| Source code                                                                               |     |
| <a href="#">Benchmark.php</a>                                                             | 100 |
| Source code                                                                               |     |
| <a href="#">Benchmark.php</a>                                                             | 68  |
| leaf Framework                                                                            |     |
| <a href="#">Benchmark.php</a>                                                             | 5   |
| leaf Framework                                                                            |     |
| <a href="#">Base.php</a>                                                                  | 4   |
| This source file is part of the leaf framework and is licensed under the New BSD license. |     |

## C

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| <a href="#">constructor leaf Response:: construct()</a>                          | 59  |
| Class constructor                                                                |     |
| <a href="#">constructor leaf Request:: construct()</a>                           | 55  |
| <a href="#">leaf Router</a>                                                      |     |
| <a href="#">constructor leaf Registry:: construct()</a>                          | 50  |
| Class constructor                                                                |     |
| <a href="#">constructor leaf Model:: construct()</a>                             | 48  |
| <a href="#">constructor leaf Router:: construct()</a>                            | 64  |
| <a href="#">constructor leaf View:: construct()</a>                              | 66  |
| <a href="#">Controller.php</a>                                                   | 102 |
| Source code                                                                      |     |
| <a href="#">CHANGELOG</a>                                                        | 97  |
| <a href="#">constructor leaf Hook Conditional:: construct()</a>                  | 73  |
| <a href="#">constructor leaf Hook:: construct()</a>                              | 72  |
| <a href="#">constructor leaf Logger:: construct()</a>                            | 46  |
| Class constructor                                                                |     |
| <a href="#">constructor leaf Loader:: construct()</a>                            | 42  |
| <a href="#">constructor leaf Config:: construct()</a>                            | 26  |
| Encapsulates all the configuration files, and stores the paremeters in an array. |     |
| <a href="#">constructor leaf Benchmark:: construct()</a>                         | 24  |
| <a href="#">constructor leaf Base:: construct()</a>                              | 22  |
| <a href="#">Registry</a>                                                         |     |
| <a href="#">Controller.php</a>                                                   | 7   |



|                                                                                              |     |
|----------------------------------------------------------------------------------------------|-----|
| Source code                                                                                  | 154 |
| <a href="#">Error.php</a>                                                                    | 154 |
| Source code                                                                                  | 89  |
| <a href="#">exceptionHandler()</a>                                                           | 89  |
| Custom exception handler.                                                                    |     |
| <a href="#">errorHandler()</a>                                                               | 88  |
| Custom error handler.                                                                        |     |
| <a href="#">Exception.php</a>                                                                | 10  |
| leaf Framework                                                                               |     |
| <a href="#">Error.php</a>                                                                    | 86  |
| This source file is part of the leaf framework and<br>is licensed under the New BSD license. |     |
| <a href="#">EndorsementManager.php</a>                                                       | 9   |
| leaf Framework                                                                               |     |

## F

|                                                                                              |     |
|----------------------------------------------------------------------------------------------|-----|
| <a href="#">Front_Controller.php</a>                                                         | 144 |
| Source code                                                                                  |     |
| <a href="#">frameworkSanityCheck()</a>                                                       | 90  |
| Performs some basic tests to check whether<br>the framework is usable or not.                |     |
| <a href="#">frameworkForceTerminate()</a>                                                    | 90  |
| Terminates the execution of the framework.                                                   |     |
| <a href="#">Front_Controller.php</a>                                                         | 78  |
| This source file is part of the leaf framework and<br>is licensed under the New BSD license. |     |

## G

|                                                                                                                                                                                                                                                                                                                         |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <a href="#">getHooks()</a>                                                                                                                                                                                                                                                                                              | 74 |
| Returns all registered hooks for one -or all- level. Also, the hooks<br>can be filtered using the parameters \$controller and \$method.<br><br>Returns an array of hooks. The array contains the following information:<br>- controller: the controller name<br>- method: the method name<br>- hooks: an array of hooks |    |

## H

|                                                                                              |     |
|----------------------------------------------------------------------------------------------|-----|
| <a href="#">Hook.php</a>                                                                     | 136 |
| Source code                                                                                  |     |
| <a href="#">Hash.php</a>                                                                     | 109 |
| Source code                                                                                  |     |
| <a href="#">Hook_Conditional.php</a>                                                         | 137 |
| Source code                                                                                  |     |
| <a href="#">Hooks.php</a>                                                                    | 138 |
| Source code                                                                                  |     |
| <a href="#">Handlers.php</a>                                                                 | 156 |
| Source code                                                                                  |     |
| <a href="#">Handlers.php</a>                                                                 | 88  |
| This source file is part of the leaf framework and<br>is licensed under the New BSD license. |     |



|                                                    |    |
|----------------------------------------------------|----|
| <i>Leaf Framework Class</i>                        |    |
| <a href="#">leaf Response::LEAF_REG_KEY</a>        | 59 |
| <a href="#">leaf Response::\$buffer</a>            | 59 |
| <a href="#">leaf Request::__toString()</a>         | 58 |
| <a href="#">leaf Request::totalSegments()</a>      | 58 |
| <a href="#">leaf Request::redirect()</a>           | 57 |
| <a href="#">leaf Request::segment()</a>            | 57 |
| <a href="#">leaf Request::segmentsToArray()</a>    | 57 |
| <a href="#">leaf Request::LEAF_CLASS_ID</a>        | 53 |
| <a href="#">leaf Request</a>                       | 53 |
| <i>Leaf Framework Class</i>                        |    |
| <a href="#">leaf Logger::__call()</a>              | 47 |
| <i>__call</i>                                      |    |
| <a href="#">leaf Model</a>                         | 48 |
| <a href="#">leaf Model::LEAF_CLASS_ID</a>          | 48 |
| <a href="#">leaf Model::LEAF_REG_KEY</a>           | 48 |
| <a href="#">leaf Logger::setTimeStampPattern()</a> | 47 |
| <i>setTimeStampPattern</i>                         |    |
| <a href="#">leaf Logger::setLevel()</a>            | 47 |
| <i>setLevel</i>                                    |    |
| <a href="#">leaf Logger::\$threshold</a>           | 45 |
| <a href="#">leaf Logger::end_flush()</a>           | 46 |
| <i>end_flush</i>                                   |    |
| <a href="#">leaf Logger::flush()</a>               | 46 |
| <i>flush</i>                                       |    |
| <a href="#">leaf Registry</a>                      | 49 |
| <i>leaf_Framework Registry Class</i>               |    |
| <a href="#">leaf Registry::\$instance</a>          | 49 |
| <a href="#">leaf Registry::toArray()</a>           | 51 |
| <a href="#">leaf Registry::unregister()</a>        | 51 |
| <a href="#">leaf Registry::__get()</a>             | 52 |
| <i>__get</i>                                       |    |
| <a href="#">leaf Registry::__set()</a>             | 52 |
| <i>__set</i>                                       |    |
| <a href="#">leaf Registry::register()</a>          | 51 |
| <i>register</i>                                    |    |
| <a href="#">leaf Registry::isRegistered()</a>      | 50 |
| <i>isRegistered</i>                                |    |
| <a href="#">leaf Registry::\$registered</a>        | 49 |
| <a href="#">leaf Registry::getInstance()</a>       | 50 |
| <a href="#">leaf Registry::instance()</a>          | 50 |
| <i>instance</i>                                    |    |
| <a href="#">leaf Response::\$useGzip</a>           | 59 |
| <a href="#">leaf Response::\$useTidy</a>           | 59 |
| <a href="#">leaf View::LEAF_REG_KEY</a>            | 66 |
| <a href="#">leaf View::__call()</a>                | 67 |
| <a href="#">leaf View::__toString()</a>            | 67 |
| <a href="#">leaf Hook</a>                          | 71 |
| <a href="#">leaf View::LEAF_CLASS_ID</a>           | 66 |
| <a href="#">leaf View</a>                          | 66 |
| <a href="#">leaf Router::segments()</a>            | 65 |
| <i>Leaf Framework Class</i>                        |    |

|                                                      |     |
|------------------------------------------------------|-----|
| <a href="#">leaf Router::segmentsSize()</a>          | 65  |
| <a href="#">leaf Router::toString()</a>              | 66  |
| <a href="#">leaf Hook::LEAF_CLASS_ID</a>             | 72  |
| <a href="#">leaf Hook::LEAF_REG_KEY</a>              | 72  |
| <a href="#">leaf Log</a>                             | 76  |
| <i>leaf Framework</i>                                |     |
| <a href="#">LICENSE</a>                              | 96  |
| <a href="#">Loader.php</a>                           | 113 |
| <i>Source code</i>                                   |     |
| <a href="#">Logger.php</a>                           | 115 |
| <i>Source code</i>                                   |     |
| <a href="#">leaf Hook Conditional::condition()</a>   | 73  |
| <a href="#">leaf Hook Conditional::LEAF_REG_KEY</a>  | 73  |
| <a href="#">leaf Hook::toString()</a>                | 72  |
| <a href="#">leaf Hook Conditional</a>                | 72  |
| <a href="#">leaf Hook Conditional::LEAF_CLASS_ID</a> | 73  |
| <a href="#">leaf Router::queryStringElements()</a>   | 65  |
| <a href="#">leaf Router::getMethodName()</a>         | 65  |
| <a href="#">leaf Response::sendResponse()</a>        | 61  |
| <a href="#">leaf Response::setHeader()</a>           | 61  |
| <a href="#">leaf Response::toString()</a>            | 61  |
| <a href="#">leaf Router</a>                          | 62  |
| <a href="#">leaf Response::outputBufferingEnd()</a>  | 61  |
| <i>End output buffering</i>                          |     |
| <a href="#">leaf Response::ouputBufferingStart()</a> | 60  |
| <i>Start output buffering</i>                        |     |
| <a href="#">leaf Response::clearHeaders()</a>        | 60  |
| <a href="#">leaf Response::flushOutputBuffer()</a>   | 60  |
| <a href="#">leaf Response::getOutputBuffer()</a>     | 60  |
| <i>Gets output buffer.</i>                           |     |
| <a href="#">leaf Router::LEAF_CLASS_ID</a>           | 62  |
| <a href="#">leaf Router::LEAF_REG_KEY</a>            | 62  |
| <a href="#">leaf Router::\$segments</a>              | 63  |
| <a href="#">leaf Router::chopSegment()</a>           | 64  |
| <a href="#">leaf Router::getClassName()</a>          | 64  |
| <a href="#">leaf Router::\$requestUri</a>            | 63  |
| <a href="#">leaf Router::\$requestMethod</a>         | 63  |
| <a href="#">leaf Router::\$queryString</a>           | 62  |
| <a href="#">leaf Router::\$queryStringElements</a>   | 63  |
| <a href="#">leaf Router::\$requestClass</a>          | 63  |
| <a href="#">leaf Logger::\$stampPat</a>              | 45  |
| <a href="#">leaf Logger::\$levels</a>                | 45  |
| <a href="#">leaf Config::offsetGet()</a>             | 27  |
| <i>Returns the value of the specified parameter.</i> |     |
| <a href="#">leaf Config::offsetSet()</a>             | 27  |
| <i>Sets a value in the specific parameter.</i>       |     |



|                                                                    |    |
|--------------------------------------------------------------------|----|
| <a href="#">leaf_Base::LEAF_REG_KEY</a>                            | 22 |
| <a href="#">leaf_Base::Registry</a>                                | 22 |
| <a href="#">leaf_Benchmark::LEAF_REG_KEY</a>                       | 24 |
| <a href="#">leaf_Benchmark::\$indexTable</a>                       | 24 |
| <a href="#">leaf_Benchmark::addIndex()</a>                         | 24 |
| <a href="#">leaf_Benchmark::LEAF_CLASS_ID</a>                      | 24 |
| <a href="#">leaf_Benchmark</a>                                     | 23 |
| <a href="#">leaf_Base::clone()</a>                                 | 22 |
| <a href="#">leaf_Base::get()</a>                                   | 23 |
| <a href="#">leaf_Base::toString()</a>                              | 23 |
| <a href="#">leaf_Dispatcher::toString()</a>                        | 31 |
| <a href="#">leaf_EndorsementManager</a>                            | 32 |
| <a href="#">leaf_Loader</a>                                        | 41 |
| <a href="#">leaf_Loader::LEAF_CLASS_ID</a>                         | 42 |
| <a href="#">leaf_Loader::LEAF_REG_KEY</a>                          | 42 |
| <a href="#">leaf_Loader::endorsed()</a>                            | 42 |
| <a href="#">leaf_Input::offsetUnset()</a>                          | 41 |
| <a href="#">leaf_Input::offsetSet()</a>                            | 41 |
| <a href="#">leaf_Input::\$input</a>                                | 40 |
| <a href="#">leaf_Input::offsetExists()</a>                         | 40 |
| <a href="#">leaf_Input::offsetGet()</a>                            | 40 |
| <a href="#">leaf_Loader::extension()</a>                           | 42 |
| <a href="#">leaf_Loader::library()</a>                             | 43 |
| <a href="#">leaf_Logger::LEAF_REG_KEY</a>                          | 44 |
| <a href="#">leaf_Logger::\$buffer</a>                              | 44 |
| <a href="#">leaf_Logger::\$filename</a>                            | 45 |
| <a href="#">leaf_Logger::\$fp</a>                                  | 45 |
| <a href="#">leaf_Logger::LEAF_CLASS_ID</a>                         | 44 |
| <a href="#">leaf_Logger</a>                                        | 44 |
| <a href="#">leaf_Loader::model()</a>                               | 43 |
| <a href="#">leaf_Loader::plugin()</a>                              | 43 |
| <a href="#">leaf_Loader::toString()</a>                            | 43 |
| <a href="#">leaf_Input::LEAF_REG_KEY</a>                           | 40 |
| <a href="#">leaf_Input::LEAF_CLASS_ID</a>                          | 39 |
| <a href="#">leaf_EndorsementManager::\$registeredEndorsed</a>      | 33 |
| <a href="#">leaf_EndorsementManager::getEndorsedClasses()</a>      | 34 |
| <a href="#">leaf_EndorsementManager::introspectEndorsedClass()</a> | 34 |
| <a href="#">leaf_EndorsementManager::isEndorsed()</a>              | 34 |
| <a href="#">leaf_EndorsementManager::\$endorsed</a>                | 33 |



|                                                              |    |
|--------------------------------------------------------------|----|
| <a href="#">leaf_EndorsementManager::\$currentClass</a>      | 33 |
| <a href="#">leaf_EndorsementManager::LEAF_CLASS_ID</a>       | 32 |
| <a href="#">leaf_EndorsementManager::LEAF_REG_KEY</a>        | 32 |
| <a href="#">leaf_EndorsementManager::\$allowEndorsement</a>  | 32 |
| <a href="#">leaf_EndorsementManager::loadEndorsedClass()</a> | 35 |
| <a href="#">leaf_Exception</a>                               | 35 |
| Exception Class                                              |    |
| <a href="#">leaf_Hash::digestMsg()</a>                       | 38 |
| <a href="#">leaf_Hash::setAlgorithm()</a>                    | 39 |
| <a href="#">leaf_Input</a>                                   | 39 |
| <a href="#">leaf_Hash::digestFile()</a>                      | 38 |
| <a href="#">leaf_Hash::\$useAltMethod</a>                    | 37 |
| <a href="#">leaf_Hash</a>                                    | 36 |
| <a href="#">leaf_Hash::\$algorithms</a>                      | 37 |
| <a href="#">leaf_Hash::\$defaultAlgorithm</a>                | 37 |
| <a href="#">LEAF_APPS</a>                                    | 2  |
| Subdirectory in which user's applications are located.       |    |

## M

|                                                                                           |     |
|-------------------------------------------------------------------------------------------|-----|
| <a href="#">Main.php</a>                                                                  | 158 |
| Source code                                                                               |     |
| <a href="#">Model.php</a>                                                                 | 119 |
| Source code                                                                               |     |
| <a href="#">Main.php</a>                                                                  | 90  |
| This source file is part of the leaf framework and is licensed under the New BSD license. |     |
| <a href="#">Model.php</a>                                                                 | 15  |
| leaf Framework<br>                                                                        |     |

## R

|                              |     |
|------------------------------|-----|
| <a href="#">Request.php</a>  | 120 |
| Source code                  |     |
| <a href="#">README</a>       | 96  |
| <a href="#">Response.php</a> | 124 |
| Source code                  |     |
| <a href="#">Router.php</a>   | 127 |

|                                                                                                                                   |     |
|-----------------------------------------------------------------------------------------------------------------------------------|-----|
| Source code                                                                                                                       | 135 |
| <a href="#">Request.php</a>                                                                                                       | 135 |
| Source code                                                                                                                       | 75  |
| <a href="#">runHooks()</a>                                                                                                        | 75  |
| Runs all registered hooks in one level.<br><br><i>İ•İöİ„İµİ»İµİ İÇİİ»İÇİİ...İ, İ„İÇİİ...İ, hooks İfİµ İ-İ½İ± İµİ€İ İ€İµİ İÇİ.</i> | 75  |
| <a href="#">runHook()</a>                                                                                                         | 75  |
| Executes a specific hook.<br><br><i>İ•İöİ„İµİ»İµİ İ-İ½İ±İ½ İfİ...İİİİİµİİİ•İİ¼İ-İ½İÇİ hook.</i>                                   | 17  |
| <a href="#">Request.php</a>                                                                                                       | 18  |
| leaf Framework                                                                                                                    | 19  |
| <a href="#">Response.php</a>                                                                                                      | 19  |
| leaf Framework                                                                                                                    | 69  |
| <a href="#">Router.php</a>                                                                                                        | 69  |
| leaf Framework                                                                                                                    | 16  |
| <a href="#">Request.php</a>                                                                                                       | 16  |
| leaf Framework                                                                                                                    |     |
| <a href="#">Registry.php</a>                                                                                                      |     |
| leaf Framework                                                                                                                    |     |

## S

|                                                                 |    |
|-----------------------------------------------------------------|----|
| <a href="#">showHtmlMessage()</a>                               | 87 |
| Prints styled text. Used in printing debug messages and errors. |    |
| <a href="#">showErrorPage404()</a>                              | 86 |
| Presents a "Page Not Found" message.                            |    |

## T

|                      |    |
|----------------------|----|
| <a href="#">TODO</a> | 96 |
|----------------------|----|

## V

|                          |     |
|--------------------------|-----|
| <a href="#">View.php</a> | 132 |
| Source code              | 20  |
| <a href="#">View.php</a> | 20  |
| leaf Framework<br>       |     |

|                                                         |    |
|---------------------------------------------------------|----|
| <a href="#">inspect var Resource()</a>                  | 82 |
| <a href="#">inspect var String()</a>                    | 83 |
| <a href="#">autoload()</a>                              | 90 |
| Includes all necessary files related to leaf's classes. |    |
| <a href="#">inspect var Object_getSubclasses()</a>      | 82 |
| <a href="#">inspect var Object()</a>                    | 82 |
| <a href="#">inspect var Boolean()</a>                   | 81 |
| <a href="#">inspect var Generic()</a>                   | 81 |
| <a href="#">inspect var Number()</a>                    | 81 |

|                                                |    |
|------------------------------------------------|----|
| <a href="#">_inspect_var_Array()</a> . . . . . | 81 |
|------------------------------------------------|----|