
JPHP Documentation

Release 1.0 alpha1

JPHP group

Jul 07, 2017

Contents

1	API (English)	1
1.1	BaseException	1
1.2	EngineException	1
1.3	ParseException	2
1.4	php	2

php - The general namespace of JPHP's Standart Library.

BaseException

class **BaseException**
 abstract class

extends: Exception

Children

- **class** *EngineException*
- **class** *ParseException*

EngineException

class **EngineException**
 extends: *BaseException*

Methods

EngineException::getErrorType()

Returns int

ParseException

class **ParseException**
extends: *BaseException*

php

android

Android

php\android\Android
Class Android

Methods

static startActivity (*\$class*)

Parameters

- **\$class** – string

app

Activity

php\android\app\Activity

Methods

__construct ()

setContentView (*\$view*)

Parameters

- **\$view** – *php\android\view\View*

getContentView ()

Returns *php\android\view\View*

setTitle (*\$title*)

Parameters

- **\$title** – string

findViewById (*\$id*)

Parameters

- **\$id** – int

Returns *php\android\view\View*

isChild ()

Returns bool

getParent ()

Returns *php\android\app\Activity*

onCreate ()

Returns *php\android\app\Activity*

Application

php\android\app\Application

Class Application

BootstrapActivity

php\android\app\BootstrapActivity

extends: *php\android\app\Activity*

The general launcher activity must be inherited by this class.

Class BootstrapActivity

R

php\android\R

final class

Methods

__construct ()

private

static id (\$name)

Parameters

- **\$name** – string

Returns int

static string (\$name)

Parameters

- **\$name** –

Returns int

text

InputType

php\android\text\InputType

Constants

`constant TYPE_MASK_CLASS`
`constant TYPE_MASK_VARIATION`
`constant TYPE_MASK_FLAGS`
`constant TYPE_NULL`
`constant TYPE_CLASS_TEXT`
`constant TYPE_TEXT_FLAG_CAP_CHARACTERS`
`constant TYPE_TEXT_FLAG_CAP_WORDS`
`constant TYPE_TEXT_FLAG_CAP_SENTENCES`
`constant TYPE_TEXT_FLAG_AUTO_CORRECT`
`constant TYPE_TEXT_FLAG_AUTO_COMPLETE`
`constant TYPE_TEXT_FLAG_MULTI_LINE`
`constant TYPE_TEXT_FLAG_IME_MULTI_LINE`
`constant TYPE_TEXT_FLAG_NO_SUGGESTIONS`
`constant TYPE_TEXT_VARIATION_NORMAL`
`constant TYPE_TEXT_VARIATION_URI`
`constant TYPE_TEXT_VARIATION_EMAIL_ADDRESS`
`constant TYPE_TEXT_VARIATION_EMAIL_SUBJECT`
`constant TYPE_TEXT_VARIATION_SHORT_MESSAGE`
`constant TYPE_TEXT_VARIATION_LONG_MESSAGE`
`constant TYPE_TEXT_VARIATION_PERSON_NAME`
`constant TYPE_TEXT_VARIATION_POSTAL_ADDRESS`
`constant TYPE_TEXT_VARIATION_PASSWORD`
`constant TYPE_TEXT_VARIATION_VISIBLE_PASSWORD`
`constant TYPE_TEXT_VARIATION_WEB_EDIT_TEXT`
`constant TYPE_TEXT_VARIATION_FILTER`
`constant TYPE_TEXT_VARIATION_PHONETIC`
`constant TYPE_TEXT_VARIATION_WEB_EMAIL_ADDRESS`
`constant TYPE_TEXT_VARIATION_WEB_PASSWORD`
`constant TYPE_CLASS_NUMBER`
`constant TYPE_NUMBER_FLAG_SIGNED`
`constant TYPE_NUMBER_FLAG_DECIMAL`
`constant TYPE_NUMBER_VARIATION_NORMAL`
`constant TYPE_NUMBER_VARIATION_PASSWORD`
`constant TYPE_CLASS_PHONE`


```
constant TYPE_CLASS_DATETIME
constant TYPE_DATETIME_VARIATION_NORMAL
constant TYPE_DATETIME_VARIATION_DATE
constant TYPE_DATETIME_VARIATION_TIME
```

view

View

php\android\view\View

Methods

```
__construct ($context)
```

Parameters

- **\$context** – *php\android\app\Activity*

```
on ($event, $callback)
```

Parameters

- **\$event** – string
- **\$callback** – callable

```
off ($event)
```

Parameters

- **\$event** – string

```
trigger ($event)
```

Parameters

- **\$event** – string

ViewGroup

php\android\view\ViewGroup

extends: *php\android\view\View*

Children

- class *php\android\widget\FrameLayout*
- class *php\android\widget\LinearLayout*
- class *php\android\widget\RelativeLayout*

Methods

```
addView ($view, $indexOrWidth, $height)
```

Parameters

- **\$view** – *php\android\view\View*
- **\$indexOrWidth** – int - (optional)
- **\$height** – int - (optional)

widget

Button

php\android\widget\Button

extends: *php\android\widget\TextView*

Class Button

EditText

php\android\widget\EditText

extends: *php\android\widget\TextView*

Class EditText

FrameLayout

php\android\widget\FrameLayout

extends: *php\android\view\ViewGroup*

Class FrameLayout

ImageView

php\android\widget\ImageView

extends: *php\android\view\View*

Methods

setImageAsset (*\$fileName*)

Loads image from assets.

throws *php\io\IOException*

Parameters

- **\$fileName** – string

LinearLayout

php\android\widget\LinearLayout

extends: *php\android\view\ViewGroup*

Class LinearLayout

RelativeLayout

php\android\widget\RelativeLayout

extends: *php\android\view\ViewGroup*

Class LinearLayout

TextView

php\android\widget\TextView

extends: *php\android\view\View*

Children

- class *php\android\widget\Button*
- class *php\android\widget\EditText*

Class TextView

Methods

setText (*\$text*)

Parameters

- **\$text** – string

getText ()

Returns string

setInputType (*\$type*)

Parameters

- **\$type** – int - see *phpandroidtextInputType* constants

getInputType ()

Returns int

Toast

php\android\widget\Toast

Class Toast

Constants

constant **LENGTH_SHORT**

constant **LENGTH_LONG**

Methods

show ()

cancel()

setText() (*\$text*)

Parameters

- **\$text** – string

setDuration() (*\$duration*)

Parameters

- **\$duration** – int

getDuration()

Returns int

static makeText() (*\$text*, *\$duration = ::*)

Show text

Parameters

- **\$text** – string
- **\$duration** – int

Returns *php\android\widget\Toast*

concurrent

ExecutorService

php\concurrent\ExecutorService

Class ExecutorService

Methods

__construct()

private

isScheduled()

Is Scheduled ?

Returns bool

isShutdown()

Is Shutdown?

Returns bool

isTerminated()

Returns bool

execute() (*\$runnable*, *\$env = null*)

Execute some \$runnable via the Executor Service

Parameters

- **\$runnable** – callable
- **\$env** – *php\lang\Environment*

submit (*\$runnable*, *\$env* = null)

Parameters

- **\$runnable** – callable
- **\$env** – *php\lang\Environment*

Returns *php\concurrent\Future*

schedule (*\$runnable*, *\$delay*, *\$env* = null)

Parameters

- **\$runnable** – callable
- **\$delay** – int
- **\$env** – *php\lang\Environment*

Returns *php\concurrent\Future*

shutdown ()

shutdownNow ()

awaitTermination (*\$timeout*)

Blocks until all tasks have completed execution after a shutdown request, or the timeout occurs, or the current thread is interrupted, whichever happens first.

throws *php\concurrent\Exception*

Parameters

- **\$timeout** – int - - in milliseconds

Returns bool

static newFixedThreadPool (*\$max*)

Parameters

- **\$max** – int

Returns *php\concurrent\ExecutorService*

static newCachedThreadPool

Returns *php\concurrent\ExecutorService*

static newSingleThreadExecutor

Creates an Executor that uses a single worker thread operating off an unbounded queue.

Returns *php\concurrent\ExecutorService*

static newScheduledThreadPool (*\$corePoolSize*)

Creates a thread pool that can schedule commands to run after a given delay, or to execute periodically.

Parameters

- **\$corePoolSize** – int

Returns *php\concurrent\ExecutorService*

Future

php\concurrent\Future

Class Future

Methods

__construct ()

private

isCancelled ()

Returns bool

isDone ()

Returns bool

cancel (\$mayInterruptIfRunning)

Parameters

- **\$mayInterruptIfRunning** – bool

Returns bool

get (\$timeout = null)

throws php\concurrent\Exception

Parameters

- **\$timeout** – null, int - - in milliseconds

Returns mixed

TimeoutException

php\concurrent\TimeoutException

extends: *php\lang\JavaException*

Class TimeoutException

format

JsonProcessor

php\format\JsonProcessor

extends: *php\format\Processor*

Class JsonProcessor

Constants

constant **SERIALIZE_PRETTY_PRINT**

constant **DESERIALIZE_AS_ARRAYS**

Methods

__construct (*\$flags* = 0)

Parameters

- **\$flags** – int

parse (*\$json*)

Parameters

- **\$json** – string, *php\io\Stream*

Returns mixed

format (*\$value*)

Parameters

- **\$value** – mixed

Returns string

formatTo (*\$value*, *\$output*)

Parameters

- **\$value** – mixed
- **\$output** – *php\io\Stream*

onSerialize (*\$nameOfType*, *\$handler* = null)

Parameters

- **\$nameOfType** – string - - null, int, float, string, bool, object, array
- **\$handler** – callable - (mixed \$value) -> mixed

onClassSerialize (*\$className*, *\$handler* = null)

Parameters

- **\$className** – string
- **\$handler** – callable

Processor

php\format\Processor

abstract class

Children

- class *php\format\JsonProcessor*
- class *php\xml\XmlProcessor*

Methods

format (*\$value*)

abstract

Parameters

- **\$value** –

formatTo (*\$value*, *\$output*)

abstract

Parameters

- *\$value* –
- *\$output* – *php\io\Stream*

parse (*\$string*)

abstract

Parameters

- *\$string* –

gdx

Application

php\gdx\Application

abstract class

Children

-
- class *php\gdx\LwjglApplication*

Constants

constant LOG_NONE

constant LOG_DEBUG

constant LOG_INFO

constant LOG_ERROR

Methods

getGraphics ()

Returns *php\gdx\Graphics*

getFiles ()

Returns *php\gdx\Files*

getInput ()

Returns *php\gdx\Input*

getAudio ()

Returns *php\gdx\Audio*

log (*\$tag*, *\$message*)

Parameters

- *\$tag* – string

- **\$message** – string

error (*\$tag*, *\$message*)

Parameters

- **\$tag** – string
- **\$message** – string

debug (*\$tag*, *\$message*)

Parameters

- **\$tag** – string
- **\$message** – string

setLogLevel (*\$level*)

Parameters

- **\$level** – int

getLogLevel ()

Returns int

getType ()

Returns string - Android, Desktop, HeadlessDesktop, Applet, WebGL, iOS

getVersion ()

Returns int the Android API level on Android, the major OS version on iOS (5, 6, 7, ..), or 0 on the desktop.

getJavaHeap ()

Returns int the Java heap memory use in bytes

getNativeHeap ()

Returns int the Native heap memory use in bytes

halt ()

getClipboard ()

Returns *php\gdx\Clipboard*

ApplicationListener

php\gdx\ApplicationListener

Methods

create ()

resize (*\$width*, *\$height*)

Parameters

- **\$width** –
- **\$height** –

render()
pause()
resume()
dispose()

assets

AssetManager

php\gdx\assets\AssetManager

Methods

__construct (*\$resolver*)

Parameters

- **\$resolver** – callable - (optional) - function(\$fileName): FileHandle

get (*\$fileName*)

Parameters

- **\$fileName** – string

Returns *php\gdx\graphics\Texture*, *php\gdx\graphics\Pixmap*, *php\gdx\audio\Music*,
php\gdx\audio\Sound

loadTexture (*\$fileName*)

Parameters

- **\$fileName** – string

loadPixmap (*\$fileName*)

Parameters

- **\$fileName** – string

loadMusic (*\$fileName*)

Parameters

- **\$fileName** – string

loadSound (*\$fileName*)

Parameters

- **\$fileName** – string

unload (*\$fileName*)

Parameters

- **\$fileName** – string

isLoaded (*\$fileName*)

Parameters

- **\$fileName** – string

containsAsset (*\$asset*)

Parameters

- **\$asset** – object, mixed

getAssetFileName (*\$asset*)

Parameters

- **\$asset** – object, mixed

disposeDependencies (*\$fileName*)

Parameters

- **\$fileName** – string

update (*\$millis*)

Parameters

- **\$millis** – int - (optional)

finishLoading ()

getLoadedAssets ()

Returns int

getQueuedAssets ()

Returns int

getProgress ()

Returns float

dispose ()

clear ()

getReferenceCount (*\$fileName*)

Parameters

- **\$fileName** – string

Returns int

setReferenceCount (*\$fileName*, *\$refCount*)

Parameters

- **\$fileName** – string
- **\$refCount** – int

getDiagnostics ()

Returns string

audio

AudioDevice

php\gdx\audio\AudioDevice

Class AudioDevice

Methods

__construct ()

private

isMono ()

Returns bool

writeSamples (\$samples, \$offset, \$numSamples)

Writes the array of 16-bit signed PCM samples to the audio device and blocks until they have been processed.

Parameters

- **\$samples** – array
- **\$offset** – int
- **\$numSamples** – int

writeFloatSamples (\$samples, \$offset, \$numSamples)

Writes the array of float PCM samples to the audio device and blocks until they have been processed.

Parameters

- **\$samples** – array
- **\$offset** – int
- **\$numSamples** – int

getLatency ()

Returns int

dispose ()

setVolume (\$volume)

Sets the volume in the range [0,1].

Parameters

- **\$volume** – double

AudioRecorder

php\gdx\audio\AudioRecorder

Class AudioRecorder

Methods

__construct ()

private

read (\$samples, \$offset, \$numSamples)

Reads in numSamples samples into the array samples starting at offset. If the recorder is in stereo you have to multiply numSamples by 2.

Parameters

- **\$samples** – array

- `$offset` – int
- `$numSamples` – int

`dispose()`

Music

`php\gdx\audio\Music`
Class Music

Methods

`__construct()`
`private`

`play()`

`pause()`

`stop()`

`isPlaying()`

Returns bool whether this music stream is playing

`setLooping($isLooping)`

Parameters

- `$isLooping` – bool

`isLooping()`

Returns bool

`setVolume($volume)`

Parameters

- `$volume` – double

`getVolume()`

Returns double

`setPan($pan, $volume)`

Sets the panning and volume of this music stream.

Parameters

- `$pan` – double - panning in the range -1 (full left) to 1 (full right). 0 is center position.
- `$volume` – double

`getPosition()`

Returns the playback position in milliseconds.

Returns double

`dispose()`

`setOnCompletionListener($listener)`

Register a callback to be invoked when the end of a music stream has been reached during playback.

Parameters

- **\$listener** – callable - (Music \$music)

Sound**php\gdx\audio\Sound**

Class Sound

Methods

__construct ()**private****play** (*\$volume*, *\$pitch*, *\$pan*)

Plays the sound. If the sound is already playing, it will be played again, concurrently.

Parameters

- **\$volume** – double - (optional)
- **\$pitch** – double - (optional)
- **\$pan** – double - (optional)

Returns int the id of the sound instance if successful, or -1 on failure.**loop** (*\$volume*, *\$pitch*, *\$pan*)

Plays the sound, looping. If the sound is already playing, it will be played again, concurrently. You need to stop the sound via a call to stop(long) using the returned id.

Parameters

- **\$volume** – double - (optional)
- **\$pitch** – double - (optional)
- **\$pan** – double - (optional)

Returns int the id of the sound instance if successful, or -1 on failure.**stop** (*\$soundId*)

Stops playing all or \$soundId instance(s) of this sound.

Parameters

- **\$soundId** – int - (optional)

pause (*\$soundId*)

Pauses the sound instance with the given id as returned by play() or all sounds. If the sound is no longer playing, this has no effect.

Parameters

- **\$soundId** – int - (optional)

resume (*\$soundId*)**Parameters**

- **\$soundId** – int - (optional)

setLooping (*\$soundId*, *\$looping*)

Sets the sound instance with the given id to be looping. If the sound is no longer playing this has no effect

Parameters

- **\$soundId** – int
- **\$looping** – bool

setPitch (*\$soundId*, *\$pitch*)

Changes the pitch multiplier of the sound instance with the given id as returned by play(). If the sound is no longer playing, this has no effect.

Parameters

- **\$soundId** – int
- **\$pitch** – float

setPan (*\$soundId*, *\$pan*, *\$volume*)

Sets the panning and volume of the sound instance with the given id as returned by play(). If the sound is no longer playing, this has no effect.

Parameters

- **\$soundId** – int
- **\$pan** – float
- **\$volume** – float

setPriority (*\$soundId*, *\$priority*)

Sets the priority of a sound currently being played back. Higher priority sounds will be considered last if the maximum number of concurrently playing sounds is exceeded. This is only a hint and might not be honored by a backend implementation.

Parameters

- **\$soundId** – int
- **\$priority** – int - the priority (0 == lowest)

dispose ()

Audio

php\gdx\Audio

Class Audio

Methods

__construct ()

private

newAudioDevice (*\$samplingRate*, *\$isMono*)

Creates a new AudioDevice either in mono or stereo mode. The AudioDevice has to be disposed via its AudioDevice->dispose() method when it is no longer used.

throws *php\gdx\GdxRuntimeException* in case the device could not be created

Parameters

- **\$samplingRate** – int
- **\$isMono** – boolean

Returns *php\gdx\audio\AudioDevice*

newAudioRecorder (*\$samplingRate*, *\$isMono*)

Creates a new AudioRecorder. The AudioRecorder has to be disposed after it is no longer used.

throws *php\gdx\GdxRuntimeException* in case the recorder could not be created

Parameters

- **\$samplingRate** – int
- **\$isMono** – bool

Returns *php\gdx\audio\AudioRecorder*

newSound (*\$fileHandle*)

Creates a new Sound which is used to play back audio effects such as gun shots or explosions. The Sound's audio data is retrieved from the file specified via the FileHandle. Note that the complete audio data is loaded into RAM. You should therefore not load big audio files with this methods. The current upper limit for decoded audio is 1 MB.

Currently supported formats are WAV, MP3 and OGG.

The Sound has to be disposed if it is no longer used via the { @link Sound#dispose() } method.

throws *php\gdx\GdxRuntimeException* in case the sound could not be loaded

Parameters

- **\$fileHandle** – *php\gdx\files\FileHandle*

Returns *php\gdx\audio\Sound*

newMusic (*\$fileHandle*)

Creates a new Music instance which is used to play back a music stream from a file. Currently supported formats are WAV, MP3 and OGG. The Music instance has to be disposed if it is no longer used via the Music->dispose() method. Music instances are automatically paused when ApplicationListener->pause() is called and resumed when ApplicationListener->resume() is called.

throws *php\gdx\GdxRuntimeException* in case the music could not be loaded

Parameters

- **\$fileHandle** – *php\gdx\files\FileHandle*

Returns *php\gdx\audio\Music*

Clipboard

php\gdx\Clipboard

Class Clipboard

Methods

__construct ()
private

getContent ()
gets the current content of the clipboard if it contains text

Returns string the clipboard content or null

setContent (*\$content*)

Sets the content of the system clipboard.

Parameters

- **\$content** – string

files

FileHandle

php\gdx\files\FileHandle

Class FileHandle

Methods

__construct (*\$path*)

Parameters

- **\$path** – string, *php\io\File*

path ()

Returns string

name ()

Returns string

extension ()

Returns string

nameWithoutExtension ()

Returns string

pathWithoutExtension ()

Returns string

type ()

Returns string

file ()

Returns *php\io\File*

read ()

Returns *php\io\Stream*

readString (*\$charset*)

Parameters

- **\$charset** – string - (optional)

Returns string

readBytes ()

Returns string binary

write (*\$append*, *\$bufferSize*)

Parameters

- **\$append** – bool
- **\$bufferSize** – int - (optional)

Returns *php\io\Stream*

writeString (*\$string*, *\$append*, *\$charset*)

Parameters

- **\$string** – string
- **\$append** – bool
- **\$charset** – string - (optional)

writeBytes (*\$binaryString*, *\$append*)

Parameters

- **\$binaryString** – string
- **\$append** – bool

getList (*\$suffix*)

Parameters

- **\$suffix** – string - (optional)

Returns *php\gdx\files\FileHandle[]*

isDirectory ()

Returns bool

child (*\$name*)

Parameters

- **\$name** – string

Returns *php\gdx\files\FileHandle*

sibling (*\$name*)

Parameters

- **\$name** –

Returns *php\gdx\files\FileHandle*

parent ()

Returns *php\gdx\files\FileHandle*

makedirs ()

throws *php\gdx\files\Exception*

exists ()

Returns bool

delete ()

Returns bool

deleteDirectory ()

Returns bool

emptyDirectory (*\$preserveTree*)

Parameters

- **\$preserveTree** – bool - (optional)

copyTo (*\$fileHandle*)

Parameters

- **\$fileHandle** – *php\gdx\files\FileHandle*

moveTo (*\$fileHandle*)

Parameters

- **\$fileHandle** – *php\gdx\files\FileHandle*

length ()

Returns int

lastModified ()

Returns int

static tempFile (*\$suffix*)

Parameters

- **\$suffix** – string

Returns *php\gdx\files\FileHandle*

static tempDirectory (*\$suffix*)

Parameters

- **\$suffix** – string

Returns *php\gdx\files\FileHandle*

Files

php\gdx\Files

Class Files

Methods

getFileHandle (*\$path*, *\$type*)

Parameters

- **\$path** – string
- **\$type** – string - - Classpath, Internal, External, Absolute, Local

Returns *php\gdx\files\FileHandle*

classpath (*\$path*)

Parameters

- **\$path** – string

Returns *php\gdx\files\FileHandle*

internal (*\$path*)

Parameters

- **\$path** –

Returns *php\gdx\files\FileHandle*

external (*\$path*)

Parameters

- **\$path** –

Returns *php\gdx\files\FileHandle*

absolute (*\$path*)

Parameters

- **\$path** –

Returns *php\gdx\files\FileHandle*

local (*\$path*)

Parameters

- **\$path** –

Returns *php\gdx\files\FileHandle*

getExternalStoragePath ()

Returns string

isExternalStorageAvailable ()

Returns bool

getLocalStoragePath ()

Returns string

isLocalStorageAvailable ()

Returns string

Gdx

php\gdx\Gdx

Methods

static app

Returns *php\gdx\Application*

static files

Returns *php\gdx\Files*

static graphics

Returns *php\gdx\Graphics*

static input

Returns *php\gdx\Input*

static audio

Returns *php\gdx\Audio*

GdxRuntimeException

php\gdx\GdxRuntimeException

graphics

Batch

php\gdx\graphics\Batch

Methods

begin()

end()

DisplayMode

php\gdx\graphics\DisplayMode

Class DisplayMode

Methods

__construct()
 private

getWidth()
 Returns int

getHeight()
 Returns int

getBitsPerPixel()
 Returns int

getRefreshRate()
 Returns int

__toString()
 Returns string

Pixmap

php\gdx\graphics\Pixmap

Class Pixmap

Methods

__construct (*\$width*, *\$height*, *\$format*)

Parameters

- **\$width** – int
- **\$height** – int
- **\$format** – string - - Alpha, Intensity, LuminanceAlpha, RGB565, RGBA4444, RGB888, RGBA8888

static ofFile (*\$fileHandle*)

Parameters

- **\$fileHandle** – *php\gdx\files\FileHandle*

Returns string

setColor (*\$redOrColor*, *\$g*, *\$b*, *\$a*)

Parameters

- **\$redOrColor** – double, int
- **\$g** – double - (optional)
- **\$b** – double - (optional)
- **\$a** – double - (optional)

fill ()

drawLine (*\$x*, *\$y*, *\$x2*, *\$y2*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$x2** – int
- **\$y2** – int

drawRectangle (*\$x*, *\$y*, *\$width*, *\$height*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

drawPixmap (*\$pixmap*, *\$x*, *\$y*, *\$srcx*, *\$srcy*, *\$srcWidth*, *\$srcHeight*)

Parameters

- **\$pixmap** – *php\gd\graphics\Pixmap*
- **\$x** – int
- **\$y** – int
- **\$srcx** – int - (optional)
- **\$srcy** – int - (optional)
- **\$srcWidth** – int - (optional)
- **\$srcHeight** – int - (optional)

fillRectangle (*\$x*, *\$y*, *\$width*, *\$height*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

drawCircle (*\$x*, *\$y*, *\$radius*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$radius** – int

fillCircle (*\$x*, *\$y*, *\$radius*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$radius** – int

fillTriangle (*\$x1*, *\$y1*, *\$x2*, *\$y2*, *\$x3*, *\$y3*)

Parameters

- **\$x1** – int
- **\$y1** – int
- **\$x2** – int
- **\$y2** – int
- **\$x3** – int
- **\$y3** – int

getPixel (*\$x*, *\$y*)

Parameters

- **\$x** – int
- **\$y** – int

Returns int The pixel color in RGBA8888 format.

getWidth()

Returns int

getHeight()

Returns int

dispose()

drawPixel (\$x, \$y, \$color)

Parameters

- **\$x** – int
- **\$y** – int
- **\$color** – int - (optional)

getGLFormat()

Returns int one of GL_ALPHA, GL_RGB, GL_RGBA, GL_LUMINANCE, or GL_LUMINANCE_ALPHA.

getGLInternalFormat()

Returns int one of GL_ALPHA, GL_RGB, GL_RGBA, GL_LUMINANCE, or GL_LUMINANCE_ALPHA.

getGLType()

Returns int one of GL_UNSIGNED_BYTE, GL_UNSIGNED_SHORT_5_6_5, GL_UNSIGNED_SHORT_4_4_4_4

getFormat()

Returns string

getBlending()

Returns string None, SourceOver

static setBlending (\$blending)

Sets the type of Blending to be used for all operations. Default is 'SourceOver'

Parameters

- **\$blending** – string

static setFilter (\$filter)

Filters to be used with Pixmap.drawPixmap(Pixmap, int, int, int, int, int, int, int, int).

Parameters

- **\$filter** – string - NearestNeighbour, BiLinear

Sprite

php\gdx\graphics\Sprite

Class Sprite

Methods

__construct (\$texture, \$width, \$height, \$x, \$y)

Parameters

- **\$texture** – *php\gdx\graphics\Texture* - (optional)
- **\$width** – int - (optional)
- **\$height** – int - (optional)
- **\$x** – int - (optional)
- **\$y** – int - (optional)

setBounds (*\$x*, *\$y*, *\$width*, *\$height*)**Parameters**

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

setSize (*\$width*, *\$height*)**Parameters**

- **\$width** – int
- **\$height** – int

setPosition (*\$x*, *\$y*)**Parameters**

- **\$x** – int
- **\$y** – int

setX (*\$x*)**Parameters**

- **\$x** – int

getX ()**Returns** int**setY** (*\$y*)**Parameters**

- **\$y** – int

getY ()**Returns** int**translateX** (*\$xAmount*)**Parameters**

- **\$xAmount** – double

translateY (*\$yAmount*)**Parameters**

- **\$yAmount** – double

translate (*\$xAmount*, *\$yAmount*)

Parameters

- **\$xAmount** – double
- **\$yAmount** – double

setAlpha (*\$alpha*)

Parameters

- **\$alpha** – double

setOrigin (*\$originX*, *\$originY*)

Parameters

- **\$originX** – int
- **\$originY** – int

setOriginCenter ()

setRotation (*\$degrees*)

Parameters

- **\$degrees** – double

getRotation ()

Returns double

rotate (*\$degrees*)

Parameters

- **\$degrees** – double

rotate90 (*\$clockwise*)

Parameters

- **\$clockwise** – bool

setScale (*\$scaleXY*, *\$scaleY*)

Parameters

- **\$scaleXY** – double
- **\$scaleY** – double - (optional)

scale (*\$amount*)

Parameters

- **\$amount** – double

flip (*\$x*, *\$y*)

Parameters

- **\$x** – int
- **\$y** – int

scroll (*\$xAmount*, *\$yAmount*)

Parameters

- **\$xAmount** – double
- **\$yAmount** – double

draw (*\$batch*, *\$alphaModulation*)

Parameters

- **\$batch** – *php\gdx\graphics\SpriteBatch*
- **\$alphaModulation** – double - (optional)

SpriteBatch

php\gdx\graphics\SpriteBatch

extends: *php\gdx\graphics\Batch*

Texture

php\gdx\graphics\Texture

Class Texture

Methods

static ofFile (*\$fileHandle*, *\$useMipMaps* = *false*, *\$format*)

Parameters

- **\$fileHandle** – *php\gdx\files\FileHandle*
- **\$useMipMaps** – bool - (optional)
- **\$format** – string - (optional)

Returns *php\gdx\graphics\Texture*

__construct (*\$pixmap*, *\$useMipMaps* = *false*, *\$format*)

Parameters

- **\$pixmap** – *php\gdx\graphics\Pixmap*
- **\$useMipMaps** – bool - (optional)
- **\$format** – string - (optional)

getWidth ()

Returns int

getHeight ()

Returns int

getDepth ()

Returns int

isManaged ()

Returns bool

Graphics

php\gdx\Graphics

Class Graphics

Methods

isGL30Available()

Returns bool

getWidth()

Returns int

getHeight()

Returns int

getDeltaTime()

Returns double

getRawDeltaTime()

Returns double

getFramesPerSecond()

Returns int

getDensity()

Returns float

supportsDisplayModeChange()

Returns bool

getDisplayModes()

Returns php\gdx\DisplayMode[] the supported fullscreen DisplayMode(s)

getDesktopDisplayMode()

Returns *php\gdx\graphics\DisplayMode*

setDisplayMode(\$widthOrDisplayMode, \$height, \$fullscreen)

Parameters

- **\$widthOrDisplayMode** – *php\gdx\graphics\DisplayMode*, int
- **\$height** – int - (optional)
- **\$fullscreen** – int - (optional)

setTitle(\$title)

Parameters

- **\$title** – string

setVSync(\$vsync)

Parameters

- **\$vsync** – bool

supportsExtension (*\$extension*)

Parameters

- **\$extension** – string

Returns bool

setContinuousRendering (*\$value*)

Parameters

- **\$value** – bool

isContinuousRendering ()

Returns bool

requestRendering ()

isFullscreen ()

Returns bool

Input

php\gdx\Input
final class

Methods

__construct ()
private

getAccelerometerX ()

Returns double The value of the accelerometer on its x-axis. ranges between [-10,10].

getAccelerometerY ()

Returns double The value of the accelerometer on its y-axis. ranges between [-10,10].

getAccelerometerZ ()

Returns double The value of the accelerometer on its y-axis. ranges between [-10,10].

getX (*\$pointer*)

Returns the x coordinate of the last touch on touch screen devices and the current mouse position on desktop for the first pointer in screen coordinates. The screen origin is the top left corner.

Parameters

- **\$pointer** – int - (optional) the pointer id. Returns the x coordinate in screen coordinates of the given pointer.

Returns int

getDeltaX (*\$pointer*)

Parameters

- **\$pointer** – int - (optional) the pointer id.

Returns int the different between the current pointer location and the last pointer location on the x-axis.

getY (*\$pointer*)

Returns the y coordinate of the last touch on touch screen devices and the current mouse position on desktop for the first pointer in screen coordinates. The screen origin is the top left corner.

Parameters

- **\$pointer** – int - (optional) the pointer id. Returns the y coordinate in screen coordinates of the given pointer.

Returns int**getDeltaY** (*\$pointer*)**Parameters**

- **\$pointer** – int - (optional) the pointer id.

Returns int the different between the current pointer location and the last pointer location on the y-axis.

isTouched (*\$pointer*)**Parameters**

- **\$pointer** – int - (optional)

Returns bool whether the screen is currently touched.

justTouched ()

Returns bool whether a new touch down event just occurred.

isButtonPressed (*\$button*)

Whether a given button is pressed or not. Button constants can be found in {[@link Buttons](#)}. On Android only the Button#LEFT constant is meaningful.

Parameters

- **\$button** – int

Returns bool**isKeyPressed** (*\$key*)

Returns whether the key is pressed.

Parameters

- **\$key** – int

setOnscreenKeyboardVisible (*\$visible*)

Sets the on-screen keyboard visible if available.

Parameters

- **\$visible** – bool

vibrate (*\$millis*)

Vibrates for the given amount of time. Note that you'll need the permission `<uses-permission android:name="android.permission.VIBRATE" />` in your manifest file in order for this to work.

Parameters

- **\$millis** – int - the number of milliseconds to vibrate.

cancelVibrate ()

getAzimuth()

The azimuth is the angle of the device's orientation around the z-axis. The positive z-axis points towards the earth's center.

Returns double

getPitch()

The pitch is the angle of the device's orientation around the x-axis. The positive x-axis roughly points to the west and is orthogonal to the z- and y-axis.

Returns double

getRoll()

The roll is the angle of the device's orientation around the y-axis. The positive y-axis points to the magnetic north pole of the earth.

Returns double

getCurrentEventTime()

Returns int the time of the event currently reported to the InputProcessor.

setCatchBackKey() (*\$catchBack*)

Sets whether the BACK button on Android should be caught. This will prevent the app from being paused. Will have no effect on the desktop.

Parameters

- **\$catchBack** – bool

setCatchMenuKey() (*\$catchMenu*)

Sets whether the MENU button on Android should be caught. This will prevent the onscreen keyboard to show up. Will have no effect on the desktop.

Parameters

- **\$catchMenu** – bool

getRotation()

Returns int the rotation of the device with respect to its native orientation.

getNativeOrientation()

Returns string the native orientation of the device.

setCursorCaught() (*\$caught*)

Only viable on the desktop. Will confine the mouse cursor location to the window and hide the mouse cursor.

Parameters

- **\$caught** – bool - whether to catch or not to catch the mouse cursor

isCursorCaught()

Returns bool whether the mouse cursor is caught.

setCursorPosition() (*\$x*, *\$y*)

Only viable on the desktop. Will set the mouse cursor location to the given window coordinates (origin top-left corner).

Parameters

- **\$x** – int
- **\$y** – int

setCursorImage (*\$pixmap, \$xHotspot, \$yHotspot*)

Only viable on the desktop. Will set the mouse cursor image to the image represented by the Pixmap. The Pixmap must be in RGBA8888 format, width & height must be powers-of-two greater than zero (not necessarily equal), and alpha transparency must be single-bit (i.e., 0x00 or 0xFF only). To revert to the default operating system cursor, pass in a null Pixmap; xHotspot & yHotspot are ignored in this case.

Parameters

- **\$pixmap** – *php\gdx\graphics\Pixmap*
- **\$xHotspot** – int
- **\$yHotspot** – int

LwjglApplication

php\gdx\LwjglApplication

extends: *php\gdx\Application*

Methods

__construct (*\$listener, \$configuration*)

Parameters

- **\$listener** – *php\gdx\ApplicationListener*
- **\$configuration** – *php\gdx\LwjglApplicationConfiguration*

LwjglApplicationConfiguration

php\gdx\LwjglApplicationConfiguration

Class ApplicationConfiguration

Properties

property useGL30

bool

property depth

int

property samples

int

property width

int

property height

int

property x

int

property y

:doc:‘ </api_en/.types/>‘

property fullscreen
bool

property vsyncEnabled
bool

property title
string

property forceExit
bool

property resizable
bool

property audioDeviceSimultaneousSources
int

property audioDeviceBufferSize
int

property audioDeviceBufferCount
int

property foregroundFPS
int

property backgroundFPS
int

property allowSoftwareMode
bool

property preferencesDirectory
string

Methods

setFromDisplayMode (*\$displayMode*)

Parameters

- **\$displayMode** – *php\gdx\graphics\DisplayMode*

static getDesktopDisplayMode

Returns *php\gdx\graphics\DisplayMode*

static getDisplayModes

Returns *php\gdx\DisplayMode[]*

static addIcon (*\$path*, *\$type*)

Adds a window icon. Icons are tried in the order added, the first one that works is used. Typically three icons should be provided: 128x128 (for Mac), 32x32 (for Windows and Linux), and 16x16 (for Windows).

Parameters

- **\$path** – string
- **\$type** – string - - Classpath, Internal, External, Absolute, Local

math

Polygon

php\gdx\math\Polygon

Class Polygon

Methods

__construct (*\$vertices*)

Parameters

- **\$vertices** – array - (optional)

getVertices ()

Returns the polygon's local vertices without scaling or rotation and without being offset by the polygon position.

Returns float[]

getTransformedVertices ()

Returns float[] vertices scaled, rotated, and offset by the polygon position.

setOrigin (*\$originX*, *\$originY*)

Sets the origin point to which all of the polygon's local vertices are relative to.

Parameters

- **\$originX** – float
- **\$originY** – float

setPosition (*\$x*, *\$y*)

Sets the polygon's position within the world.

Parameters

- **\$x** – float
- **\$y** – float

setVertices (*\$vertices*)

Parameters

- **\$vertices** – array

translate (*\$x*, *\$y*)

Translates the polygon's position by the specified horizontal and vertical amounts.

Parameters

- **\$x** – float
- **\$y** – float

setRotation (*\$degrees*)

Sets the polygon to be rotated by the supplied degrees.

Parameters

- **\$degrees** – float

rotate (*\$degrees*)

Applies additional rotation to the polygon by the supplied degrees.

Parameters

- **\$degrees** – float

setScale (*\$scaleX*, *\$scaleY*)

Sets the amount of scaling to be applied to the polygon.

Parameters

- **\$scaleX** – float
- **\$scaleY** – float

scale (*\$amount*)

Applies additional scaling to the polygon by the supplied amount.

Parameters

- **\$amount** – float

dirty ()

area ()

Returns the area contained within the polygon.

Returns float

contains (*\$x*, *\$y*)

Parameters

- **\$x** – float
- **\$y** – float

Returns bool

getX ()

Returns float

getY ()

Returns float

getOriginX ()

Returns float

getOriginY ()

Returns float

getRotation ()

Returns float

getScaleX ()

Returns float

getScaleY ()

Returns float

Polyline

`php\gdx\math\Polyline`

Class Polyline

Methods

__construct (*\$vertices*)

Parameters

- **\$vertices** – array - (optional)

getVertices ()

Returns the polyline's local vertices without scaling or rotation and without being offset by the polyline position.

Returns float[]

getTransformedVertices ()

Returns float[] vertices scaled, rotated, and offset by the polyline position.

getLength ()

Returns float Returns the euclidean length of the polyline without scaling

getScaledLength ()

Returns float Returns the euclidean length of the polyline

calculateLength ()

calculateScaledLength ()

dirty ()

setOrigin (*\$originX*, *\$originY*)

Sets the origin point to which all of the polyline's local vertices are relative to.

Parameters

- **\$originX** – float
- **\$originY** – float

setPosition (*\$x*, *\$y*)

Sets the polyline's position within the world.

Parameters

- **\$x** – float
- **\$y** – float

setVertices (*\$vertices*)

Parameters

- **\$vertices** – array

translate (*\$x*, *\$y*)

Translates the polyline's position by the specified horizontal and vertical amounts.

Parameters

- **\$x** – float

- **\$y** – float

setRotation (*\$degrees*)

Sets the polyline to be rotated by the supplied degrees.

Parameters

- **\$degrees** – float

rotate (*\$degrees*)

Applies additional rotation to the polyline by the supplied degrees.

Parameters

- **\$degrees** – float

setScale (*\$scaleX*, *\$scaleY*)

Sets the amount of scaling to be applied to the polyline.

Parameters

- **\$scaleX** – float
- **\$scaleY** – float

scale (*\$amount*)

Applies additional scaling to the polyline by the supplied amount.

Parameters

- **\$amount** – float

getX ()

Returns float

getY ()

Returns float

getOriginX ()

Returns float

getOriginY ()

Returns float

getRotation ()

Returns float

getScaleX ()

Returns float

getScaleY ()

Returns float

Vector2

php\gdx\math\Vector2

Class Vector2

Methods

__construct (\$x, \$y)

Parameters

- **\$x** – float, *php\gdx\math\Vector2* - (optional)
- **\$y** – float - (optional)

x ()

Returns float

y ()

Returns float

len ()

Returns float The euclidean length

len2 ()

Returns float The squared euclidean length

set (\$x, \$y)

Sets this vector from the given vector or x, y

Parameters

- **\$x** – float, *php\gdx\math\Vector2*
- **\$y** – float - (optional)

Returns *php\gdx\math\Vector2*

sub (\$x, \$y)

Subtracts the given vector from this vector.

Parameters

- **\$x** – float, *php\gdx\math\Vector2*
- **\$y** – float - (optional)

Returns *php\gdx\math\Vector2*

nor ()

Normalizes this vector. Does nothing if it is zero.

Returns *php\gdx\math\Vector2*

add (\$x, \$y)

Adds the given vector to this vector

Parameters

- **\$x** – float, *php\gdx\math\Vector2*
- **\$y** – float - (optional)

dot (\$ox, \$oy)

Parameters

- **\$ox** – float, *php\gdx\math\Vector2*

- **\$oy** – float - (optional)

Returns float The dot product between this and the other vector

scl (\$x, \$y)

Scales this vector by a scalar

Parameters

- **\$x** – float
- **\$y** – float - (optional)

Returns *php\gdx\math\Vector2*

mulAdd (\$vec, \$scalar)

First scale a supplied vector, then add it to this vector.

Parameters

- **\$vec** – *php\gdx\math\Vector2*
- **\$scalar** – float

Returns *php\gdx\math\Vector2*

dst (\$x, \$y)

Parameters

- **\$x** – float, *php\gdx\math\Vector2*
- **\$y** – float - (optional)

Returns float the distance between this and the other vector

dst2 (\$x, \$y)

Parameters

- **\$x** – float, *php\gdx\math\Vector2*
- **\$y** – float - (optional)

Returns float the squared distance between this and the other vector

limit (\$limit)

Parameters

- **\$limit** – float

Returns *php\gdx\math\Vector2*

clamp (\$min, \$max)

Parameters

- **\$min** – float
- **\$max** – float

Returns *php\gdx\math\Vector2*

crs (\$x, \$y)

Parameters

- **\$x** – float, *php\gdx\math\Vector2*
- **\$y** – float - (optional)

Returns float

angle()

Returns float

getAngleRad()

Returns float

setAngle(\$degrees)

Parameters

- **\$degrees** – float

Returns *php\gdx\math\Vector2*

setAngleRad(\$radians)

Parameters

- **\$radians** – float

Returns *php\gdx\math\Vector2*

rotate(\$degrees)

Parameters

- **\$degrees** – float

Returns *php\gdx\math\Vector2*

rotateRad(\$radians)

Parameters

- **\$radians** – float

Returns *php\gdx\math\Vector2*

rotate90(\$dir)

Parameters

- **\$dir** – int

Returns *php\gdx\math\Vector2*

lerp(\$target, \$alpha)

Linearly interpolates between this vector and the target vector by alpha which is in the range [0,1].
The result is stored in this vector.

Parameters

- **\$target** – *php\gdx\math\Vector2*
- **\$alpha** – float

Returns *php\gdx\math\Vector2*

epsilonEquals(\$other, \$epsilon)

Compares this vector with the other vector, using the supplied epsilon for fuzzy equality testing.

Parameters

- **\$other** – *php\gdx\math\Vector2*
- **\$epsilon** – float

Returns bool

isUnit (*\$margin*)

Whether this vector is a unit length vector

Parameters

- **\$margin** – float - (optional)

Returns bool

isZero (*\$margin*)

Parameters

- **\$margin** – float - (optional)

isOnLine (*\$other*, *\$epsilon*)

Parameters

- **\$other** – *php\gd\math\Vector2*
- **\$epsilon** – float - (optional)

Returns bool

isCollinear (*\$other*, *\$epsilon*)

Parameters

- **\$other** – *php\gd\math\Vector2*
- **\$epsilon** – float - (optional)

Returns bool

isCollinearOpposite (*\$other*, *\$epsilon*)

Parameters

- **\$other** – *php\gd\math\Vector2*
- **\$epsilon** – float - (optional)

Returns bool

isPerpendicular (*\$vector*, *\$epsilon*)

Parameters

- **\$vector** – *php\gd\math\Vector2*
- **\$epsilon** – float - (optional)

hasSameDirection (*\$vector*)

Parameters

- **\$vector** – *php\gd\math\Vector2*

Returns bool

hasOppositeDirection (*\$vector*)

Parameters

- **\$vector** – *php\gd\math\Vector2*

Returns bool

io

File

php\io\File

Class File

Constants

constant `PATH_SEPARATOR`

constant `DIRECTORY_SEPARATOR`

constant `PATH_NAME_CASE_INSENSITIVE`

Methods

__construct (*\$path*, *\$child* = *NULL*)

Parameters

- **\$path** – string
- **\$child** – null, string

exists ()

Returns bool

canExecute ()

Returns bool

canWrite ()

Returns bool

canRead ()

Returns bool

getName ()

Returns string

getAbsolutePath ()

Returns string

getCanonicalPath ()

throws *php\io\IOException*

Returns string

getParent ()

Returns string

getPath ()

Returns string

getAbsolutePath ()

Returns *php\io\File*

getCanonicalFile()
throws *php\io\IOException*
Returns *php\io\File*

getParentFile()
Returns *php\io\File*

mkdir()
Returns bool

mkdirs()
Returns bool

isFile()
Returns bool

isDirectory()
Returns bool

isAbsolute()
Returns bool

isHidden()
Returns bool

delete()
Returns bool

deleteOnExit()
Returns void

createNewFile(\$withDirs = false)
Parameters
• **\$withDirs** – bool
Returns bool

lastModified()
Returns int

length()
Returns int

renameTo(\$newName)
Parameters
• **\$newName** – string
Returns bool

setExecutable(\$value, \$ownerOnly = true)
Parameters
• **\$value** – bool

- **\$ownerOnly** – bool

Returns bool

setWritable (*\$value*, *\$ownerOnly* = *true*)

Parameters

- **\$value** – bool
- **\$ownerOnly** – bool

Returns bool

setReadable (*\$value*, *\$ownerOnly* = *true*)

Parameters

- **\$value** – bool
- **\$ownerOnly** – bool

Returns bool

setReadOnly ()

Returns bool

setLastModified (*\$time*)

Parameters

- **\$time** – int

Returns bool

compareTo (*\$file*)

Parameters

- **\$file** – string, *php\io\File*

Returns int

find (*\$filter* = *null*)
throws *php\io\IOException*

Parameters

- **\$filter** – callable

Returns string[]

findFiles (*\$filter* = *null*)
throws *php\io\IOException*

Parameters

- **\$filter** – callable - (File \$directory, \$name)

Returns *php\io\File*[]

static createTemp (*\$prefix*, *\$suffix*, *\$directory* = *null*)

Parameters

- **\$prefix** – string
- **\$suffix** – string
- **\$directory** – null, *php\io\File*, string

Returns *php\io\File*

static listRoots

List the available filesystem roots. Returns an array of objects denoting the available filesystem roots, or empty array if the set of roots could not be determined. The array will be empty if there are no filesystem roots.

Returns *php\io\File[]*

static of (\$path)

Parameters

- **\$path** – string

Returns *php\io\File*

FileStream

php\io\FileStream

extends: *php\io\Stream*

Methods

read (\$length)

throws *php\io\IOException*

Parameters

- **\$length** – int - - count of bytes

Returns mixed

readFully ()

throws *php\io\IOException*

Returns mixed

write (\$value, \$length = null)

throws *php\io\IOException*

Parameters

- **\$value** – string
- **\$length** – null, int

Returns int

eof ()

Returns bool

seek (\$position)

throws *php\io\IOException*

Parameters

- **\$position** – int

Returns mixed

getPosition ()

throws *php\io\IOException*

Returns int

close()

Returns mixed

length()

Returns int

getFilePointer()

throws *php\io\IOException*

Returns int

truncate(\$size)

throws *php\io\IOException*

Parameters

- **\$size** – int

IOException

php\io\IOException

extends: *php\lang\JavaException*

Class IOException

MemoryStream

php\io\MemoryStream

extends: *php\io\MiscStream*

Class MemoryStream

Methods

__construct()

MiscStream

php\io\MiscStream

extends: *php\io\Stream*

Children

- **class** *php\io\MemoryStream*

Class MiscStream

Methods

read(\$length)

throws *php\io\IOException*

Parameters

- **\$length** – int - - count of bytes

Returns mixed

readFully()

throws *php\io\IOException*

Returns mixed

write(\$value, \$length = null)

throws *php\io\IOException*

Parameters

- **\$value** – string
- **\$length** – null, int

Returns int

eof()

Returns bool

seek(\$position)

throws *php\io\IOException*

Parameters

- **\$position** – int

Returns mixed

getPosition()

throws *php\io\IOException*

Returns int

close()

Returns mixed

length()

Returns int

flush()

throws *php\io\IOException*

ResourceStream

php\io\ResourceStream

extends: *php\io\Stream*

Class ResourceStream

Methods

__construct(\$path)

Parameters

- **\$path** – string

read (*\$length*)
throws *php\io\IOException*

Parameters

- **\$length** – int - - count of bytes

Returns mixed

readFully ()
throws *php\io\IOException*

Returns mixed

write (*\$value*, *\$length = null*)
throws *php\io\IOException*

Parameters

- **\$value** – string
- **\$length** – null, int

Returns int

eof ()

Returns bool

seek (*\$position*)
throws *php\io\IOException*

Parameters

- **\$position** – int

Returns mixed

getPosition ()
throws *php\io\IOException*

Returns int

close ()

Returns mixed

static getResources (*\$name*)

Parameters

- **\$name** – string

Returns *php\io\ResourceStream[]*

Stream

php\io\Stream
abstract class

Children

- class *php\io\FileStream*
- class *php\io\MiscStream*

- class *php\io\ResourceStream*
- class *php\net\NetStream*

Properties

property path

string

private

property mode

string

private

Methods

getPath ()

Returns string

getMode ()

Returns string

read (\$length)

abstract

throws *php\io\IOException*

Parameters

- **\$length** – int - - count of bytes

Returns mixed

readFully ()

abstract

throws *php\io\IOException*

Returns mixed

write (\$value, \$length = null)

abstract

throws *php\io\IOException*

Parameters

- **\$value** – string
- **\$length** – null, int

Returns int

eof ()

abstract

Returns bool

seek (\$position)

abstract

throws *php\io\IOException*

Parameters

- **\$position** – int

Returns mixed**getPosition()****abstract****throws** *php\io\IOException***Returns** int**close()****abstract****Returns** mixed**__construct** (*\$path*, *\$mode* = null)**Parameters**

- **\$path** – string
- **\$mode** – null, string

Returns *php\io\Stream***setContext** (*\$context*)**Parameters**

- **\$context** –

Returns void**getContext()****Returns** mixed**static of** (*\$path*, *\$mode* = 'r')**throws** *php\io\IOException***Parameters**

- **\$path** – string
- **\$mode** – string

Returns *php\io\Stream***static getContents** (*\$path*, *\$mode* = 'r')

Create a stream, call and return the result of the readFully() method, finally call the close() method.

throws *php\io\IOException***Parameters**

- **\$path** – string
- **\$mode** – string

Returns string binary**static putContents** (*\$path*, *\$data*, *\$mode* = 'w+')

Create a stream, call and return the result of the write() method, finally call the close() method.

throws *php\io\IOException***Parameters**

- **\$path** – string
- **\$data** – string
- **\$mode** – string

static exists (*\$path*)

Checks stream is exists. It tries to open a stream and if all is ok, returns true and closes it.

throws `php\io\Exception` if you check external streams like http or ftp

Parameters

- **\$path** – string

Returns bool

static register (*\$protocol*, *\$className*)

Parameters

- **\$protocol** – string
- **\$className** – string

static unregister (*\$protocol*)

Parameters

- **\$protocol** –

jsoup

Connection

`php\jsoup\Connection`
abstract class

Constants

constant `METHOD_POST`

constant `METHOD_GET`

Methods

data (*\$data*)

Parameters

- **\$data** – array

Returns *php\jsoup\Connection*

cookies (*\$data*)

Parameters

- **\$data** – array

Returns *php\jsoup\Connection*

headers (*\$data*)

Parameters

- **\$data** – array

Returns *php\jsoup\Connection***url** (*\$url*)**Parameters**

- **\$url** – string

Returns *php\jsoup\Connection***method** (*\$method*)**Parameters**

- **\$method** – string - POST or GET

Returns *php\jsoup\Connection***userAgent** (*\$userAgent*)**Parameters**

- **\$userAgent** – string

Returns *php\jsoup\Connection***maxBodySize** (*\$bytes*)**Parameters**

- **\$bytes** – int

Returns *php\jsoup\Connection***timeout** (*\$millis*)**Parameters**

- **\$millis** – int

Returns *php\jsoup\Connection***referrer** (*\$referrer*)**Parameters**

- **\$referrer** – string

Returns *php\jsoup\Connection***followRedirects** (*\$enable*)**Parameters**

- **\$enable** – bool

Returns *php\jsoup\Connection***ignoreHttpErrors** (*\$enable*)**Parameters**

- **\$enable** – bool

Returns *php\jsoup\Connection***ignoreContentType** (*\$enable*)

Parameters

- **\$enable** – bool

Returns *php\jsoup\Connection*

execute()

Returns *php\jsoup\ConnectionResponse*

get()

Returns *php\jsoup\Document*

post()

Returns *php\jsoup\Document*

request()

Returns *php\jsoup\ConnectionRequest*

response()

Returns *php\jsoup\ConnectionResponse*

ConnectionRequest

php\jsoup\ConnectionRequest

abstract class

Methods

timeout(\$millis)

Setter and getter for timeout.

Parameters

- **\$millis** – int - (optional)

Returns int, *php\jsoup\ConnectionRequest*

maxBodySize(\$bytes)

Setter and getter for max of body size.

Parameters

- **\$bytes** – int - (optional)

Returns int, *php\jsoup\ConnectionRequest*

followRedirects(\$enable)

Setter and getter.

Parameters

- **\$enable** – bool - (optional)

Returns bool, *php\jsoup\ConnectionRequest*

ignoreHttpErrors(\$enable)

Parameters

- **\$enable** – bool - (optional)

Returns bool, *php\jsoup\ConnectionRequest*

ignoreContentType (*\$enable*)

Parameters

- **\$enable** – bool - (optional)

Returns bool, *php\jsoup\ConnectionRequest*

ConnectionResponse

php\jsoup\ConnectionResponse
abstract class

Methods

statusCode ()

Get the status code of the response.

Returns int

statusMessage ()

Get the status message of the response.

Returns string

charset ()

Returns string

body ()

Returns string

bodyAsBytes ()

Returns string binary string

contentType ()

Get the response content type (e.g. “text/html”);

Returns string

parse ()

Returns *php\jsoup\Document*

Document

php\jsoup\Document
abstract class

Methods

location ()

Returns string

title (*\$value*)

Parameters

- **\$value** – string - (optional)

Returns string

head()

Returns *php\jsoup\Element*

body()

Returns *php\jsoup\Element*

select (\$cssQuery)

Parameters

- **\$cssQuery** – string

Returns *php\jsoup\Elements*

Element

php\jsoup\Element
abstract class

Methods

html (\$html)

Parameters

- **\$html** – string - (optional)

Returns string

text()

Returns string

nodeName()

Returns string

tagName (\$tagName)

Parameters

- **\$tagName** – string - (optional)

Returns string

isBlock()

Returns bool

id()

Returns string

attr (\$attributeKey, \$attributeValue)

Parameters

- **\$attributeKey** –
- **\$attributeValue** –

– (optional)

Returns `php\jsoup\This`

dataset ()

Returns `array`

parent ()

Returns `php\jsoup\Element`

parents ()

Returns `php\jsoup\Elements`

child (*\$index*)

Parameters

- **\$index** – `int`

Returns `php\jsoup\Element`

children ()

Returns `php\jsoup\Elements`

select (*\$cssQuery*)

Parameters

- **\$cssQuery** – `string`

Returns `php\jsoup\Elements`

Elements

`php\jsoup\Elements`

abstract `class`

implements: `Iterator`

Methods

text ()

Returns `string`

hasText ()

Returns `bool`

html (*\$html*)

Parameters

- **\$html** – `string` - (optional)

Returns `string`

outerHtml ()

Returns `string`

attr (*\$attributeKey*, *\$value*)

Parameters

- **\$attributeKey** – string
- **\$value** –
– (optional)

Returns string, php\jsoup\This**hasAttr** (*\$attributeKey*)**Parameters**

- **\$attributeKey** – string

Returns bool**removeAttr** (*\$attributeKey*)**Parameters**

- **\$attributeKey** – string

Returns *php\jsoup\Elements***addClass** (*\$class*)**Parameters**

- **\$class** – string

Returns *php\jsoup\Elements***removeClass** (*\$class*)**Parameters**

- **\$class** – string

Returns *php\jsoup\Elements***hasClass** (*\$class*)**Parameters**

- **\$class** – string

Returns bool**toggleClass** (*\$class*)**Parameters**

- **\$class** – string

Returns bool**val** (*\$value*)**Parameters**

- **\$value** – string - (optional)

Returns php\jsoup\This**prepend** (*\$html*)**Parameters**

- **\$html** – string

Returns *php\jsoup\Elements*

append (*\$html*)

Parameters

- *\$html* – string

Returns *php\jsoup\Elements*

before (*\$html*)

Parameters

- *\$html* – string

Returns *php\jsoup\Elements*

after (*\$html*)

Parameters

- *\$html* – string

Returns *php\jsoup\Elements*

select (*\$query*)

Parameters

- *\$query* – string

Returns *php\jsoup\Elements*

first ()

Returns *php\jsoup\Element*

last ()

Returns *php\jsoup\Element*

not (*\$query*)

Parameters

- *\$query* – string

Returns *php\jsoup\Elements*

is (*\$query*)

Parameters

- *\$query* – string

Returns bool

parents ()

Returns *php\jsoup\Elements*

Jsoup

php\jsoup\Jsoup

final class

Methods

__construct ()

private

static connect (\$url)

Parameters

- **\$url** – string

Returns *php\jsoup\Connection*

static parse (\$source, \$encoding, \$baseUri)

Parameters

- **\$source** – string, *php\io\File*, *php\io\Stream*
- **\$encoding** – string
- **\$baseUri** – string

Returns *php\jsoup\Document*

static parseText (\$text, \$baseUri)

Parameters

- **\$text** – string
- **\$baseUri** – string - (optional)

Returns *php\jsoup\Document*

lang

ClassLoader

php\lang\ClassLoader

abstract class

Methods

loadClass (\$name)

abstract

Parameters

- **\$name** – string

register ()

unregister ()

Environment

php\lang\Environment
Class Environment

Constants

constant **CONCURRENT**

constant **HOT_RELOAD**

Methods

__construct (*\$parent = NULL, \$flags = 0*)

Parameters

- **\$parent** – *php\lang\Environment*
- **\$flags** – int - Environment::HOT_RELOAD, Environment::CONCURRENT

execute (*\$runnable*)

Executes \$runnable in the environment

Parameters

- **\$runnable** – callable - - in new environment

Returns mixed

importClass (*\$className*)

Imports the \$className to the environment

throws php\lang\Exception - if class not found or already registered

Parameters

- **\$className** – string

exportClass (*\$className*)

Exports the \$className from the environment

throws php\lang\Exception - if class not found or already registered

Parameters

- **\$className** – string

importFunction (*\$functionName*)

Imports the \$functionName to the environment

throws php\lang\Exception - if function not found or already registered

Parameters

- **\$functionName** – string

exportFunction (*\$functionName*)

Exports the \$functionName from the environment

throws php\lang\Exception - if function not found or already registered

Parameters

- **\$functionName** – string

importAutoLoaders ()

defineConstant (*\$name*, *\$value*, *\$caseSensitive* = *true*)

throws `php\lang\Exception` - if constant already registered or value is not scalar type

Parameters

- **\$name** – string
- **\$value** – mixed - - scalar value
- **\$caseSensitive** – bool

onMessage (*\$callback*)

Handles messages that sent to the environment

Parameters

- **\$callback** – callable

sendMessage (*\$message*)

Send message to the environment

Parameters

- **\$message** – mixed

Returns mixed

static current

Get environment of current execution

Returns *php\lang\Environment*

IllegalArgumentException

php\lang\IllegalArgumentException

extends: *php\lang\JavaException*

Class `IllegalArgumentException`

IllegalStateException

php\lang\IllegalStateException

extends: *php\lang\JavaException*

Class `IllegalStateException`

InterruptedException

php\lang\InterruptedException

extends: *php\lang\JavaException*

Class `InterruptedException`

Invoker

php\lang\Invoker

Class for calling methods/functions/etc.

Methods

__construct (*\$callback*)

Parameters

- **\$callback** – callable

callArray (*\$args*)

Call with array arguments

Parameters

- **\$args** – array

Returns mixed

call ()

Call the current callback

Returns int, mixed

__invoke ()

canAccess ()

Check access to invoke the method at a moment

Returns bool

getDescription ()

Returns description of the method - name + argument info

Returns string

getArgumentCount ()

Returns argument count of the method

Returns int

isClosure ()

Checks it is a closure

Returns bool

isNamedFunction ()

Checks it is a named function

Returns bool

isStaticCall ()

Checks it is a static call

Returns bool

isDynamicCall ()

Checks it is a dynamic call

Returns bool

static of (*\$callback*)

Parameters

- **\$callback** – mixed, callable

Returns *php\lang\Invoker*, null - returns null if passed is not callable

JavaClass

php\lang\JavaClass

final class

Methods

__construct (*\$className*)

throws *php\lang\JavaException* if not found class

Parameters

- **\$className** – string - - full name of java class

isStatic ()

Returns bool

isFinal ()

Returns bool

isAbstract ()

Returns bool

isInterface ()

Returns bool

isEnum ()

Returns bool

isAnnotation ()

Returns bool

isArray ()

Returns bool

isPrimitive ()

Returns bool

isAnonymousClass ()

Returns bool

isMemberClass ()

Returns bool

getName ()

Returns string

getSimpleName ()

Returns string

getCanonicalName()
Returns string

getSuperClass()
Returns *php\lang\JavaClass*, null

getModifiers()
Returns int

isAnnotationPresent(\$annotationClassName)
throws *php\lang\JavaException* if class not found

Parameters

- **\$annotationClassName** – string

Returns bool

getInterfaces()
Returns *php\lang\JavaClass[]*

getDeclaredMethod(\$name, \$types)
throws *php\lang\JavaException*

Parameters

- **\$name** – string
- **\$types** – array

Returns *php\lang\JavaMethod*

getDeclaredMethods()
Returns *php\lang\JavaMethod[]*

getDeclaredField(\$name)
throws *php\lang\JavaException*

Parameters

- **\$name** –

Returns *php\lang\JavaField*

getDeclaredFields()
Returns *php\lang\JavaField[]*

newInstance()
throws *php\lang\JavaException*

Returns *php\lang\JavaObject*

newInstanceArgs(\$types, \$arguments)
throws *php\lang\JavaException*

Parameters

- **\$types** – array
- **\$arguments** – array

Returns *php\lang\JavaObject*

isAssignableFrom(\$class)

Parameters

- **\$class** – *php\lang\JavaClass*

Returns bool**isSubClass** (*\$className*)throws *php\lang\JavaException***Parameters**

- **\$className** – string

Returns bool**getEnumConstants** ()**Returns** *php\lang\JavaObject[]***getResource** (*\$name*)**Parameters**

- **\$name** – string

Returns string, null - filename**static primitive** (*\$name*)**Parameters**

- **\$name** – string - - [int, byte, short, char, float, double, boolean, long]

JavaException

php\lang\JavaException

extends: Exception

Children

- class *php\concurrent\TimeoutException*
- class *php\io\IOException*
- class *php\lang\IllegalArgumentException*
- class *php\lang\IllegalStateException*
- class *php\lang\InterruptedException*
- class *php\lang\NumberFormatException*
- class *php\net\SocketException*
- class *php\sql\SQLException*
- class *php\util\RegexException*

Class JavaException

Methods

isRuntimeException ()

Check exception instance of java.lang.RuntimeException

Returns bool

isNullPointerException()

Check exception instance of java.lang.NullPointerException

Returns bool

isIllegalArgumentException()

Check exception instance of java.lang.IllegalArgumentException

Returns bool

isNumberFormatException()

Check exception instance of java.lang.NumberFormatException

Returns bool

getExceptionClass()

Returns *php\lang\JavaClass*

getJavaException()

Returns *php\lang\JavaObject*

printJVMStackTrace()

JavaField

php\lang\JavaField

final class

extends: *php\lang\JavaReflection*

Methods

get (*\$object = null*)

throws *php\lang\JavaException*

Parameters

- **\$object** – *php\lang\JavaObject*

Returns mixed

set (*\$object = null*, *\$value*)

throws *php\lang\JavaException*

Parameters

- **\$object** – *php\lang\JavaObject*
- **\$value** –

isStatic()

Returns bool

isFinal()

Returns bool

isPublic()

Returns bool

isProtected()
Returns bool

isPrivate()
Returns bool

isTransient()
Returns bool

isVolatile()
Returns bool

getModifiers()
Returns int

getName()
Returns string

getDeclaringClass()
Returns *php\lang\JavaClass*

JavaMethod

php\lang\JavaMethod
final class

extends: *php\lang\JavaReflection*

Methods

invoke (*\$object = null*)
Invoke method

Parameters

- **\$object** – *php\lang\JavaObject*

invokeArgs (*\$object = null, \$arguments*)

Parameters

- **\$object** – *php\lang\JavaObject*
- **\$arguments** – array

getName()
Returns string

isStatic()
Returns bool

isFinal()
Returns bool

isAbstract()
Returns bool

isPublic()
Returns bool

isProtected()
Returns bool

isPrivate()
Returns bool

isNative()
Returns bool

isSynchronized()
Returns bool

isVarArgs()
Returns bool

getDeclaringClass()
Returns *php\lang\JavaClass*

getReturnedType()
Returns *php\lang\JavaClass*

isAnnotationPresent (\$annotationClassName)
throws *php\lang\JavaException*

Parameters

- \$annotationClassName – string

Returns bool

getParameterTypes()
Returns *php\lang\JavaClass[]*

getParameterCount()
Returns int

JavaObject

php\lang\JavaObject
final class

Methods

getClass()
Get class of object
Returns *php\lang\JavaClass*

getClassName()
Get name of class of object
Returns string

JavaReflection

php\lang\JavaReflection
abstract class

Children

- final class *php\lang\JavaField*
- final class *php\lang\JavaMethod*

Module

php\lang\Module
Class Module

Methods

__construct (*\$source*, *\$compiled* = false, *\$debugInformation* = true)
Register all functions and classes of module in current environment

Parameters

- **\$source** – *php\io\File*, *php\io\Stream*, string
- **\$compiled** – bool
- **\$debugInformation** – bool

getName ()

Returns string

call (*\$variables* = null)

Include module and return result

Parameters

- **\$variables** – array

Returns mixed

dump (*\$target*, *\$saveDebugInfo* = true)
throws >

Parameters

- **\$target** – *php\io\File*, *php\io\Stream*, string
- **\$saveDebugInfo** – bool

NumberFormatException

php\lang\NumberFormatException
extends: *php\lang\JavaException*
Class NumberFormatException

Process

php\lang\Process

Class Process

Methods

__construct (*\$commands*, *\$directory* = null, *\$environment* = null)

Parameters

- **\$commands** – array
- **\$directory** – null, string, *php\io\File*
- **\$environment** – array

start ()

throws *php\lang\IllegalStateException*

Returns *php\lang\Process*

startAndWait ()

Causes the current thread to wait, if necessary, until the process represented by this *Process* object has terminated. This method returns immediately if the subprocess has already terminated. If the subprocess has not yet terminated, the calling thread will be blocked until the subprocess exits.

throws *php\lang\IllegalStateException*

Returns *php\lang\Process*

getExitValue ()

Returns the exit value for the subprocess.

throws *php\lang\IllegalStateException*

Returns int, null - null if process is working

destroy ()

Kills the subprocess. The subprocess represented by this *Process* object is forcibly terminated.

throws *php\lang\IllegalStateException*

getInput ()

Returns the input stream connected to the normal output of the subprocess. The stream obtains data piped from the standard output of the process represented by this *Process* object.

throws *php\lang\IllegalStateException*

Returns *php\io\Stream*

getOutput ()

Returns the output stream connected to the normal input of the subprocess. Output to the stream is piped into the standard input of the process represented by this *Process* object.

throws *php\lang\IllegalStateException*

Returns *php\io\Stream*

getError ()

Returns the input stream connected to the error output of the subprocess. The stream obtains data piped from the error output of the process represented by this *Process* object.

throws *php\lang\IllegalStateException*

Returns *php\io\Stream*

System

php\lang\System

final class

Methods

__construct ()

private

static halt (\$status)

Exit from program with status globally

Parameters

- **\$status** – int

static gc

static getEnv

Returns string[]

static getProperty (\$name, \$def = '')

Gets a system property by name

Parameters

- **\$name** –
- **\$def** – string

Returns string

Thread

php\lang\Thread

Class Thread

Constants

constant MAX_PRIORITY

constant MIN_PRIORITY

constant NORM_PRIORITY

Methods

__construct (\$runnable, \$env = null, \$group = null)

Parameters

- **\$runnable** – callable
- **\$env** – *php\lang\Environment*
- **\$group** – *php\lang\ThreadGroup*

getId()

Returns int

getName()

Returns string

setName(\$value)

Parameters

- **\$value** – string

getGroup()

Returns *php\lang\ThreadGroup*

isDaemon()

Returns bool

setDaemon(\$value)

Parameters

- **\$value** – bool

isInterrupted()

Returns bool

isAlive()

Returns bool

start()

run()

interrupt()

join(\$millis = 0, \$nanos = 0)

Waits at most \$millis milliseconds plus \$nanos nanoseconds for this thread to die.

Parameters

- **\$millis** – int
- **\$nanos** – int

static doYield

static sleep(\$millis, \$nanos = 0)

Causes the currently executing thread to sleep (temporarily cease execution)

Parameters

- **\$millis** – int
- **\$nanos** – int

static getActiveCount

Returns int

static current

Get current thread

Returns *php\lang\Thread*

static sync (*\$object*, *\$runnable*)

Parameters

- **\$object** – object
- **\$runnable** – callable

ThreadGroup

php\lang\ThreadGroup

Class ThreadGroup

Methods

__construct (*\$name*, *\$parent* = *null*)

Parameters

- **\$name** –
- **\$parent** – *php\lang\ThreadGroup*

getName ()

Returns string

getParent ()

Returns *php\lang\ThreadGroup*, null

getActiveCount ()

Returns int

getActiveGroupCount ()

Returns int

isDaemon ()

Returns bool

setDaemon (*\$value*)

Parameters

- **\$value** – bool

isDestroyed ()

Returns bool

getMaxPriority ()

Returns int

setMaxPriority (*\$value*)

Parameters

- **\$value** – int

destroy ()

checkAccess ()

Determines if the currently running thread has permission to modify this thread group.

throws *php\lang\JavaException*

interrupt ()

lib

Binary

php\lib\Binary

Class for working with binary strings

Class Binary

Methods

__construct ()

private

static of (\$string)

Parameters

- **\$string** – string, array

Returns string - binary string

Number

php\lib\Number

Utils for numbers

Class num

Methods

__construct ()

private

static compare (\$num1, \$num2)

Compare two numbers

Note: it can be used as comparator for number sorting

Parameters

- **\$num1** – int, double
- **\$num2** – int, double

Returns int 0 if are equal, 1 if \$num1 > \$num2, -1 if \$num1 < \$num2

static toBin (\$number)

Returns a string representation of the \$number argument as an unsigned integer in base 2.

Parameters

- **\$number** – int

Returns string**static toOctal** (*\$number*)

Returns a string representation of the \$number argument as an unsigned integer in base 8.

Parameters

- **\$number** – int

Returns string**static toHex** (*\$number*)

Returns a string representation of the \$number argument as an unsigned integer in base 16.

Parameters

- **\$number** – int

Returns string**static toString** (*\$number*, *\$radix*)

Returns a string representation of the first argument in the radix specified by the second argument.

Parameters

- **\$number** – int
- **\$radix** – int

Returns string**static reverse** (*\$number*)

Returns the value obtained by reversing the order of the bits in the two's complement binary representation of the specified {*@code long*} value.

Parameters

- **\$number** – int

Returns int**static decode** (*\$string*)

Decodes a string into a integer. Accepts decimal, hexadecimal, and octal numbers

Parameters

- **\$string** – string

Returns string or false if invalid number format**static format** (*\$number*, *\$pattern*, *\$decSep* = '.', *\$groupSep* = ',')**Parameters**

- **\$number** – int, double
- **\$pattern** –
- **\$decSep** – string
- **\$groupSep** – string

Returns string

String

`php\lib\String`

Class `str`

Methods

__construct ()

private

static pos (*\$string, \$search, \$fromIndex = 0*)

Returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

Parameters

- **\$string** – string
- **\$search** – string - the substring to search for
- **\$fromIndex** – int - the index from which to start the search.

Returns int - returns -1 if not found

static posIgnoreCase (*\$string, \$search, \$fromIndex = 0*)

The same method as `pos()` only with ignoring case characters

Parameters

- **\$string** – string
- **\$search** – string - the substring to search for.
- **\$fromIndex** – int - the index from which to start the search.

Returns int - returns -1 if not found

static lastPos (*\$string, \$search, \$fromIndex = null*)

Returns the index within this string of the last occurrence of the specified substring. The last occurrence of the empty string `""` is considered to occur at the index value `$string.length`.

Parameters

- **\$string** – string
- **\$search** – string - the substring to search for.
- **\$fromIndex** – null, int - - null means `$fromIndex` will be equal `$string.length`

Returns int - returns -1 if not found

static lastPosIgnoreCase (*\$string, \$search, \$fromIndex = null*)

The same method as `lastPos()` only with ignoring case characters

Parameters

- **\$string** – string
- **\$search** – string - the substring to search for.
- **\$fromIndex** – null, int - - null means `$fromIndex` will be equal `$string.length`

Returns int

static sub (*\$string*, *\$beginIndex*, *\$endIndex* = null)

Returns a new string that is a substring of this string. The substring begins at the specified *\$beginIndex* and extends to the character at index *\$endIndex* - 1. Thus the length of the substring is *endIndex* - *beginIndex*.

Parameters

- ***\$string*** – string
- ***\$beginIndex*** – int
- ***\$endIndex*** – null, int - When *\$endIndex* equals to null then it will be equal *\$string.length*

Returns string - return false if params are invalid

static compare (*\$string1*, *\$string2*)

Compares two strings lexicographically. The comparison is based on the Unicode value of each character in the strings.

The character sequence represented by *\$string1* String is compared lexicographically to the character sequence represented by *\$string2*. The result is a negative integer if *\$string1* lexicographically precedes *\$string2*. The result is a positive integer if *\$string1* lexicographically follows *\$string2*. The result is zero if the strings are equal; **compare** returns **0** exactly when the strings are equal

Parameters

- ***\$string1*** – string - - first string
- ***\$string2*** – string - - second string

Returns int

static compareIgnoreCase (*\$string1*, *\$string2*)

The same method as **compare** () only with ignoring case characters

Parameters

- ***\$string1*** – string
- ***\$string2*** – string

Returns int

static equalsIgnoreCase (*\$string1*, *\$string2*)

Checks that the strings are equal with ignoring case characters

Parameters

- ***\$string1*** – string
- ***\$string2*** – string

Returns bool

static startsWith (*\$string*, *\$prefix*, *\$offset* = 0)

Tests if the substring of this string beginning at the specified index starts with the specified prefix.

Returns `true` if the character sequence represented by the argument is a prefix of the substring of this object starting at index *offset*; `false` otherwise. The result is `false` if *toffset* is negative or greater than the length of this *\$string*; otherwise the result is the same as the result of the expression

`startsWith(sub($offset), $prefix)`

Parameters

- **\$string** – string
- **\$prefix** – string
- **\$offset** – int - where to begin looking in this string

Returns bool**static endsWith** (*\$string*, *\$suffix*)

Tests if this string ends with the specified suffix.

Parameters

- **\$string** – string
- **\$suffix** – string

Returns bool**static lower** (*\$string*)Converts all of the characters in *\$string* to lower case using the rules of the default locale.**Parameters**

- **\$string** – string

Returns string**static upper** (*\$string*)Converts all of the characters in *\$string* to upper case using the rules of the default locale.**Parameters**

- **\$string** – string

Returns string**static length** (*\$string*)Returns the length of *\$string*. The length is equal to the number of *Unicode code units* in the string.**Parameters**

- **\$string** – string

Returns int**static replace** (*\$string*, *\$target*, *\$replacement*)

Replaces each substring of this string that matches the literal target sequence with the specified literal replacement sequence. The replacement proceeds from the beginning of the string to the end, for example, replacing “aa” with “b” in the string “aaa” will result in “ba” rather than “ab”.

Parameters

- **\$string** – string
- **\$target** – string - The sequence of char values to be replaced
- **\$replacement** – string - The replacement sequence of char values

Returns string

static repeat (*\$string*, *\$amount*)

Return s a new string consisting of the original *\$string* repeated

Parameters

- ***\$string*** – string
- ***\$amount*** – int - number of times to repeat str

Returns string

```
trim($string, $charList = '  
' )
```

Returns a copy of the string, with leading and trailing whitespace omitted.

param *\$string* string

param *\$charList* string

returns string

```
trimRight($string, $charList = '  
' )
```

param *\$string* string

param *\$charList* string

returns string

```
trimLeft($string, $charList = '  
' )
```

param *\$string* string

param *\$charList* string

returns string

static reverse (*\$string*)

Parameters

- ***\$string*** – string

Returns string

static shuffle (*\$string*)

Returns a randomized string based on chars in *\$string*

Parameters

- ***\$string*** – string

Returns string

```
static random ($length = 16, $set = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJK-  
LZXCVCBNM0123456789')
```

Parameters

- ***\$length*** – int
- ***\$set*** – string

Returns string

static split (*\$string*, *\$separator*, *\$limit* = 0)

The method like `explode()` in Zend PHP

Parameters

- **\$string** – string
- **\$separator** – string
- **\$limit** – int

Returns array

static join (*\$iterable*, *\$separator*, *\$limit* = 0)

The method like `implode()` in Zend PHP

Parameters

- **\$iterable** – array, `php\lib\Iterator`
- **\$separator** – string
- **\$limit** – int

Returns string

static encode (*\$string*, *\$charset*)

Converts *\$string* by using *\$charset* and returns a binary string

Parameters

- **\$string** – string
- **\$charset** – string - e.g. UTF-8, Windows-1251, etc.

Returns string binary string

static decode (*\$string*, *\$charset*)

Decodes *\$string* by using *\$charset* to UNICODE, returns a unicode string

Parameters

- **\$string** – string
- **\$charset** – string - e.g. UTF-8, Windows-1251, etc.

Returns string binary string

static isNumber (*\$string*, *\$bigNumbers* = true)

Returns *true* if *\$string* is integer number (e.g: '12893', '3784', '0047')

- for 123 - true
- for 00304 - true
- for 3389e4 - false
- for 3.49 - false
- for “23 “ - false

Parameters

- **\$string** – string
- **\$bigNumbers** – bool

Returns bool

static isLower (*\$string*)

Parameters

- **\$string** – string

Returns bool

static isUpper (*\$string*)

Parameters

- **\$string** –

Returns bool

mail

Email

php\mail\Email

Class Email

Methods

setFrom (*\$email*, *\$name*, *\$charset*)

Parameters

- **\$email** – string
- **\$name** – string - (optional)
- **\$charset** – string - (optional)

Returns php\mail\this

setCharset (*\$charset*)

Parameters

- **\$charset** – string

Returns php\mail\this

setSubject (*\$subject*)

Parameters

- **\$subject** – string

Returns php\mail\this

setTo (*\$addresses*)

Parameters

- **\$addresses** – array

Returns php\mail\this

setCc (*\$addresses*)

Parameters

- **\$addresses** – array

Returns php\mail\this

setBcc (*\$addresses*)

Parameters

- **\$addresses** – array

Returns php\mail\this

setBounceAddress (*\$email*)

Parameters

- **\$email** – array

Returns php\mail\this

setHeaders (*\$headers*)

Parameters

- **\$headers** – array

Returns php\mail\this

setMessage (*\$message*)

Parameters

- **\$message** – string

Returns php\mail\this

setHtmlMessage (*\$message*)

Parameters

- **\$message** – string

Returns php\mail\this

setTextMessage (*\$message*)

Parameters

- **\$message** – string

Returns php\mail\this

attach (*\$content*, *\$contentType*, *\$name*, *\$description* = '')

Parameters

- **\$content** – string, *php\io\File*, *php\io\Stream*
- **\$contentType** – string
- **\$name** – string
- **\$description** – string

Returns php\mail\this

send (*\$backend*)

Sends the email. Internally we build a MimeMessage which is afterwards sent to the SMTP server.

Parameters

- **\$backend** – *php\mail\EmailBackend*

Returns string the message id of the underlying MimeMessage

EmailBackend

`php\mail\EmailBackend`

Class EmailBackend

Properties

property hostName

string

The host name of the SMTP server.

property smtpPort

string

The listening port of the SMTP server.

property sslSmtpPort

string

The current SSL port used by the SMTP transport.

property sendPartial

bool

Sending partial email.

property socketTimeout

int

The socket I/O timeout value in milliseconds.

property socketConnectionTimeout

int

The socket connection timeout value in milliseconds.

property sslOnConnect

bool

Whether SSL/TLS encryption for the transport is currently enabled (SMTPS/POPS).

property sslCheckServerIdentity

bool

Whether the server identity is checked as specified by RFC 2595.

Methods

getHostName ()

protected

Returns string

setHostName (\$hostName)

protected

Parameters

- **\$hostName** – string

getSmtpport ()
protected

Returns string

setSmtpport (\$smtpPort)
protected

Parameters

- **\$smtpPort** – string

getSslSmtpport ()
protected

Returns string

setSslSmtpport (\$sslSmtpport)
protected

Parameters

- **\$sslSmtpport** – string

isSendPartial ()
protected

Returns boolean

setSendPartial (\$sendPartial)
protected

Parameters

- **\$sendPartial** – boolean

getSocketTimeout ()
protected

Returns int

setSocketTimeout (\$socketTimeout)
protected

Parameters

- **\$socketTimeout** – int

getSocketConnectionTimeout ()
protected

Returns int

setSocketConnectionTimeout (\$socketConnectionTimeout)
protected

Parameters

- **\$socketConnectionTimeout** – int

isSslOnConnect ()
protected

Returns boolean

setSslOnConnect (\$sslOnConnect)
protected

Parameters

- **\$sslOnConnect** – boolean

isSslCheckServerIdentity ()**protected****Returns** boolean**setSslCheckServerIdentity** (*\$sslCheckServerIdentity*)**protected****Parameters**

- **\$sslCheckServerIdentity** – boolean

setAuthentication (*\$login*, *\$password*)

Sets the userName and password if authentication is needed. If this method is not used, no authentication will be performed.

Parameters

- **\$login** – string
- **\$password** – string

clearAuthentication ()**net****NetStream****php\net\NetStream****extends:** *php\io\Stream*

http, ftp protocols

Class NetStream

Methods

read (*\$length*)**Parameters**

- **\$length** –

readFully (*\$bufferSize* = 4096)**Parameters**

- **\$bufferSize** – int

Returns mixed, void**readFullyWithCallback** (*\$bufferSize*, *\$callback*)**Parameters**

- **\$bufferSize** – int
- **\$callback** – callable - (NetStream \$this, \$len)

write (*\$value*, *\$length* = null)

Parameters

- **\$value** –
- **\$length** –

eof()

seek() (*\$position*)

Parameters

- **\$position** –

getPosition()

close()

getUrl()

Returns *php\net\URL*

setProxy() (*\$proxy*)

Parameters

- **\$proxy** – *php\net\Proxy*

getProxy()

Returns *php\net\Proxy*

getUrlConnection()

Returns *php\net\URLConnection*

Proxy

php\net\Proxy

Class Proxy

Methods

__construct() (*\$type*, *\$host*, *\$port*)

Parameters

- **\$type** – string - - DIRECT, HTTP or SOCKS
- **\$host** – string
- **\$port** – int

address()

Returns string host with port

type()

Returns string DIRECT, HTTP or SOCKS

ServerSocket

`php\net\ServerSocket`

Class SocketServer

Methods

__construct (*\$port = null*, *\$backLog = 50*)

Parameters

- *\$port* – int
- *\$backLog* – int

accept ()

throws *php\io\IOException*

Returns *php\net\Socket*

bind (*\$hostname*, *\$port*, *\$backLog = 50*)

throws *php\net\SocketException*

Parameters

- *\$hostname* – string
- *\$port* – int
- *\$backLog* – int

close ()

throws *php\io\IOException*

isClosed ()

Returns bool

isBound ()

Returns the binding state of the ServerSocket.

Returns bool

setSoTimeout (*\$timeout*)

Enable/disable SO_TIMEOUT with the specified timeout, in milliseconds.

throws *php\net\SocketException*

Parameters

- *\$timeout* – int

setReuseAddress (*\$on*)

Enable/disable the SO_REUSEADDR socket option.

throws *php\net\SocketException*

Parameters

- *\$on* – bool

setReceiveBufferSize (*\$size*)

throws *php\net\SocketException*

Parameters

- **\$size** – int

setPerformancePreferences (*\$connectTime*, *\$latency*, *\$bandWidth*)

Sets performance preferences for this ServerSocket.

! Not implemented yet for TCP/IP

Parameters

- **\$connectTime** – int
- **\$latency** – int
- **\$bandWidth** – int

static findAvailableLocalPort

Returns int

Socket

php\net\Socket

Class Socket

Methods

__construct (*\$host = null*, *\$port = null*)

Parameters

- **\$host** – null, string
- **\$port** – null, int

getOutput ()

throws *php\io\IOException*

Returns *php\io\MiscStream*

getInput ()

throws *php\io\IOException*

Returns *php\io\MiscStream*

getLocalAddress ()

Returns string

getAddress ()

Returns string

getLocalPort ()

Returns int

getPort ()

Returns int

close ()

throws *php\io\IOException*

shutdownInput ()

throws *php\io\IOException*

shutdownOutput ()
throws *php\io\IOException*

isConnected ()
Returns bool

isClosed ()
Returns bool

isBound ()
Returns bool

isInputShutdown ()
Returns bool

isOutputShutdown ()
Returns bool

connect (\$hostname, \$port, \$timeout = null)
Connects this socket to the server

Parameters

- **\$hostname** – string
- **\$port** – int
- **\$timeout** – null, int

bind (\$hostname, \$port)
Binds the socket to a local address.

throws *php\net\SocketException*

Parameters

- **\$hostname** – string
- **\$port** – int

bindDefault ()

setSoTimeout (\$timeout)
Enable/disable SO_TIMEOUT with the specified timeout, in milliseconds.

throws *php\net\SocketException*

Parameters

- **\$timeout** – int

setSoLinger (\$on, \$linger)
throws *php\net\SocketException*

Parameters

- **\$on** – bool
- **\$linger** – int

setReuseAddress (\$on)
Enable/disable the SO_REUSEADDR socket option.

throws *php\net\SocketException*

Parameters

- **\$on** – bool

setReceiveBufferSize (*\$size*)
throws *php\net\SocketException*

Parameters

- **\$size** – int

setTcpNoDelay (*\$on*)
throws *php\net\SocketException*

Parameters

- **\$on** – bool

setKeepAlive (*\$on*)
throws *php\net\SocketException*

Parameters

- **\$on** – bool

setOOBInline (*\$on*)
throws *php\net\SocketException*

Parameters

- **\$on** – bool

setSendBufferSize (*\$size*)
throws *php\net\SocketException*

Parameters

- **\$size** – int

setTrafficClass (*\$tc*)
Sets traffic class or type-of-service octet in the IP header for packets sent from this Socket.

Parameters

- **\$tc** – int

setPerformancePreferences (*\$connectTime*, *\$latency*, *\$bandWidth*)
Sets performance preferences for this ServerSocket.

! Not implemented yet for TCP/IP

Parameters

- **\$connectTime** – int
- **\$latency** – int
- **\$bandWidth** – int

sendUrgentData (*\$data*)
Send one byte of urgent data on the socket. The byte to be sent is the lowest eight bits of the data parameter.

throws *php\net\SocketException*

Parameters

- **\$data** – int

SocketException

php\net\SocketException

extends: *php\lang\JavaException*

Class SocketException

URL

php\net\URL

Class URL

Methods

__construct (*\$uri*)

Parameters

- **\$uri** – string

openConnection (*\$proxy*)

Parameters

- **\$proxy** – *php\net\Proxy* - (optional)

Returns *php\net\URLConnection*

getAuthority ()

Gets the authority part of this URL

Returns string

getPort ()

Returns int

getDefaultPort ()

Gets the default port number of the protocol associated with this URL. If the URL scheme or the URLStreamHandler for the URL do not define a default port number, then -1 is returned.

Returns int

getProtocol ()

Gets the protocol name of this URL

Returns string

getHost ()

Returns string

getFile ()

Gets the file name of this `URL`. The returned file portion will be the same as `getPath()`, plus the concatenation of the value of `getQuery()`, if any. If there is no query portion, this method and `getPath()` will return identical results.

Returns string

getPath ()

Returns string

getQuery ()

Returns string

getRef()

Gets the anchor (also known as the “reference”) of this URL

Returns string

sameFile(\$url)

Compares two URLs, excluding the fragment component.

Parameters

- *\$url* – *php\ne\URL*

Returns bool

toString()

Returns string

toExternalForm()

Constructs a string representation of this URL. The string is created by calling the toExternalForm method of the stream protocol handler for this object.

Returns string

openStream()

Returns *php\io\Stream*

__toString()

Returns string

URLConnection

php\net\URLConnection

Class URLConnection

Properties

property doOutput

bool

property doInput

bool

property requestMethod

string

POST, GET, PUT, etc.

property connectTimeout

php\net\int millis

that specifies the connect timeout value in milliseconds

property readTimeout

php\net\int millis

the read timeout to a specified timeout, in milliseconds.

property useCaches

bool

property ifModifiedSince

php\net\int millis

property followRedirects

bool

property url

php\net\URL

read-only

property responseCode

int

read-only

property responseMessage

int

read-only

property contentLength

php\net\int bytes

read-only

int the content length of the resource that this connection's URL references, -1 if the content length is not known, or if the content length is greater than Integer.MAX_VALUE.

property contentType

string

read-only

property contentEncoding

string

read-only

property expiration

int

read-only

property lastModified

int

read-only

property usingProxy

bool

read-only

Methods

__construct (*\$parent*)

protected

Parameters

- *\$parent* – *php\net\URLConnection*

connect ()

getHeaderField (*\$name*)

Parameters

- *\$name* – string

getHeaderFields ()

Returns array

getInputStream ()

Returns *php\io\Stream*

getErrorStream ()

Returns *php\io\Stream*

getOutputStream ()

Returns *php\io\Stream*

setRequestProperty (*\$name*, *\$value*)

Parameters

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

getRequestProperty (*\$name*)

Parameters

- *\$name* – string

getRequestProperties ()

Returns array

disconnect ()

setChunkedStreamingMode (*\$chunklen*)

This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is **not** known in advance. In this mode, chunked transfer encoding is used to send the request body. Note, not all HTTP servers support this mode.

Parameters

- *\$chunklen* – int - The number of bytes to write in each chunk.

If chunklen is less than or equal to zero, a default value will be used.

static guessContentTypeFromStream (*\$stream*)

Tries to determine the type of an input stream based on the characters at the beginning of the input stream. This method can be used by subclasses that override the `getContentType` method.

Parameters

- *\$stream* – *php\io\Stream*

Returns string

static guessContentTypeFromName (*\$name*)

Parameters

- *\$name* – string

sql

SqlConnection

php\sql\SqlConnection
abstract class

Constants

constant TRANSACTION_READ_UNCOMMITTED

constant TRANSACTION_READ_COMMITTED

constant TRANSACTION_REPEATABLE_READ

constant TRANSACTION_NONE

constant TRANSACTION_SERIALIZABLE

Properties

property autoCommit

bool

Disable this to use transaction mode.

property readOnly

bool

property transactionIsolation

int

See SqlConnection::TRANSACTION_* constants

property catalog

string

property schema

string

Methods

identifier (*\$name*)

throws *php\sql\SqlException*

Parameters

- *\$name* – string

Returns string

query (*\$sql*, *\$arguments* = null)

Parameters

- *\$sql* – string
- *\$arguments* – array

Returns *php\sql\SqlStatement*

commit ()

Makes all changes made since the previous commit/rollback permanent and releases any database locks currently held by this Connection object.

throws *php\sql\SQLException*

rollback ()

Undoes all changes made in the current transaction and releases any database locks currently held by this Connection object.

throws *php\sql\SQLException*

close ()

throws *php\sql\SQLException*

getCatalogs ()

Returns array

getSchemas ()

Returns array

getMetaData ()

Returns array

SqlConnectionPool

php\sql\SqlConnectionPool

Class SqlConnectionPool

Methods

__construct (*\$parent*)

protected

Parameters

- **\$parent** – *php\sql\SqlConnectionPool*

getConnection ()

Returns *php\sql\SqlConnection*

setUser (*\$username*)

Parameters

- **\$username** – string

Returns php\sql\this

setPassword (*\$password*)

Parameters

- **\$password** – string

Returns php\sql\this

setMaxPoolSize (*\$value*)

Parameters

- **\$value** – int

Returns `php\sql\`*\$this*

setIdleTimeout (*\$millis*)

Parameters

- **\$millis** – int

Returns `php\sql\`*\$this*

setMaxLifetime (*\$millis*)

Parameters

- **\$millis** – int

Returns `php\sql\`*\$this*

setMinimumIdle (*\$millis*)

Parameters

- **\$millis** – int

Returns `php\sql\`*\$this*

SqlDriverManager

`php\sql\SqlDriverManager`

Class DriverManager

Methods

static install (*\$driverName*)

throws *php\sql\SQLException* if cannot install or find driver.

Parameters

- **\$driverName** – string - - mysql, pgsql, postgres, mssql, firebird, sybase, sqlite, etc.

static getConnection (*\$url*, *\$options*)

throws *php\sql\SQLException* if a database access error occurs

Parameters

- **\$url** – string
- **\$options** – array - (optional) username, password, etc.

Returns *php\sql\SqlConnection*

static getPool (*\$url*, *\$driverName*, *\$options*)

Parameters

- **\$url** – string
- **\$driverName** – string - - mysql, pgsql, postgres, mssql, firebird, sybase, sqlite, etc.
- **\$options** – array - (optional)

Returns *php\sql\SqlConnectionPool*

SQLException

php\sql\SQLException
extends: *php\lang\JavaException*
Class SQLException

SqlResult

php\sql\SqlResult
abstract class

Methods

isLast ()
throws *php\sql\SQLException*
Returns bool

isFirst ()
throws *php\sql\SQLException*
Returns bool

delete ()
Deletes current row.
throws *php\sql\SQLException*

isDeleted ()
throws *php\sql\SQLException*
Returns bool

refresh ()
throws *php\sql\SQLException*

get (\$column)
throws *php\sql\SQLException*
Parameters
• **\$column** – string
Returns mixed, *php\io\Stream*, *php\time\Time*

toArray (\$assoc = true)
throws *php\sql\SQLException*
Parameters
• **\$assoc** – bool
Returns array

SqlStatement

php\sql\SqlStatement
abstract class
implements: Iterator

Methods

bind (*\$index*, *\$value*)

Parameters

- **\$index** – int
- **\$value** – mixed

bindDate (*\$index*, *\$time*)

Parameters

- **\$index** – int
- **\$time** – *php\time\Time*

bindTime (*\$index*, *\$time*)

Parameters

- **\$index** – int
- **\$time** – *php\time\Time*

bindTimestamp (*\$index*, *\$time*)

Parameters

- **\$index** – int
- **\$time** – *php\time\Time*, int

bindBlob (*\$index*, *\$blob*)

Parameters

- **\$index** – int
- **\$blob** – string, *php\io\File*, *php\io\Stream*

fetch ()

throws *php\sql\SQLException*

Returns *php\sql\SqlResult*

update ()

throws *php\sql\SQLException*

Returns int

getLastInsertId ()

Returns mixed

getGeneratedKeys ()

Returns *php\sql\SqlResult*

current ()

Returns *php\sql\SqlResult*

next ()

key ()

valid ()

`rewind()`

swing

Border

`php\swing\Border`

Class Border

Methods

`isOpaque()`

Returns `bool`

`static createEmpty($top, $left, $bottom, $right)`

Parameters

- `$top` – int
- `$left` – int
- `$bottom` – int
- `$right` – int

Returns *php\swing\Border*

`static createBevel($type, $highlightColor, $shadowColor)`

Parameters

- `$type` – string - - RAISED or LOWERED
- `$highlightColor` – *php\swing\Color*, array, int
- `$shadowColor` – *php\swing\Color*, array, int

Returns *php\swing\Border*

`static createSoftBevel($type, $highlightColor, $shadowColor)`

Parameters

- `$type` – string - - RAISED or LOWERED
- `$highlightColor` – *php\swing\Color*, array, int
- `$shadowColor` – *php\swing\Color*, array, int

Returns *php\swing\Border*

`static createEtchedBevel($type, $highlightColor, $shadowColor)`

Parameters

- `$type` – string - - RAISED or LOWERED
- `$highlightColor` – *php\swing\Color*, array, int
- `$shadowColor` – *php\swing\Color*, array, int

Returns *php\swing\Border*

`static createTitled($title, $border = null, $titleFont = null, $titleColor = null)`

Parameters

- **\$title** – string
- **\$border** – *php\swing\Border*
- **\$titleFont** – *php\swing\Font*
- **\$titleColor** – *php\swing\Color*, array, int

Returns *php\swing\Border*

static createLine (*\$color*, *\$size* = 1, *\$rounded* = false)

Parameters

- **\$color** – *php\swing\Color*, array, int
- **\$size** – int
- **\$rounded** – bool

Returns *php\swing\Border*

static createDashed (*\$color*, *\$thickness* = 1, *\$length* = 2, *\$spacing* = 1, *\$rounded* = false)

Parameters

- **\$color** – *php\swing\Color*, array, int
- **\$thickness** – int
- **\$length** – int
- **\$spacing** – int
- **\$rounded** – bool

Returns *php\swing\Border*

Color

php\swing\Color

Properties

property rgb

int

read-only

property alpha

int

read-only

property red

int

read-only

property green

int

read-only

property blue

int

read-only

Methods

__construct (*\$rgb*, *\$hasAlpha* = *false*)

Parameters

- *\$rgb* – int
- *\$hasAlpha* – bool

darker ()

Creates a new Color that is a darker version of this

Returns *php\swing\Color*

brighter ()

Creates a new Color that is a brighter version of this

Returns *php\swing\Color*

static rgb (*\$r*, *\$g*, *\$b*, *\$alpha* = 255)

Parameters

- *\$r* – int -- 0 .. 255
- *\$g* – int -- 0 .. 255
- *\$b* – int -- 0 .. 255
- *\$alpha* – int -- 0 .. 255

Returns *php\swing\Color*

static floatRgb (*\$r*, *\$g*, *\$b*, *\$alpha* = 1.0)

Parameters

- *\$r* – double -- between 0 and 1
- *\$g* – double -- between 0 and 1
- *\$b* – double -- between 0 and 1
- *\$alpha* – double -- between 0 and 1

Returns *php\swing\Color*

static decode (*\$nm*)

Decode color from a hex string (#RGB, 0xRGB, etc...)

Parameters

- *\$nm* –

Returns *php\swing\Color*

event

CaretEvent

`php\swing\event\CaretEvent`

Events:

- `caretUpdate`

Class `CaretEvent`

Properties

property dot

`int`

read-only

property mark

`int`

read-only

property target

php\swing\UIElement

read-only

ComponentEvent

`php\swing\event\ComponentEvent`

abstract class

Children

- **class** *php\swing\event\FocusEvent*
- **class** *php\swing\event\KeyEvent*
- **class** *php\swing\event\MouseEvent*
- **class** *php\swing\event\WindowEvent*

Properties

property target

php\swing\UIElement

read-only

FocusEvent

php\swing\event\FocusEvent

extends: *php\swing\event\ComponentEvent*

Events:

- focus
- blur

Class FocusEvent

Properties

property temporary

bool

read-only

property paramString

string

read-only

HyperlinkEvent

php\swing\event\HyperlinkEvent

extends: *php\swing\event\SimpleEvent*

Class HyperlinkEvent

Properties

property url

string

read-only

property description

string

read-only

property attributes

string[]

read-only

ItemEvent

php\swing\event\ItemEvent

Events:

- changed

Class CaretEvent

Properties

property item

string

read-only**property target***php\swing\UIElement***read-only**

Methods

isSelected()**Returns** bool**isDeselected()****Returns** bool

KeyEvent

php\swing\event\KeyEvent**extends:** *php\swing\event\ComponentEvent*

Events:

- keyUp
- keyDown
- keyPress

Class KeyEvent

Properties

property keyChar

string

read-only**property keyCode**

int

read-only**property keyLocation**

int

read-only**property actionKey**

bool

read-only

MouseEvent

php\swing\event\MouseEvent

extends: *php\swing\event\ComponentEvent*

Children

- **class** *php\swing\event\MouseWheelEvent*

Events:

- click
- mousePress
- mouseRelease
- mouseEnter
- mouseExit
- mouseDrag
- mouseMove

Class MouseEvent

Properties

property x
int

read-only

property y
int

read-only

property screenX
int

read-only

property screenY
int

read-only

property button
int

read-only

property clickCount
int

read-only

property popupTrigger
bool

read-only

MouseEvent

php\swing\event\MouseEvent

extends: *php\swing\event\MouseEvent*

Events:

- mouseWheel

Class MouseEvent

Properties

property scrollAmount

int

read-only

property scrollType

int

read-only

property wheelRotation

int

read-only

property unitsToScroll

int

read-only

SimpleEvent

php\swing\event\SimpleEvent

Class SimpleEvent

Properties

property target

php\swing\UIElement

read-only

WindowEvent

php\swing\event\WindowEvent

extends: *php\swing\event\ComponentEvent*

Events:

- windowOpen

- windowClose

- windowClosing

•windowActive

Class WindowEvent

Properties

property oldState

int

read-only

property newState

int

read-only

property oppositeWindow

php\swing\UIWindow

read-only

Font

php\swing\Font

Constants

constant PLAIN

constant BOLD

constant ITALIC

Properties

property family

string

read-only

Family name of font

property fontName

string

read-only

property name

string

read-only

property psName

string

read-only

PostScript name of font

property size

int

read-only**property size2D**

int

read-only**property style**

int

read-only**property italicAngle**

double

read-only**property attributes**

string[]

read-only**property numGlyphs**

int

read-only**Methods**

__construct (*\$name*, *\$style*, *\$size*)**Parameters**

- **\$name** – string
- **\$style** – int - - PLAIN, BOLD, ITALIC
- **\$size** – int

isBold()**Returns** bool**isItalic**()**Returns** bool**isPlain**()**Returns** bool**isTransformed**()**Returns** bool**getBaselineFor** (*\$symbol*)**Parameters**

- **\$symbol** – string - - one char

Returns int**canDisplay** (*\$symbol*)

Parameters

- **\$symbol** – string - - one char

Returns bool**canDisplayUpTo** (*\$string*)

Indicates whether or not this Font can display a specified String.

Parameters

- **\$string** – string

Returns int - an offset into \$string that points to the first character in \$string that this

Font cannot display; or -1 if this Font can display all characters in \$string.

static decode (*\$str*)

Decode font by using a specified string

Parameters

- **\$str** – string

Returns *php\swing\Font***static create** (*\$source*, *\$trueType* = true)

Create new font by using Stream or File

throws *php\io\IOException***throws** *php\swing\IOException***Parameters**

- **\$source** – string, *php\io\Stream*
- **\$trueType** – bool

Returns *php\swing\Font***static get** (*\$name*)

Get font by name

Parameters

- **\$name** – string

Returns *php\swing\Font*, null - return null if not exists

Graphics

php\swing\Graphics

Class Graphics

Properties

property color*php\swing\Color*

Foreground color

property font*php\swing\Font*

Font of text

Methods

__construct ()**private****getTextWidth** (\$str)

Return width of str for drawText + current font

Parameters

- **\$str** –

Returns int**getTextHeight** ()

Return height of one line text with current font

Returns int**setPaintMode** ()**drawLine** (\$x1, \$y1, \$x2, \$y2)

Draw line

Parameters

- **\$x1** – int
- **\$y1** – int
- **\$x2** – int
- **\$y2** – int

drawRect (\$x, \$y, \$width, \$height)

Draw rect

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

fillRect (\$x, \$y, \$width, \$height)**Parameters**

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

draw3DRect (\$x, \$y, \$width, \$height, \$raised)

Draws a 3-D highlighted outline of the specified rectangle.

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int
- **\$raised** –

fill3DRect (*\$x*, *\$y*, *\$width*, *\$height*, *\$raised*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int
- **\$raised** –

drawOval (*\$x*, *\$y*, *\$width*, *\$height*)

Draw oval

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

fillOval (*\$x*, *\$y*, *\$width*, *\$height*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

drawArc (*\$x*, *\$y*, *\$width*, *\$height*, *\$startAngle*, *\$arcAngle*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int
- **\$startAngle** – int
- **\$arcAngle** – int

fillArc (*\$x*, *\$y*, *\$width*, *\$height*, *\$startAngle*, *\$arcAngle*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int
- **\$startAngle** – int
- **\$arcAngle** – int

drawPolygon (*\$xy*)

Parameters

- **\$xy** – array - - [[x1, y1], [x2, y2], ...]

fillPolygon (*\$xy*)

Parameters

- **\$xy** – array - - [[x1, y1], [x2, y2], ...]

drawPolyline (*\$xy*)

Parameters

- **\$xy** – array - - [[x1, y1], [x2, y2], ...]

drawImage (*\$image*, *\$x* = 0, *\$y* = 0, *\$newWidth* = null, *\$newHeight* = null)

Parameters

- **\$image** – *php\swing\Image*
- **\$x** – int
- **\$y** – int
- **\$newWidth** – null, int
- **\$newHeight** – null, int

drawText (*\$text*, *\$x*, *\$y*)

Parameters

- **\$text** – string
- **\$x** – int
- **\$y** – int

clipRect (*\$x*, *\$y*, *\$width*, *\$height*)

Intersects the current clip with the specified rectangle.

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

clearRect (*\$x*, *\$y*, *\$width*, *\$height*)

Clears the specified rectangle by filling it with the background color of the current drawing surface.

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int

setXORMode (*\$color*)

Parameters

- **\$color** – *php\swing\Color*, int, array

translate (*\$x*, *\$y*)

Translates the origin of the graphics context to the point (x, y) in the current coordinate system.

Parameters

- **\$x** – int
- **\$y** – int

copyArea (*\$x*, *\$y*, *\$width*, *\$height*, *\$dx*, *\$dy*)

Copies an area of the component by a distance specified by \$dx and \$dy

Parameters

- **\$x** – int
- **\$y** – int
- **\$width** – int
- **\$height** – int
- **\$dx** – int
- **\$dy** – int

create (*\$x = null*, *\$y = null*, *\$w = null*, *\$h = null*)

Create new copy Graphics from this

Parameters

- **\$x** – int, null
- **\$y** – int, null
- **\$w** – int, null
- **\$h** – int, null

Returns *php\swing\Graphics*

dispose ()

Image

php\swing\Image

Constants

constant **TYPE_INT_RGB**

constant **TYPE_INT_ARGB**

```
constant TYPE_INT_ARGB_PRE
constant TYPE_INT_BGR
constant TYPE_3BYTE_BGR
constant TYPE_4BYTE_ABGR
constant TYPE_4BYTE_ABGR_PRE
constant TYPE_USHORT_565_RGB
constant TYPE_USHORT_555_RGB
constant TYPE_BYTE_GRAY
constant TYPE_USHORT_GRAY
constant TYPE_BYTE_BINARY
constant TYPE_BYTE_INDEXED
```

Properties

```
property type
    int
    read-only
property width
    int
    read-only
property height
    int
    read-only
```

Methods

```
__construct ($width, $height, $type = ::)
```

Parameters

- **\$width** – int
- **\$height** – int
- **\$type** – int

```
getSubimage ($x, $y, $w, $h)
```

Parameters

- **\$x** – int
- **\$y** – int
- **\$w** – int
- **\$h** – int

Returns *php\swing\Image*

```
getRGB ($x, $y)
```

Parameters

- **\$x** – int
- **\$y** – int

Returns int

setRGB (*\$x*, *\$y*, *\$rgb*)

Parameters

- **\$x** – int
- **\$y** – int
- **\$rgb** – int - - color

getGraphics ()

Returns *php\swing\Graphics*

getProperty (*\$name*)

Parameters

- **\$name** –

Returns *php\lang\JavaObject*, null

static read (*\$stream*)

throws *php\io\IOException*

Parameters

- **\$stream** – *php\io\Stream*, *php\io\File*, string - - file path or stream

Returns *php\swing\Image*

static write (*\$image*, *\$format*, *\$stream*)

throws *php\io\IOException*

Parameters

- **\$image** – *php\swing\Image*
- **\$format** –
- **\$stream** – *php\io\Stream*, *php\io\File*, string - - file path or stream

Scope

php\swing\Scope

Class Scope

Methods

static getDefault

Returns *php\swing\Scope*

ScopeValue

php\swing\ScopeValue
final class

Methods

__construct ()
private

getValue ()
Returns mixed

setValue (\$value)
Parameters

- \$value –

bind (\$object, \$property)
Parameters

- \$object – object
- \$property – string

static of (\$name)
Parameters

- \$name – string

Returns *php\swing\ScopeValue*

SwingUtilities

php\swing\SwingUtilities
final class

Methods

__construct ()
private

static getScreenSize
Returns screen size as an array [width, height]
Returns array

static setExceptionHandler (\$handler)
Parameters

- \$handler – callable - (Exception|JavaObject \$exception)

static invokeLater (\$runner)
Parameters

- \$runner – callable

SwingWorker

`php\swing\SwingWorker`
abstract class

Methods

doInBackground()
abstract
protected
Returns mixed

get(\$timeout = -1)
throws *php\concurrent\TimeoutException*
throws *php\lang\InterruptedException*
Parameters
• **\$timeout** – int
Returns mixed

getProgress()
Returns int

setProgress(\$val)
protected
Parameters
• **\$val** – int

publish(\$values)
protected
Parameters
• **\$values** – array

process(\$values)
protected
Parameters
• **\$values** – array

isDone()
Returns bool

isCanceled()
Returns bool

getState()
Returns string PENDING, STARTED, DONE

cancel(\$mayInterruptIfRunning)
Parameters
• **\$mayInterruptIfRunning** – bool

run ()
execute ()

text

Style

php\swing\text\Style
Class TextStyle

Properties

property background
php\swing\Color
property foreground
php\swing\Color
property bold
bool

Timer

php\swing\Timer

Properties

property repeat
bool
property delay
int
Delay in milliseconds for repeats
property initDelay
int
Initial Delay in milliseconds for first trigger
property actionCommand
string
User data

Methods

__construct (*\$delay*, *\$callback*)
Parameters

- **\$delay** – int
- **\$callback** – callable

start ()

stop()

restart()

isRunning()

Returns bool

tree

TreeModel

php\swing\tree\TreeModel

Class TreeModel

Properties

property root

php\swing\tree\TreeNode

Methods

__construct (*\$root*, *\$askAllowsChildren* = *false*)

Parameters

- *\$root* – *php\swing\tree\TreeNode*
- *\$askAllowsChildren* – bool

nodeChanged (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

nodeStructureChanged (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

removeNodeFromParent (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

reload (*\$node* = *null*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

TreeNode

`php\swing\tree\TreeNode`

Class `TreeNode`

Properties

property index

int

read-only

property depth

int

read-only

property level

int

read-only

property allowsChildren

bool

property parent

php\swing\tree\TreeNode

property userData

mixed

Methods

__construct (*\$object = null*, *\$allowsChildren = true*)

Parameters

- **\$object** – mixed
- **\$allowsChildren** – bool

isRoot ()

Returns bool

isLeaf ()

Returns bool

getRoot ()

Returns *php\swing\tree\TreeNode*

isNodeChild (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

Returns bool

isNodeAncestor (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

isNodeDescendant (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

isNodeRelated (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

isNodeSibling (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

getNextNode ()

Returns *php\swing\tree\TreeNode*, null

getNextLeaf ()

Returns *php\swing\tree\TreeNode*, null

getNextSibling ()

Returns *php\swing\tree\TreeNode*, null

getPreviousNode ()

Returns *php\swing\tree\TreeNode*, null

getPreviousLeaf ()

Returns *php\swing\tree\TreeNode*, null

getPreviousSibling ()

Returns *php\swing\tree\TreeNode*, null

getFirstChild ()

Returns *php\swing\tree\TreeNode*, null

getFirstLeaf ()

Returns *php\swing\tree\TreeNode*, null

getLastChild ()

Returns *php\swing\tree\TreeNode*, null

getLastLeaf ()

Returns *php\swing\tree\TreeNode*, null

add (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

insert (*\$childIndex*, *\$node*)

Parameters

- **\$childIndex** – int
- **\$node** – *php\swing\tree\TreeNode*

insertAfter (*\$child*, *\$node*)

throws >

Parameters

- **\$child** – *php\swing\tree\TreeNode*
- **\$node** – *php\swing\tree\TreeNode*

insertBefore (*\$child*, *\$node*)

throws >

Parameters

- **\$child** – *php\swing\tree\TreeNode*
- **\$node** – *php\swing\tree\TreeNode*

remove (*\$child*)

Parameters

- **\$child** – *php\swing\tree\TreeNode*

removeByIndex (*\$childIndex*)

Parameters

- **\$childIndex** – int

removeAllChildren ()

removeFromParent ()

getIndex (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

Returns int

getChild (*\$index*)

Parameters

- **\$index** – int

Returns *php\swing\tree\TreeNode*, null

getChildCount ()

Returns int

duplicate ()

Returns *php\swing\tree\TreeNode*

UIAbstractIButton

php\swing\UIAbstractIButton
extends: *php\swing\UIContainer*

Children

- **class** *php\swing\UIButton*
- **class** *php\swing\UICheckbox*
- **class** *php\swing\UIMenuItem*
- **class** *php\swing\UIToggleButton*

Class UIAbstractIButton

Properties

property text

string

Text of button

property selected

bool

property verPosition

int

Direction

property horPosition

int

Direction

property verAlignment

int

Direction

property horAlignment

php\swing\intlstring

Direction

property iconTextGap

int

property borderPainted

bool

property focusPainted

bool

property rolloverEnabled

bool

property contentAreaFilled

bool

property **buttonGroup**
string

Methods

setIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string - - filename or Image

setDisabledIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string

setSelectedIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string

setPressedIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string

setRolloverIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string

setDisabledSelectedIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string

setRolloverSelectedIcon (*\$icon*)

Parameters

- *\$icon* – *php\swing\Image*, string

doClick (*\$pressTime* = 68)

Parameters

- **\$pressTime** – int - the time to “hold down” the button, in milliseconds

static getButtons (*\$buttonGroup*)

Parameters

- **\$buttonGroup** – string

Returns *php\swing\UIAbstractButton[]*

static getSelectedButtons (*\$buttonGroup*)

Parameters

- **\$buttonGroup** – string

Returns *php\swing\UIAbstractButton[]*

UIButton

php\swing\UIButton
extends: *php\swing\UIAbstractButton*
Class UIButton

UICheckbox

php\swing\UICheckbox
extends: *php\swing\UIAbstractButton*
Class UICheckbox

UICheckboxMenuItem

php\swing\UICheckboxMenuItem
extends: *php\swing\UIMenuItem*
Class UICheckboxMenuItem

UIColorChooser

php\swing\UIColorChooser
extends: *php\swing\UIContainer*
Class UIColorChooser

Properties

property color
php\swing\Color

property dragEnabled
bool

UICombobox

php\swing\UICombobox
extends: *php\swing\UIContainer*

Properties

property readOnly
bool

property popupVisible
bool

property lightweightPopup
bool

property itemCount
int

read-only

property selectedIndex
int

property maxRowCount
int

Methods

setItems (*\$items*)

Parameters

- **\$items** – array

getItem (*\$index*)

Parameters

- **\$index** – int

Returns string, null

addItem (*\$item*)

Parameters

- **\$item** – string

insertItem (*\$index*, *\$item*)

Parameters

- **\$index** – int
- **\$item** – string

removeItem (*\$index*)

Parameters

- **\$index** – int

removeAllItems ()

onCellRender (*\$callback = null*)

Parameters

- **\$callback** – callable - (UICombobox \$self, UILabel \$template, \$value, int \$index, bool isSelected, bool cellHasFocus)

UIContainer

php\swing\UIContainer

abstract class

extends: *php\swing\UIElement*

Children

- **abstract class** *php\swing\UITextElement*
- **abstract class** *php\swing\UIWindow*
- **class** *php\swing\UIAbstractButton*
- **class** *php\swing\UIColorChooser*
- **class** *php\swing\UICombobox*
- **class** *php\swing\UIDesktopPanel*
- **class** *php\swing\UIFileChooser*
- **class** *php\swing\UIImage*
- **class** *php\swing\UIInternalForm*
- **class** *php\swing\UILabel*
- **class** *php\swing\UIListbox*
- **class** *php\swing\UIMenuBar*
- **class** *php\swing\UIPanel*
- **class** *php\swing\UIPopupMenu*
- **class** *php\swing\UIProgress*
- **class** *php\swing\UIScrollPanel*
- **class** *php\swing\UISlider*
- **class** *php\swing\UITable*
- **class** *php\swing\UITabs*
- **class** *php\swing\UIToolBar*
- **class** *php\swing\UITree*
- **class** *php\swing\UIUnknown*

Methods

add (*\$component*, *\$index* = null, *\$constraints* = null)
Add child component

Parameters

- **\$component** – *php\swing\UIElement*
- **\$index** – null, int
- **\$constraints** – null, int

setLayout (*\$type*)

Parameters

- **\$type** – string - - absolute, grid, flow, grid-bag, border, card

remove (*\$component*)

Parameters

- **\$component** – *php\swing\UIElement*

removeByIndex (*\$index*)
throws >

Parameters

- **\$index** – int

removeAll ()

getComponentCount ()

Returns int

getComponent (*\$index*)
throws >

Parameters

- **\$index** – int

Returns *php\swing\UIElement*

getComponents ()

Returns *php\swing\UIElement[]*

getComponentByGroup (*\$group*)
Find first component by group

Parameters

- **\$group** – string

Returns *php\swing\UIElement*, NULL

getComponentsByGroup (*\$group*)
Find all components in group

Parameters

- **\$group** – string

Returns *php\swing\UIElement[]*

UIDesktopPanel

php\swing\UIDesktopPanel
extends: *php\swing\UIContainer*

Class UIDesktopPanel

UIDialog

php\swing\UIDialog
extends: *php\swing\UIWindow*

Class UIDialog

Constants

constant **PLAIN_MESSAGE**

constant **ERROR_MESSAGE**

constant **INFORMATION_MESSAGE**
constant **WARNING_MESSAGE**
constant **QUESTION_MESSAGE**
constant **DEFAULT_OPTION**
constant **OK_CANCEL_OPTION**
constant **YES_NO_CANCEL_OPTION**
constant **YES_NO_OPTION**
constant **YES_OPTION**
constant **NO_OPTION**
constant **CANCEL_OPTION**
constant **OK_OPTION**
constant **CLOSED_OPTION**

Properties

property **modal**
bool

property **modalType**
php\swing\string - modeless or document_modal or application_modal or toolkit_modal

Methods

__construct (*\$owner = null*)

Parameters

- **\$owner** – *php\swing\UIWindow*

showModal ()

static message (*\$text, \$title, \$type = ::*)
Show basic message

Parameters

- **\$text** – string
- **\$title** – string
- **\$type** – int

static confirm (*\$text, \$title, \$optionType = ::, \$type = ::*)

Parameters

- **\$text** – string
- **\$title** – string
- **\$optionType** – int
- **\$type** – int

Returns int

static input (*\$message*, *\$initialValue* = '')

Parameters

- **\$message** – string
- **\$initialValue** – string

Returns string

UEdit

php\swing\UEdit

extends: *php\swing\UITextElement*

Class UEdit

Properties

property columns

int

property scrollOffset

int

UEditorArea

php\swing\UEditorArea

extends: *php\swing\UITextElement*

Class UEditorArea

Properties

property contentType

string

Example: text/html or text/plain

UIElement

php\swing\UIElement

abstract class

Children

- **abstract class** *php\swing\UIContainer*

Properties

property uid

string

read-only

property group

string

property x

int

Position X

property y

int

Position Y

property w

int

Width

property h

int

Height

property size

array

Size (width and height), [int, int]

property preferredSize

php\swing\array [int, int]

property minSize

array

Min Size (width and height) [int, int]

property autosize

bool

property position

array

Position (x and y), [int, int]

property screenPosition

array

Screen Position (x and y), [int, int]

property absolutePosition

array

read-only

Absolute Position (x and y), [int, int]

property visible

bool

property enabled

bool

property focusable

bool

property align

php\swing\string - NONE, LEFT, RIGHT, TOP, BOTTOM, CLIENT

property anchors

array

[LEFT, TOP, RIGHT, BOTTOM]

property font

php\swing\Font

property border

php\swing\Border

property background

php\swing\Color

property foreground

php\swing\Color

property tooltipText

string

property doubleBuffered

bool

property opaque

bool

property ignoreRepaint

bool

property autoscrolls

bool

property uiClassId

string

read-only

property popupMenu

php\swing\UIPopupMenu

property cursor

string

property padding

php\swing\array [int, int, int, int]

property owner

php\swing\UIContainer

property parent

php\swing\UIContainer

property firstParent

php\swing\UIContainer

Methods

__construct ()

getGraphics ()

Get graphic canvas object

Returns *php\swing\Graphics*

on (*\$name*, *\$callback*, *\$group* = 'general')

Events - Mouse: click mousePress mouseRelease mouseEnter mouseExit mouseMove mouseDrag

Keyboard: keyDown keyUp keyPress

Focus: focus blur

Add callback for event

Parameters

- **\$name** –
 - * name of event
- **\$callback** – callable
- **\$group** – string

off (*\$name*, *\$group* = *NULL*)

Remove all event callbacks (if group == null), or only group

Parameters

- **\$name** –
- **\$group** – null, string

Returns bool

trigger (*\$name*)

Trigger callback by event name

Parameters

- **\$name** –

addAllowedEventType (*\$name*)
protected

Parameters

- **\$name** – string

setAction (*\$name*, *\$callback* = *null*)

Parameters

- **\$name** – string
- **\$callback** – callable

setInputKey (*\$keyString*, *\$actionName*)

Parameters

- **\$keyString** – string
- **\$actionName** – string

hasFocus ()

Returns bool

add (*\$component*, *\$index* = *null*)

Add child component

Parameters

- **\$component** – *php\swing\UIElement*
- **\$index** – null, int

getComponentAt (\$x, \$y)

Determines if this component or one of its immediate subcomponents contains the (x, y) location, and if so, returns the containing component. This method only looks one level deep.

Parameters

- **\$x** –
- **\$y** –

Returns *php\swing\UIElement*

printOne (\$canvas)

Prints this component.

Parameters

- **\$canvas** – *php\swing\Graphics*

printAll (\$canvas)

Prints this component and all of its subcomponents.

Parameters

- **\$canvas** – *php\swing\Graphics*

paintOne (\$canvas)

Paints this component.

Parameters

- **\$canvas** – *php\swing\Graphics*

paintAll (\$canvas)

Paints this component and all of its subcomponents.

Parameters

- **\$canvas** – *php\swing\Graphics*

updateUI ()

invalidate ()

repaint ()

revalidate ()

repaintRegion (\$x, \$y, \$w, \$h)

Parameters

- **\$x** – int
- **\$y** – int
- **\$w** – int
- **\$h** – int

grabFocus ()

getTextWidth (\$str)

Return width of str for drawText + current font

Parameters

- **\$str** –

Returns int**getTextHeight** ()

Return height of one line text with current font

Returns int**show** ()**hide** ()**toggle** ()**removeSelf** ()**static getByUid** (\$uid)

Get component by unique id

Parameters

- **\$uid** – string

Returns *php\swing\UIElement***UIFileChooser****php\swing\UIFileChooser****extends:** *php\swing\UIContainer*

Class UIFileChooser

Constants

constant FILES_ONLY**constant DIRECTORIES_ONLY****constant FILES_AND_DIRECTORIES****Properties**

property dialogTitle

string

property approveButtonText

string

property selectedFile*php\io\File***property selectedFiles**

php\swing\File[]

property multiSelection

bool

property selectionMode

int

Example: FILES_AND_DIRECTORIES, FILES_ONLY or DIRECTORIES_ONLY

property dragEnabled

bool

property fileHiding

bool

property controlButtonVisible

bool

property acceptAllFileFilterUsed

bool

Methods

showDialog (*\$approveButtonText*, *\$parent = null*)**Parameters**

- **\$approveButtonText** – string
- **\$parent** – *php\swing\UIWindow*

Returns bool**showSaveDialog** (*\$parent = null*)**Parameters**

- **\$parent** – *php\swing\UIWindow*

Returns bool**showOpenDialog** (*\$parent = null*)**Parameters**

- **\$parent** – *php\swing\UIWindow*

Returns bool**addChoosableFilter** (*\$filter*, *\$description*)**Parameters**

- **\$filter** – callable - (File \$file) -> bool
- **\$description** – string

addChoosableExtensions (*\$extensions*, *\$description*, *\$showDirectories = true*)**Parameters**

- **\$extensions** – array - ['jpg', 'gif', 'png', ... etc]
- **\$description** – string
- **\$showDirectories** – bool

resetChoosableFilters ()**isTraversable** (*\$file*)**Parameters**

- **\$file** – string, *php\io\File*

Returns bool

ensureFileIsVisible (*\$file*)

Parameters

- **\$file** – string, *php\io\File*

approveSelection ()

cancelSelection ()

changeToParentDirectory ()

onFileView (*\$field*, *\$callback* = null)

Parameters

- **\$field** – string - - name (string), description (string), icon (Image), traversable (bool)
- **\$callback** – callable - (File \$file)

UIForm

php\swing\UIForm

extends: *php\swing\UIWindow*

Constants

constant NOTHING_ON_CLOSE

constant HIDE_ON_CLOSE

constant DISPOSE_ON_CLOSE

constant EXIT_ON_CLOSE

Properties

property maximized

bool

Methods

setDefaultCloseOperation (*\$action*)

Parameters

- **\$action** – int

setIconImage (*\$image* = null)

Parameters

- **\$image** – *php\swing\Image*

UIImage

php\swing\UIImage

extends: *php\swing\UIContainer*

Class UIImage

Properties

property stretch

bool

property centered

bool

property proportional

bool

property smooth

bool

property mosaic

bool

Methods

setImage (*\$image*)

Parameters

- *\$image* – *php\swing\Image*

UIInternalForm

php\swing\UIInternalForm

extends: *php\swing\UIContainer*

Class UIInternalForm

Properties

property title

string

property selected

bool

property resizable

bool

Methods

setLayeredPanel (*\$panel*)

Parameters

- *\$panel* – *php\swing\UIDesktopPanel*

setContent (*\$content*)

Parameters

- **\$content** – *php\swing\UIContainer*

UILabel

php\swing\UILabel

extends: *php\swing\UIContainer*

Properties

property text

string

Text of label

property verPosition

int

Direction

property horPosition

int

Direction

property verAlignment

int

Direction

property horAlignment

php\swing\intlstring

Direction

property iconTextGap

int

Methods

setLabelFor (*\$component*)

Parameters

- **\$component** – *php\swing\UIElement*

setIcon (*\$icon*)

Parameters

- **\$icon** – *php\swing\Image*, string - - filename or Image

setDisabledIcon (*\$icon*)

Parameters

- **\$icon** – *php\swing\Image*, string

UListbox

php\swing\UListbox

extends: *php\swing\UIContainer*

Class UListbox

Properties

property multiple

bool

property selectedIndex

int

property selectedIndexes

int[]

property maxSelectionIndex

int

property minSelectionIndex

int

property visibleRowCount

int

property selectionBackground

php\swing\Color

property selectionForeground

php\swing\Color

property horScrollPolicy

php\swing|string - ALWAYS, HIDDEN, AUTO

property verScrollPolicy

php\swing|string - ALWAYS, HIDDEN, AUTO

property itemCount

int

read-only

Methods

setItems (*\$items*)

Parameters

- **\$items** – array

getItem (*\$index*)

Parameters

- **\$index** – int

Returns string, null

addItem (*\$item*)

Parameters

- **\$item** – string

insertItem (*\$index*, *\$item*)

Parameters

- **\$index** – int
- **\$item** – string

removeItem (*\$index*)

Parameters

- **\$index** – int

removeAllItems ()

onCellRender (*\$callback = null*)

Parameters

- **\$callback** – callable - (UIListbox \$self, UILabel \$template, \$value, int \$index, bool isSelected, bool cellHasFocus)

UIManager

php\swing\UIManager
final class

Methods

__construct ()
private

static setLookAndFeel (*\$name*)
Set look and feel globally

Parameters

- **\$name** –
– * name of theme

static getSystemLookAndFeel
Get name of OS native look and feel

Returns string

UIMenu

php\swing\UIMenu
extends: *php\swing\UIMenuItem*

Class UIMenu events: *select, deselect, cancel*

Properties

property itemCount
int

read-only

property delay
int

property popupVisible
bool

Methods

getItem (*\$pos*)

Parameters

- **\$pos** – int

Returns *php\swing\UIMenuItem*

addSeparator ()

setMenuPosition (*\$x*, *\$y*)

Parameters

- **\$x** – int
- **\$y** – int

isTopLevelMenu ()

Returns bool

UIMenuBar

php\swing\UIMenuBar
extends: *php\swing\UIContainer*

Class UIMenuBar

Properties

property borderPainted
bool

UIMenuItem

php\swing\UIMenuItem
extends: *php\swing\UIAbstractIButton*

Children

- **class** *php\swing\UICheckboxMenuItem*
- **class** *php\swing\UIMenu*

Class UIMenuItem

Properties

property **accelerator**

string

Examples: “control DELETE”, “alt shift X”, “alt shift released X”, “typed a”

UIPanel

php\swing\UIPanel

extends: *php\swing\UIContainer*

Class UIPanel

UIPopupMenu

php\swing\UIPopupMenu

extends: *php\swing\UIContainer*

Class UIPopupMenu

Events: open, close, cancel

Properties

property **borderPainted**

bool

property **lightweightPopup**

bool

property **label**

string

Methods

addSeparator()

setPopupSize() (*\$width*, *\$height*)

Parameters

- **\$width** – int
- **\$height** – int

setSelected() (*\$component*)

Parameters

- **\$component** – *php\swing\UIElement*

show() (*\$invoker*, *\$x*, *\$y*)

Parameters

- **\$invoker** – *php\swing\UIElement*
- **\$x** – int
- **\$y** – int

UIProgress

php\swing\UIProgress

extends: *php\swing\UIContainer*

Class UIProgress

Properties

property value

int

property max

int

property min

int

property text

string

property textPainted

bool

property borderPainted

bool

UIRadioButton

php\swing\UIRadioButton

extends: *php\swing\UIToggleButton*

Class UIRadioButton

UIReader

php\swing\UIReader

Class XmlUIReader

Properties

property useInternalForms

bool

Enables that the reader will create instances of UIInternalForm insteadof UIForm and UIDialog

Methods

read (*\$stream*)

Parameters

- *\$stream* – *php\io\Stream*, *php\io\File*, string

Returns *php\swing\UIElement*

onRead (*\$handle = NULL*)

Parameters

- **\$handle** – callable - (UIElement \$el, \$var)

onTranslate (*\$handle = NULL*)

Parameters

- **\$handle** – callable - (UIElement \$el, \$value) -> mixed

UIRichTextArea

php\swing\UIRichTextArea

extends: *php\swing\UITextElement*

Class UIRichTextArea

Properties

property logicalStyle

php\swing\text\Style

property horScrollPolicy

php\swing\string - ALWAYS, HIDDEN, AUTO

property verScrollPolicy

php\swing\string - ALWAYS, HIDDEN, AUTO

Methods

addStyle (*\$name, \$parent = null*)

Parameters

- **\$name** – string
- **\$parent** – *php\swing\text\Style*

Returns *php\swing\text\Style*

getStyle (*\$name*)

Parameters

- **\$name** – string

Returns *php\swing\text\Style*

appendText (*\$text, \$style*)

Parameters

- **\$text** – string
- **\$style** – *php\swing\text\Style*

UIScrollPanel

php\swing\UIScrollPanel

extends: *php\swing\UIContainer*

Class UIScrollPanel

Properties

property horScrollPolicy

php\swing\string - ALWAYS, HIDDEN, AUTO

property verScrollPolicy

php\swing\string - ALWAYS, HIDDEN, AUTO

UISlider

php\swing\UISlider

extends: *php\swing\UIContainer*

Class UISlider

Properties

property value

int

property min

int

property max

int

property extent

int

property valueIsAdjusting

bool

property inverted

bool

property paintLabels

bool

property paintTicks

bool

property paintTrack

bool

property snapToTicks

bool

property labelTable

string[]

property majorTickSpacing

int

property `minorTickSpacing`
int

UITable

php\swing\UITable
extends: *php\swing\UIContainer*
Class UITable

Properties

property `dragEnabled`
bool

property `selectionBackground`
php\swing\Color

property `selectionForeground`
php\swing\Color

property `gridColor`
php\swing\Color

property `editingColumn`
int

property `editingRow`
int

property `rowMargin`
int

Methods

setRowHeight (*\$height*, *\$row = null*)

Parameters

- **\$height** – int
- **\$row** – null, int

getRowHeight (*\$row = null*)

Parameters

- **\$row** – null, int

Returns int

setValueAt (*\$value*, *\$row*, *\$column*)

Parameters

- **\$value** – string, null
- **\$row** – int
- **\$column** – int

getValueAt (*\$row*, *\$column*)

Parameters

- **\$row** – int
- **\$column** – int

Returns string, null**columnAtPoint** (\$x, \$y)**Parameters**

- **\$x** – int
- **\$y** – int

Returns int**rowAtPoint** (\$x, \$y)**Parameters**

- **\$x** – int
- **\$y** – int

Returns int**editCellAt** (\$row, \$column)**Parameters**

- **\$row** – int
- **\$column** – int

Returns bool**addColumnSelectionInterval** (\$index0, \$index1)**Parameters**

- **\$index0** – int
- **\$index1** – int

addRowSelectionInterval (\$index0, \$index1)**Parameters**

- **\$index0** – int
- **\$index1** – int

getColumnName (\$column)**Parameters**

- **\$column** – int

Returns string**setEditingColumn** (\$column)**Parameters**

- **\$column** – int

setEditingRow (\$row)**Parameters**

- **\$row** – int

UITabs

php\swing\UITabs

extends: *php\swing\UIContainer*

Class UITabs

Events: change

Properties

property selectedIndex

int

property selectedComponent

php\swing\UIElement

property tabPlacement

php\swing|string - left, right, top, bottom

property tabCount

int

read-only

Methods

addTab (*\$title*, *\$component*, *\$icon* = null)

Parameters

- **\$title** – string
- **\$component** – *php\swing\UIElement*
- **\$icon** – *php\swing\Image*

getTitleAt (*\$index*)

Parameters

- **\$index** – int

Returns string

setTitleAt (*\$index*, *\$value*)

Parameters

- **\$index** – int
- **\$value** – string

getToolTipTextAt (*\$index*)

Parameters

- **\$index** – int

Returns string

setToolTipTextAt (*\$index*, *\$value*)

Parameters

- **\$index** – int
- **\$value** – string

getTabIconAt (*\$index*)**Parameters**

- **\$index** – int

Returns *php\swing\Image***setTabIconAt** (*\$index*, *\$image*)**Parameters**

- **\$index** – int
- **\$image** – *php\swing\Image*

getTabComponentAt (*\$index*)**Parameters**

- **\$index** – int

Returns *php\swing\UIElement***setTabComponentAt** (*\$index*, *\$component*)**Parameters**

- **\$index** – int
- **\$component** – *php\swing\UIElement*

removeTabAt (*\$index*)**Parameters**

- **\$index** – int

removeAll ()**indexOfPosition** (*\$x*, *\$y*)**Parameters**

- **\$x** – int
- **\$y** – int

Returns int**isEnabledAt** (*\$index*)**Parameters**

- **\$index** – int

Returns bool**setEnabledAt** (*\$index*, *\$enabled*)**Parameters**

- **\$index** – int
- **\$enabled** – bool

UITextArea

php\swing\UITextArea

extends: *php\swing\UITextElement*

Class UITextArea

Properties

property lineWrap

bool

property wrapStyleWord

bool

property rows

int

property columns

int

property tabSize

int

property lineCount

int

read-only

property horScrollPolicy

php\swing\string - ALWAYS, HIDDEN, AUTO

property verScrollPolicy

php\swing\string - ALWAYS, HIDDEN, AUTO

UITextElement

php\swing\UITextElement

abstract class

extends: *php\swing\UIContainer*

Children

- class *php\swing\UIEdit*
- class *php\swing\UIEditorArea*
- class *php\swing\UIRichTextArea*
- class *php\swing\UITextArea*

Properties

property text

string

property readOnly

bool

property caretPos
int

property caretColor
php\swing\Color

property selectedStart
int

property selectedEnd
int

property selectedText
int

read-only

property selectionColor
php\swing\Color

property selectionTextColor
php\swing\Color

property disabledTextColor
php\swing\Color

property margin
php\swing\array [top, left, bottom, right]
int[]

Methods

copy ()

cut ()

paste ()

select (\$selStart, \$selEnd)

Parameters

- **\$selStart** – int
- **\$selEnd** – int

selectAll ()

replaceSelection (\$content)

Parameters

- **\$content** – string

printDialog ()
A convenience print method that displays a print dialog, and then prints this element in interactive mode with no header or footer text. Note: this method blocks until printing is done.

Returns bool

UIToggleButton

php\swing\UIToggleButton
extends: *php\swing\UIAbstractButton*

Children

- class *php\swing\UIRadioButton*
- Class UIToggleButton

UIToolBar

php\swing\UIToolBar
extends: *php\swing\UIContainer*

Class UIToolBar

Properties

property vertical
bool

property borderPainted
bool

property rollover
bool

property floatable
bool

Methods

addSeparator (*\$w = null, \$h = null*)

Parameters

- **\$w** – null, int
- **\$h** – null, int

UITree

php\swing\UITree
extends: *php\swing\UIContainer*

Class UITree

Properties

property model
php\swing\tree\TreeModel

property root
php\swing\tree\TreeNode

property rootVisible
bool

property editable
bool

property visibleRowCount
int

property dragEnabled
bool

property expandsSelectedPaths
bool

property invokesStopCellEditing
bool

property scrollsOnExpand
bool

property maxSelectionRow
int

read-only

property minSelectionRow
int

read-only

property leadSelectionRow
int

read-only

property selectionRows
int[]

property rowCount
int

read-only

property selectionCount
int

read-only

property rowHeight
int

property selectedNode
php\swing\tree\TreeNode

property selectedNodes
php\swing\TreeNode[]

property editingNode
php\swing\tree\TreeNode

read-only

property horScrollPolicy
php\swing\string - ALWAYS, HIDDEN, AUTO

property verScrollPolicy
php\swing\string - ALWAYS, HIDDEN, AUTO

Methods

onCellRender (*\$renderer*)

Parameters

- **\$renderer** – callable

addSelectionNode (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

removeSelectionNode (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

expandNode (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

expandRow (*\$row*)

Parameters

- **\$row** – int

collapseNode (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

collapseRow (*\$row*)

Parameters

- **\$row** – int

expandNodeAll (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

collapseNodeAll (*\$node*)

Parameters

- **\$node** – *php\swing\tree\TreeNode*

isExpandedRow (*\$row*)

Parameters

- **\$row** – int

Returns bool

isExpandedNode (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

isSelected (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

isNodeEditable (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

isVisible (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

hasBeenExpanded (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

Returns bool

fireTreeExpanded (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

fireTreeCollapsed (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

fireTreeWillExpand (*\$node*)

throws >

Parameters

- *\$node* – *php\swing\tree\TreeNode*

fireTreeWillCollapse (*\$node*)

throws >

Parameters

- *\$node* – *php\swing\tree\TreeNode*

makeVisible (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

scrollToNode (*\$node*)

Parameters

- *\$node* – *php\swing\tree\TreeNode*

cancelEditing ()

clearSelection ()

UIUnknown

php\swing\UIUnknown

extends: *php\swing\UIContainer*

Class UIUnknown

UIWindow

php\swing\UIWindow

abstract class

extends: *php\swing\UIContainer*

Children

- class *php\swing\UIDialog*
- class *php\swing\UIForm*

Properties

property title

string

property opacity

php\swing\float - 0 .. 1

property alwaysOnTop

bool

property resizable

bool

property undecorated

bool

Methods

isActive ()

Returns bool

isAlwaysOnTopSupported ()

Returns bool

moveToCenter ()

time

Time

php\time\Time

Class Time, Immutable

Methods

__construct (*\$date*, *\$timezone* = *null*)

Parameters

- **\$date** – int - unix long timestamp (in millis)
- **\$timezone** – *php\time\TimeZone* - - if null then gets default timezone

getTime ()

Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this Time object.

Returns int

getTimeZone ()

Get timezone of the time object

Returns *php\time\TimeZone*

year ()

Get the current year

Returns int

month ()

Get the current month of the year, 1 - Jan, 12 - Dec

Returns int

week ()

Get week of year

Returns int

weekOfMonth ()

Get week of month

Returns int

day ()

Get day of year

Returns int

dayOfMonth ()

Get day of month

Returns int

dayOfWeek ()

Get day of week

Returns int

dayOfWeekInMonth ()

Returns int

hour ()

Get hour, indicating the hour of the morning or afternoon. hour() is used for the 12-hour clock (0 - 11). Noon and midnight are represented by 0, not by 12.

Returns int

hourOfDay ()

Get hour of the day

Returns int

minute ()

Get minute of the hour

Returns int

second ()

Get second of the minute

Returns int

millisecond ()

Get millisecond of the second

Returns int

compare (\$time)

Compares the time values

Returns the value 0 if the time represented by the argument is equal to the time represented by this Time; a value less than 0 if the time of this Time is before the time represented by the argument; and a value greater than 0 if the time of this Time is after the time represented by the argument.

Parameters

- **\$time** – *php\time\Time*

Returns int

withTimeZone (\$timeZone)

Parameters

- **\$timeZone** – *php\time\TimeZone*

Returns *php\time\Time*

add (\$args)

Get a new time + \$args

use negative values to minus

Parameters

- **\$args** – array - [millis, sec, min, hour, day, month, year]

Returns *php\time\Time*

replace (\$args)

Clones the current datetime and replaces some fields to new values \$args

Parameters

- **\$args** – array - [millis, sec, min, hour, day, month, year]

Returns *php\time\Time*

toString (*\$format*)

Format the current datetime to string with \$format

- G Era designator Text AD
- y Year Year 1996; 96
- M Month in year Month July; Jul; 07
- w Week in year Number 27
- W Week in month Number 2
- D Day in year Number 189
- d Day in month Number 10
- F Day of week in month Number 2
- E Day in week Text Tuesday; Tue
- a Am/pm marker Text PM
- H Hour in day (0-23) Number 0
- k Hour in day (1-24) Number 24
- K Hour in am/pm (0-11) Number 0
- h Hour in am/pm (1-12) Number 12
- m Minute in hour Number 30
- s Second in minute Number 55
- S Millisecond Number 978
- z Time zone General time zone Pacific Standard Time; PST; GMT-08:00
- Z Time zone RFC 822 time zone -0800

Parameters

- **\$format** – string - date time format

Returns string

__toString ()

Format the time to yyyy-MM-dd'T'HH:mm:ss

Returns string

__clone ()

private

static now (*\$timeZone = null*)

Returns now time object (date + time)

Parameters

- **\$timeZone** – *php\time\TimeZone*

Returns *php\time\Time*

static today (*\$timeZone = null*)

Returns today date (without time)

Parameters

- **\$timeZone** – *php\time\TimeZone*

Returns *php\time\Time*

static of (*\$args*, *\$timeZone* = *null*)

Create a new time by using the *\$args* arrays that can contain the *sec*, *min*, *hour* and other keys:

```
$time = Time::of(['year' => 2013, 'month' => 1, 'day' => 1]) // 01 Jan.
↪2013
```

Parameters

- **\$args** – array - [*millis*, *sec*, *min*, *hour*, *day*, *month*, *year*]
- **\$timeZone** – *php\time\TimeZone* - if *null* then it uses the default timezone

Returns *php\time\Time*

static seconds

Returns the current time in seconds (like the *millis()* method only in seconds)

Returns *int*

static millis

Returns the current time in milliseconds. Note that while the unit of time of the return value is a millisecond, the granularity of the value depends on the underlying operating system and may be larger. For example, many operating systems measure time in units of tens of milliseconds.

Returns *int*

static nanos

Returns the current value of the running Java Virtual Machine's high-resolution time source, in nanoseconds.

This method can only be used to measure elapsed time and is not related to any other notion of system or wall-clock time. The value returned represents nanoseconds since some fixed but arbitrary *origin* time (perhaps in the future, so values may be negative). The same origin is used by all invocations of this method in an instance of a Java virtual machine; other virtual machine instances are likely to use a different origin

Returns *int*

TimeFormat

php\time\TimeFormat

Class TimeFormat, Immutable

Methods

__construct (*\$format*, *\$locale* = *null*, *\$formatSymbols* = *null*)

Parameters

- **\$format** – string
- **\$locale** – *php\uti\Locale* - if *null* then it uses the default locale
- **\$formatSymbols** – array - [*months* => [...], *short_months*, *eras*, *weekdays*, *short_weekdays*, *local_pattern_chars*]

format (*\$time*)

Parameters

- **\$time** – *php\time\Time*

Returns string**parse** (*\$string*, *\$timeZone* = null)**Parameters**

- **\$string** – string
- **\$timeZone** – *php\time\TimeZone*

Returns *php\time\Time*, null if parse error then returns null**__clone** ()**private****TimeZone****php\time\TimeZone**

Class TimeZone, Immutable

Methods

__construct (*\$rawOffset*, *\$ID*, *\$options* = null)**Parameters**

- **\$rawOffset** – int
- **\$ID** – string
- **\$options** – array

Returns *php\time\TimeZone***getId** ()

Get id of the timezone

Returns string**getRawOffset** ()

Get raw offset of the timezone

Returns string**__clone** ()**private****static UTC**

Returns UTC Time zone

Returns *php\time\TimeZone***static of** (*\$ID*)**Parameters**

- **\$ID** – string - code of timezone, e.g.: 'UTC'

Returns *php\time\TimeZone*

static **setDefault** (*\$zone*, *\$globally* = *false*)

Set default time zone for Time objects, by default - the default timezone is UTC

Parameters

- **\$zone** – *php\time\TimeZone*
- **\$globally** – bool

static **getDefault** (*\$globally* = *false*)

Get default timezone

Parameters

- **\$globally** – bool - if *false* - only for the current environment

Returns *php\time\TimeZone*

static **getAvailableIDs** (*\$rawOffset* = *null*)

Returns all available ids of timezones

Parameters

- **\$rawOffset** – int, null

Returns string[]

util

Flow

php\util\Flow

implements: Iterator

A special class to work with arrays and iterators under flows. Flows are used for the lazy array/iterator operations, to save the RAM memory.

Class Flow, Immutable

Methods

__construct (*\$collection*)

Create a new flow, you can also use *of()* method

Parameters

- **\$collection** – array, Iterator

withKeys ()

Enables to save keys for the next operation

Returns *php\util\Flow*

onlyKeys (*\$keys*, *\$ignoreCase* = *false*)

Parameters

- **\$keys** – array, Traversable
- **\$ignoreCase** – bool

Returns *php\util\Flow*

append (*\$collection*)

Appends a new collection to the current flow, do not remember that you can pass a flow to this method

Parameters

- **\$collection** – array, Iterator

Returns *php\util\Flow*

find (*\$filter = null*)

Finds elements by using the \$filter callback, elements - for each iteration that returns `true`

Parameters

- **\$filter** – callable

Returns *php\util\Flow*

findOne (*\$filter = null*)

Finds the first element by using the \$filter callback, when \$filter will return the first `true`

Parameters

- **\$filter** – callable

Returns mixed

group (*\$callback*)**Parameters**

- **\$callback** – callable

Returns *php\util\Flow*

each (*\$callback*)

Iterates elements. It will break if \$callback returns `false` strongly

Parameters

- **\$callback** – callable - (\$el[, \$key]): bool

Returns int - iteration count

eachSlice (*\$sliceSize, \$callback, \$withKeys = false*)

Iterates elements as slices (that are passing as arrays to \$callback). It will break if \$callback returns `false` strongly

Parameters

- **\$sliceSize** – int
- **\$callback** – callable - (array \$items): bool
- **\$withKeys** – bool

Returns int - slice iteration count

map (*\$callback*)

Iterates elements and returns a new flow of the result Example:

```
$newFlow = Flow::of([1,2,3])->map(function($el){ return $el * 10 });
```

// the new flow will contain 10, 20 and 30

Parameters

- **\$callback** – callable - (\$el[, \$key])

Returns *php\uti\Flow*

keys ()

Create a new flow by using the keys of the current flow

Returns *php\uti\Flow*

skip (\$n)

Skips \$n elements in the current collection

Parameters

- **\$n** – int - skip count

Returns *php\uti\Flow*

limit (\$count)

Limits collection with \$count

Parameters

- **\$count** – int - count of limit

Returns *php\uti\Flow*

reduce (\$callback)

Iterates elements and gets a result of this operation It can be used for calculate some results, for example:

```
// calculates a sum of elements
```

```
$sum = ..->reduce(function($result, $el){ $result = $result + $el });
```

Parameters

- **\$callback** – callable - (\$result, \$el[, \$key])

Returns int

sort (\$comparator = null)

Sort the last result of the flow, also see: `php\lib\items::sort()`

Note: use the `withKeys()` method to save keys

Parameters

- **\$comparator** – callable - (\$o1, \$o2) -> int, where -1 smaller, 0 equal, 1 greater

Returns array

sortByKeys (\$comparator = null)

The same method as `sort()` only based on keys instead of values

Note: use the `withKeys()` method to save keys

Parameters

- **\$comparator** – callable - (\$key1, \$key2) -> int

Returns array

toArray()

Convert elements to an array

Note: use the `withKeys()` method to save keys

Returns array

toString(\$separator)

Join elements to a string similar to `implode()` in PHP

Parameters

- **\$separator** – string

Returns string

count()

Returns int

current()

Returns mixed

next()

Returns void

key()

Returns mixed

valid()

Returns bool

rewind()

Returns void

__clone()

private

static of(\$collection)

Creates a new flow for an array of Iterator

Parameters

- **\$collection** – array, Traversable

Returns *php\util\Flow*

static ofRange(\$from, \$to, \$step = 1)

Creates a new flow for a number range

Parameters

- **\$from** – int
- **\$to** – int
- **\$step** – int

Returns *php\util\Flow*

static ofString (*\$string*, *\$chunkSize* = 1)
Creates a new flow for the string

Parameters

- **\$string** – string
- **\$chunkSize** – int - how many characters to combine for one item ?

Returns *php\util\Flow*

static ofStream (*\$stream*, *\$chunkSize* = 1)
Creates a new flow for the Stream object

Parameters

- **\$stream** – *php\io\Stream* - stream object
- **\$chunkSize** – int - size for `Stream.read($size)` method

Returns *php\util\Flow*

Locale

php\util\Locale

Class Locale, Immutable

Methods

__construct (*\$lang*, *\$country* = '', *\$variant* = '')

Parameters

- **\$lang** – string
- **\$country** – string
- **\$variant** – string

getLanguage ()

Returns string

getDisplayLanguage (*\$locale* = null)

Parameters

- **\$locale** – *php\util\Locale*

Returns string

getCountry ()

Returns string

getDisplayCountry (*\$locale* = null)

Parameters

- **\$locale** – *php\util\Locale*

Returns string

getVariant ()

Returns string

getDisplayVariant (*\$locale = null*)

Parameters

- *\$locale* – *php\uti\Locale*

Returns string

getISO3Country ()

Returns string

getISO3Language ()

Returns string

__toString ()

Returns a string representation of this Locale object, consisting of language, country, variant, script, and extensions as below:

language + “_” + country + “_” + (variant + “_#” | “#”) + script + “-” + extensions

Returns string

__clone ()

private

static ENGLISH

Returns *php\uti\Locale*

static US

Returns *php\uti\Locale*

static UK

Returns *php\uti\Locale*

static CANADA

Returns *php\uti\Locale*

static CANADA_FRENCH

Returns *php\uti\Locale*

static FRENCH

Returns *php\uti\Locale*

static FRANCE

Returns *php\uti\Locale*

static ITALIAN

Returns *php\uti\Locale*

static ITALY

Returns *php\uti\Locale*

static GERMAN

Returns *php\uti\Locale*

static GERMANY

Returns *php\uti\Locale*

static JAPAN

Returns *php\uti\Locale*

static JAPANESE

Returns *php\uti\Locale*

static KOREA

Returns *php\uti\Locale*

static KOREAN

Returns *php\uti\Locale*

static CHINA

Returns *php\uti\Locale*

static CHINESE

Returns *php\uti\Locale*

static TAIWAN

Returns *php\uti\Locale*

static RUSSIAN

Returns *php\uti\Locale*

static RUSSIA

Returns *php\uti\Locale*

static ROOT

Returns *php\uti\Locale*

static getDefault (*\$globally = false*)

Get default locale (if *globally = false* - only for the current environment)

Parameters

- **\$globally** – bool

Returns *php\uti\Locale*

static setDefault (*\$locale, \$globally = false*)

Set default locale

Parameters

- **\$locale** – *php\uti\Locale*
- **\$globally** – bool - if *false* - only for the current environment

static getAvailableLocales

Returns an array of all installed locales. The returned array represents the union of locales supported by the Java runtime environment

Returns *php\uti\Locale[]* An array of installed locales.

Promise

php\util\Promise

Methods

__construct ()

private

static create (*\$initialValue*, *\$callback*)

Parameters

- **\$initialValue** – mixed - (optional)
- **\$callback** – callable - (\$value) - assign callback

Returns mixed

Regex

php\util\Regex

implements: Iterator

<http://www.regular-expressions.info/java.html>

Class Regex, Immutable

Constants

constant CANON_EQ

constant CASE_INSENSITIVE

constant UNICODE_CASE

constant COMMENTS

constant DOTALL

constant LITERAL

constant MULTILINE

constant UNIX_LINES

Methods

__construct ()

private

getPattern ()

Get the current pattern

Returns string

getFlags ()

Get the current flags

Returns int

static of (*\$pattern*, *\$flag* = 0)

Creates a new Regex of regex with \$string and \$flag

throws *php\util\RegexException*

Parameters

- **\$pattern** – string - regular expression
- **\$flag** – int - `Regex::CASE_INSENSITIVE` and other constants

Returns *php\util\Regex*

matches ()

Attempts to match the entire region against the pattern.

Returns bool

find (*\$start* = null)

Resets this matcher and then attempts to find the next subsequence of the input sequence that matches the pattern, starting at the specified index.

throws *php\util\RegexException*

Parameters

- **\$start** – int, null

Returns bool

replace (*\$replacement*)

Replaces every subsequence of the input sequence that matches the pattern with the given replacement string.

This method first resets this matcher. It then scans the input sequence looking for matches of the pattern. Characters that are not part of any match are appended directly to the result string; each match is replaced in the result by the replacement string.

throws *php\util\RegexException*

Parameters

- **\$replacement** – string

Returns string

replaceFirst (*\$replacement*)

Replaces the first subsequence of the input sequence that matches the pattern with the given replacement string.

throws *php\util\RegexException*

Parameters

- **\$replacement** – string

Returns string

replaceWithCallback (*\$callback*)

throws *php\util\RegexException*

Parameters

- **\$callback** – callable - (Regex \$pattern) -> string

Returns string

with (*\$string*)

Duplicates this pattern with a new *\$string*

Parameters

- **\$string** – string

Returns *php\util\Regex*

withFlags (*\$flags*)

Clone this object with the new *\$flags*

Parameters

- **\$flags** – int

Returns *php\util\Regex*

group (*\$group = null*)

Returns the input subsequence captured by the given group during the previous match operation.

throws *php\util\RegexException*

Parameters

- **\$group** – null, int

Returns string

getGroupCount ()

Returns the number of capturing groups in this matcher's pattern.

Returns int

start (*\$group = null*)

Returns the start index of the previous match.

throws *php\util\RegexException*

Parameters

- **\$group** – null, int

Returns int

end (*\$group = null*)

Returns the offset after the last character matched.

throws *php\util\RegexException*

Parameters

- **\$group** – null, int

Returns int

hitEnd ()

Returns true if the end of input was hit by the search engine in the last match operation performed by this matcher.

Returns bool

requireEnd ()

Returns true if more input could change a positive match into a negative one.

If this method returns true, and a match was found, then more input could cause the match to be lost. If this method returns false and a match was found, then more input might change the match but the match won't be lost. If a match was not found, then `requireEnd` has no meaning.

Returns bool

lookingAt()

Attempts to match the input sequence, starting at the beginning of the region, against the pattern.

Returns bool

region(\$start, \$end)

Sets the limits of this matcher's region. The region is the part of the input sequence that will be searched to find a match. Invoking this method resets the matcher, and then sets the region to start at the index specified by the \$start parameter and end at the index specified by the \$end parameter.

throws *php\util\RegexException*

Parameters

- **\$start** – int
- **\$end** – int

Returns *php\util\Regex*

regionStart()

Reports the start index of this matcher's region. The searches this matcher conducts are limited to finding matches within `regionStart()` (inclusive) and `regionEnd()` (exclusive).

Returns int

regionEnd()

Reports the end index (exclusive) of this matcher's region. The searches this matcher conducts are limited to finding matches within `regionStart()` (inclusive) and `regionEnd()` (exclusive).

Returns int

reset(\$string = null)

Resets this matcher.

Resetting a matcher discards all of its explicit state information and sets its append position to zero. The matcher's region is set to the default region, which is its entire character sequence. The anchoring and transparency of this matcher's region boundaries are unaffected.

Parameters

- **\$string** – null, string - The new input character sequence

Returns *php\util\Regex*

current()

Returns null, string

next()

Returns null, string

key()

Returns int

valid()

Returns bool

rewind()

Returns bool

__clone()
private

static match (*\$pattern*, *\$string*)

Tells whether or not this string matches the given regular expression. See also `java.lang.String.matches()`

throws *php\util\RegexException*

Parameters

- **\$pattern** – string - regular expression
- **\$string** – string

Returns `bool`

static split (*\$pattern*, *\$string*, *\$limit* = 0)

Splits this string around matches of the given regular expression. See also `java.lang.String.split()`

throws *php\util\RegexException*

Parameters

- **\$pattern** – string - the delimiting regular expression
- **\$string** – string
- **\$limit** – int - the result threshold

Returns array the array of strings computed by splitting this string around matches of the given regular expression

static quote (*\$string*)

Returns a literal pattern `String` for the specified `String`.

This method produces a `String` that can be used to create a `Regex` that would match the string `$string` as if it were a literal pattern. Metacharacters or escape sequences in the input sequence will be given no special meaning.

Parameters

- **\$string** – string - The string to be literalized

Returns `string` A literal string replacement

static quoteReplacement (*\$string*)

Returns a literal replacement `String` for the specified `String`.

This method produces a `String` that will work as a literal replacement `$string` in the `replaceWithCallback()` method of the `php\util\Regex` class. The `String` produced will match the sequence of characters in `$string` treated as a literal sequence. Slashes (‘’) and dollar signs (‘\$’) will be given no special meaning.

Parameters

- **\$string** – string

Returns `string`

RegexException

php\util\RegexException

extends: *php\lang\JavaException*

Class RegexException

Scanner

`php\util\Scanner`

implements: Iterator

A simple text scanner which can parse primitive types and strings using regular expressions.

Class Scanner

Methods

__construct (*\$source*, *\$charset* = *null*)

throws *php\lang\IllegalArgumentException* if *\$charset* is invalid

Parameters

- **\$source** – string, *php\io\Stream*
- **\$charset** – string, null - e.g.: UTF-8, windows-1251, etc., only for Stream objects

hasNext (*\$pattern* = *null*)

Parameters

- **\$pattern** – *php\util\Regex*

Returns bool

next (*\$pattern* = *null*)

Parameters

- **\$pattern** – *php\util\Regex*

Returns string, null null if doesn't has the next pattern

nextLine ()

Returns string, null null if doesn't has the next line

hasNextLine ()

Returns bool

nextInt (*\$radix* = *null*)

Parameters

- **\$radix** – null, int - if null then uses the default radix

Returns int, null null if doesn't has the next int

hasNextInt (*\$radix* = *null*)

Parameters

- **\$radix** – null, int - if null then uses the default radix

Returns bool

nextDouble ()

Returns float, null null if does not has the next double

hasNextDouble()

Returns bool

skip(\$pattern)

Parameters

- **\$pattern** – *php\uti\Regex*

Returns bool true on success, false on fail

useDelimiter(\$delimiter)

Parameters

- **\$delimiter** – *php\uti\Regex*

Returns *php\uti\Scanner*

useLocale(\$locale)

Parameters

- **\$locale** – *php\uti\Locale*

Returns *php\uti\Scanner*

useRadix(\$value)

Parameters

- **\$value** – int

Returns *php\uti\Scanner*

getIOException()

Get the last io exception (if does not occur then returns null)

Returns *php\io\IOException*, null

reset()

current()

Uses the result of the last called next() method

Returns string

key()

Returns int

valid()

Returns bool

rewind()

__clone()

private

Shared

php\util\Shared

Class to work with shared memory of Environments

Class Shared

Methods

__construct ()
private

static value (\$name, \$creator = null)
Get or create if does not exist and get a shared value

Parameters

- **\$name** – string
- **\$creator** – callable - returns init value

Returns *php\util\SharedValue*

static reset (\$name)
Removes the value by \$name.

Parameters

- **\$name** – php\util\String

Returns *php\util\SharedValue* removed value

static resetAll

SharedCollection

php\util\SharedCollection

abstract class

extends: *php\util\SharedMemory*

implements: Countable, Traversable

Children

- class *php\util\SharedMap*
- class *php\util\SharedQueue*
- class *php\util\SharedStack*

Methods

isEmpty ()
abstract

Returns bool

count ()
abstract

Returns int

clear ()
abstract

Remove all elements.

Returns void

SharedMap

`php\util\SharedMap`

extends: *php\util\SharedCollection*

Class SharedMap

Methods

__construct (*\$array*)

Parameters

- **\$array** – array, Traversable - (optional)

get (*\$key*, *\$default = null*)

Parameters

- **\$key** – string
- **\$default** – mixed

Returns mixed

getOrCreate (*\$key*, *\$createCallback*)

Parameters

- **\$key** – string
- **\$createCallback** – callable

Returns mixed

has (*\$key*)

Parameters

- **\$key** – string

Returns bool

count ()

Returns int

set (*\$key*, *\$value*, *\$override = true*)

Parameters

- **\$key** – string
- **\$value** – mixed
- **\$override** – bool

Returns mixed previous value

remove (*\$key*)

Parameters

- **\$key** – string

Returns mixed

clear ()

isEmpty()

Returns bool

SharedMemory

php\util\SharedMemory

abstract class

Children

- **abstract class** *php\util\SharedCollection*
- **class** *php\util\SharedValue*

Methods

synchronize (*\$callback*)

You can use a shared value as a mutex

Parameters

- **\$callback** – callable - (SharedValue \$this)

Returns mixed result of execution of \$callback

SharedQueue

php\util\SharedQueue

extends: *php\util\SharedCollection*

Class SharedQueue

Methods

__construct (*\$array*)

Parameters

- **\$array** – array, Traversable - (optional)

isEmpty()

Returns bool

count()

Returns int

clear()

Remove all elements.

Returns void

add (*\$value*)

Parameters

- **\$value** – mixed

Returns bool

peek()

Returns mixed

poll()

Retrieves and removes the head of this queue.

Returns mixed

SharedStack

php\util\SharedStack

extends: *php\util\SharedCollection*

Class SharedStack

Methods

__construct (\$array)

Parameters

- **\$array** – array, Traversable - (optional)

push (\$value)

Parameters

- **\$value** – mixed

Returns mixed peek value

pop()

Returns mixed peek value

peek()

Returns mixed

count()

Returns int

clear()

isEmpty()

Returns bool

SharedValue

php\util\SharedValue

extends: *php\util\SharedMemory*

Class SharedValue

Methods

__construct (\$value)

Parameters

- **\$value** – mixed - (optional)

get ()**Returns** mixed**set** (*\$value*, *\$override = true*)**Parameters**

- **\$value** – mixed
- **\$override** – bool

Returns mixed**remove** ()**Returns** mixed**isEmpty** ()**Returns** bool**getAndSet** (*\$updateCallback*)**Parameters**

- **\$updateCallback** – callable - (*\$oldValue*) returns a new value

Returns mixed**setAndGet** (*\$updateCallback*)**Parameters**

- **\$updateCallback** – callable - (*\$oldValue*) returns a new value

Returns mixed

webserver

WebRequest

php\webserver\WebRequest

Class WebRequest

Properties

property method

string

read-only**property scheme**

string

read-only**property pathInfo**

string

read-only

property queryString

string

read-only

property authType

string

read-only

property url

string

read-only

property port

int

read-only

property ip

string

read-only

property cookies

array

read-only

Array of arrays [name, value, path, domain, httpOnly, secure, maxAge, comment]

Methods

__construct (*\$parent*)

protected

Parameters

- **\$parent** – *php\webserver\WebRequest*

getBody ()

Returns string

getBodyStream ()

Returns *php\io\Stream*

getMethod ()

protected

Returns string

getPathInfo ()

protected

Returns string

getAuthType ()

protected

Returns string

getQueryString ()

protected

Returns string

getUrl()
protected

Returns string

getScheme()
protected

Returns string

getPort()
protected

Returns int

getIp()
protected

Returns string

static current

Returns *php\webserver\WebRequest*

WebResponse

php\webserver\WebResponse

Class WebResponse

Properties

property status
int

property contentType
string

property characterEncoding
string

property bufferSize
int

Methods

__construct() (*\$parent*)
protected

Parameters

- **\$parent** – *php\webserver\WebResponse*

setHeader() (*\$name*, *\$value*)

Parameters

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

getHeader (*\$name*)

Parameters

- **\$name** – string

Returns string

getHeaders (*\$name*)

Parameters

- **\$name** – string

Returns string[]

getHeaderNames ()

Returns string[]

addHeader (*\$name*, *\$value*)

Parameters

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

redirect (*\$location*, *\$httpStatus* = 301)

Parameters

- **\$location** – string
- **\$httpStatus** – int

encodeRedirectURL (*\$url*)

Parameters

- **\$url** – string

writeToBody (*\$content*)

Parameters

- **\$content** – string

setContentLength (*\$length*)

Parameters

- **\$length** – int

addCookie (*\$cookie*)

Parameters

- **\$cookie** – array - [name, value, maxAge, path, domain, httpOnly, secure, comment]

getStatus ()

protected

Returns int

setStatus (*\$status*)

protected

Parameters

- **\$status** – int

getContentType()
protected

Returns string

setContentType(\$contentType)
protected

Parameters

- **\$contentType** – string

getCharacterEncoding()
protected

Returns string

setCharacterEncoding(\$characterEncoding)
protected

Parameters

- **\$characterEncoding** – string

getBufferSize()
protected

Returns int

setBufferSize(\$bufferSize)
protected

Parameters

- **\$bufferSize** – int

static current

Returns *php\webserver\WebResponse*

WebServer

php\webserver\WebServer

Embedded http web server.

Class WebServer

Properties

property id
string

read-only

property port
int

property isolated
bool

property importAutoloaders
bool

property hotReload
bool

Methods

__construct (*\$onRequest*)

Parameters

- **\$onRequest** – callable

run ()

addStaticHandler (*\$handler*)

Parameters

- **\$handler** – array - [path, location, cache, cachePeriod, gzip]

Returns *php\webserver\WebServer*

getId ()

protected

Returns string

getPort ()

protected

Returns int

setPort (*\$port*)

protected

Parameters

- **\$port** – int

isIsolated ()

protected

Returns boolean

setIsolated (*\$isolated*)

protected

Parameters

- **\$isolated** – boolean

isImportAutoloaders ()

protected

Returns boolean

setImportAutoloaders (*\$importAutoloaders*)

protected

Parameters

- **\$importAutoloaders** – boolean

isHotReload ()

protected

Returns boolean

setHotReload (*\$hotReload*)
protected

Parameters

- **\$hotReload** – boolean

__clone ()
private

static current

Returns *php\webserver\WebServer*

xml

DomDocument

php\xml\DomDocument
abstract class

extends: *php\xml\DomElement*

Methods

getDocumentElement ()

Returns *php\xml\DomDocument*

getElementById (*\$id*)

Parameters

- **\$id** – string

Returns *php\xml\DomElement*

getInputEncoding ()

Returns string

getXmlEncoding ()

Returns string

getXmlVersion ()

Returns string

getXmlStandalone ()

Returns bool

setXmlStandalone (*\$value*)

Parameters

- **\$value** – bool

getStrictErrorChecking ()

Returns bool

setStrictErrorChecking (*\$value*)

Parameters

- **\$value** – bool

getDocumentURI ()

Returns string

setDocumentURI (\$value)

Parameters

- **\$value** – string

createElement (\$tagName, \$model)

Parameters

- **\$tagName** – string
- **\$model** – Traversable, array - (optional)

Returns *php\xml\DomElement*

createElementNS (\$namespaceURI, \$qualifiedName)

Parameters

- **\$namespaceURI** – string
- **\$qualifiedName** – string

Returns *php\xml\DomElement*

importNode (\$importedNode, \$deep)

Parameters

- **\$importedNode** – *php\xml\DomNode*
- **\$deep** – bool

Returns *php\xml\DomNode*

adoptNode (\$source)

Parameters

- **\$source** – *php\xml\DomNode*

Returns *php\xml\DomNode*

renameNode (\$node, \$namespaceURI, \$qualifiedName)

Parameters

- **\$node** – *php\xml\DomNode*
- **\$namespaceURI** – string
- **\$qualifiedName** – string

normalizeDocument ()

DomElement

`php\xml\DomElement`

abstract class

extends: *php\xml\DomNode*

Children

- **abstract class** *php\xml\DomDocument*

Methods

`__get ($name)`

Parameters

- **\$name** – string

Returns string Value of attribute by \$name

`__set ($name, $value)`

Set attribute value

Parameters

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

`__unset ($name)`

Remove attribute by name

Parameters

- **\$name** – string

`__isset ($name)`

Check attribute exists by name

Parameters

- **\$name** –

Returns bool

`getTagName ()`

Returns string

`getAttribute ($name)`

Parameters

- **\$name** – string

Returns string

`hasAttribute ($name)`

Parameters

- **\$name** – string

Returns bool

hasAttributeNS (*\$namespaceURI*, *\$localName*)

Parameters

- **\$namespaceURI** – string
- **\$localName** – string

Returns bool

setAttribute (*\$name*, *\$value*)

Parameters

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

setAttributes (*\$attributes*)

Parameters

- **\$attributes** – array, Traversable

removeAttribute (*\$name*)

Parameters

- **\$name** – string

getElementsByTagName (*\$name*)

Parameters

- **\$name** – string

Returns *php\xml\DomNodeList*

getElementsByTagNameNS (*\$namespaceURI*, *\$localName*)

Parameters

- **\$namespaceURI** – string
- **\$localName** – string

Returns *php\xml\DomNodeList*

getAttributeNS (*\$namespaceURI*, *\$localName*)

Parameters

- **\$namespaceURI** – string
- **\$localName** – string

Returns string

setAttributeNS (*\$namespaceURI*, *\$qualifiedName*, *\$value*)

Parameters

- **\$namespaceURI** – string
- **\$qualifiedName** – string
- **\$value** – string

removeAttributeNS (*\$namespaceURI*, *\$localName*)

Parameters

- **\$namespaceURI** – string
- **\$localName** – string

setIdAttribute (*\$name*, *\$isId*)

Parameters

- **\$name** – string
- **\$isId** – bool

setIdAttributeNS (*\$namespaceURI*, *\$localName*, *\$isId*)

Parameters

- **\$namespaceURI** – string
- **\$localName** – string
- **\$isId** – string

DomNode

php\xml\DomNode
abstract class

Children

- **abstract class** *php\xml\DomElement*

Methods

get (*\$xpathExpression*)

Parameters

- **\$xpathExpression** – string

Returns string

find (*\$xpathExpression*)

Parameters

- **\$xpathExpression** – string

Returns *php\xml\DomNode*

findAll (*\$xpathExpression*)

Parameters

- **\$xpathExpression** – string

Returns *php\xml\DomNodeList*

getBaseURI ()

Returns string

getNamespaceURI ()

Returns string

getLocalName()
Returns string

getNodeType()
Returns int

getNodeName()
Returns string

getNodeValue()
Returns string

getPrefix()
Returns string

getTextContent()
Returns string

getFirstChild()
Returns *php\xml\DomNode*

getLastChild()
Returns *php\xml\DomNode*

getNextSibling()
Returns *php\xml\DomNode*

getPreviousSibling()
Returns *php\xml\DomNode*

getParentNode()
Returns *php\xml\DomNode*

getOwnerDocument()
Returns *php\xml\DomDocument*

hasAttributes()
Returns bool

hasChildNodes()
Returns bool

isDefaultNamespace(\$namespace)
Parameters

- **\$namespace** – string

isEqualNode(\$node)
Parameters

- **\$node** – *php\xml\DomNode*

Returns bool

isSameNode(\$node)

Parameters

- **\$node** – *php\xml\DOMNode*

Returns bool**isSupported** (*\$feature*, *\$version*)**Parameters**

- **\$feature** – string
- **\$version** – string

Returns bool**lookupNamespaceURI** (*\$prefix*)**Parameters**

- **\$prefix** – string

Returns string**lookupPrefix** (*\$namespaceURI*)**Parameters**

- **\$namespaceURI** – string

Returns string**normalize** ()**setTextContent** (*\$content*)**Parameters**

- **\$content** – string

setPrefix (*\$prefix*)**Parameters**

- **\$prefix** – string

cloneNode (*\$deep*)**Parameters**

- **\$deep** – bool

appendChild (*\$node*)**Parameters**

- **\$node** – *php\xml\DOMNode*

Returns *php\xml*\$this**removeChild** (*\$node*)**Parameters**

- **\$node** – *php\xml\DOMNode*

Returns *php\xml*\$this**replaceChild** (*\$newNode*, *\$oldNode*)**Parameters**

- **\$newNode** – *php\xml\DomNode*
- **\$oldNode** – *php\xml\DomNode*

Returns *php\xml\this*

insertBefore (*\$newNode*, *\$refNode*)

Parameters

- **\$newNode** – *php\xml\DomNode*
- **\$refNode** – *php\xml\DomNode*

Returns *php\xml\this*

toModel ()

Returns *array*

DomNodeList

php\xml\DomNodeList

abstract class

implements: *ArrayAccess*, *Iterator*, *Countable*

XmlProcessor

php\xml\XmlProcessor

extends: *php\format\Processor*

Methods

format (*\$value*)

Parameters

- **\$value** – *php\xml\DomDocument*

Returns *string xml*

formatTo (*\$value*, *\$output*)

Parameters

- **\$value** – *php\xml\DomDocument*
- **\$output** – *php\io\Stream*

parse (*\$string*)

Parameters

- **\$string** – *php\io\Stream*, *string* - stream of string of xml

Returns *php\xml\DomDocument*

createDocument ()

Returns *php\xml\DomDocument*

JPHP is a new implementation for PHP which uses the Java VM. It supports many features of the PHP language (5.4+). How does it work? JPHP is a compiler like `javac`, it compiles php sources to JVM bytecode and then can execute the result on the Java VM.

Note: Supports: JDK 1.6+ (OpenJDK, Oracle), Linux, Windows, MacOS, etc.

Symbols

() (method), [2](#), [3](#), [5–106](#), [109](#), [113–127](#), [129](#), [131–134](#), [137–150](#), [152–155](#), [157](#), [158](#), [160–199](#)

(property), [36](#), [37](#), [53](#), [87](#), [96](#), [97](#), [99](#), [105](#), [107–114](#), [119](#), [123–125](#), [128](#), [130](#), [131](#), [134–137](#), [140–145](#), [147–152](#), [154](#), [156–160](#), [162](#), [186–188](#), [190](#)

B

BaseException (class), [1](#)

BOLD (global constant), [112](#)

C

CANCEL_OPTION (global constant), [134](#)

CANON_EQ (global constant), [175](#)

CASE_INSENSITIVE (global constant), [175](#)

CLOSED_OPTION (global constant), [134](#)

COMMENTS (global constant), [175](#)

CONCURRENT (global constant), [64](#)

D

DEFAULT_OPTION (global constant), [134](#)

DESERIALIZE_AS_ARRAYS (global constant), [10](#)

DIRECTORIES_ONLY (global constant), [140](#)

DIRECTORY_SEPARATOR (global constant), [46](#)

DISPOSE_ON_CLOSE (global constant), [142](#)

DOTALL (global constant), [175](#)

E

EngineException (class), [1](#)

ERROR_MESSAGE (global constant), [133](#)

EXIT_ON_CLOSE (global constant), [142](#)

F

FILES_AND_DIRECTORIES (global constant), [140](#)

FILES_ONLY (global constant), [140](#)

G

getErrorType() (EngineException method), [1](#)

H

HIDE_ON_CLOSE (global constant), [142](#)

HOT_RELOAD (global constant), [64](#)

I

INFORMATION_MESSAGE (global constant), [133](#)

ITALIC (global constant), [112](#)

L

LENGTH_LONG (global constant), [7](#)

LENGTH_SHORT (global constant), [7](#)

LITERAL (global constant), [175](#)

LOG_DEBUG (global constant), [12](#)

LOG_ERROR (global constant), [12](#)

LOG_INFO (global constant), [12](#)

LOG_NONE (global constant), [12](#)

M

MAX_PRIORITY (global constant), [75](#)

METHOD_GET (global constant), [55](#)

METHOD_POST (global constant), [55](#)

MIN_PRIORITY (global constant), [75](#)

MULTILINE (global constant), [175](#)

N

NO_OPTION (global constant), [134](#)

NORM_PRIORITY (global constant), [75](#)

NOTHING_ON_CLOSE (global constant), [142](#)

O

OK_CANCEL_OPTION (global constant), [134](#)

OK_OPTION (global constant), [134](#)

P

ParseException (class), [2](#)

PATH_NAME_CASE_INSENSITIVE (global constant), [46](#)

PATH_SEPARATOR (global constant), [46](#)

PLAIN (global constant), [112](#)

PLAIN_MESSAGE (global constant), [133](#)

Q

QUESTION_MESSAGE (global constant), [134](#)

S

SERIALIZE_PRETTY_PRINT (global constant), [10](#)

T

TRANSACTION_NONE (global constant), [99](#)

TRANSACTION_READ_COMMITTED (global constant), [99](#)

TRANSACTION_READ_UNCOMMITTED (global constant), [99](#)

TRANSACTION_REPEATABLE_READ (global constant), [99](#)

TRANSACTION_SERIALIZABLE (global constant), [99](#)

TYPE_3BYTE_BGR (global constant), [119](#)

TYPE_4BYTE_ABGR (global constant), [119](#)

TYPE_4BYTE_ABGR_PRE (global constant), [119](#)

TYPE_BYTE_BINARY (global constant), [119](#)

TYPE_BYTE_GRAY (global constant), [119](#)

TYPE_BYTE_INDEXED (global constant), [119](#)

TYPE_CLASS_DATETIME (global constant), [4](#)

TYPE_CLASS_NUMBER (global constant), [4](#)

TYPE_CLASS_PHONE (global constant), [4](#)

TYPE_CLASS_TEXT (global constant), [4](#)

TYPE_DATETIME_VARIATION_DATE (global constant), [5](#)

TYPE_DATETIME_VARIATION_NORMAL (global constant), [5](#)

TYPE_DATETIME_VARIATION_TIME (global constant), [5](#)

TYPE_INT_ARGB (global constant), [118](#)

TYPE_INT_ARGB_PRE (global constant), [118](#)

TYPE_INT_BGR (global constant), [119](#)

TYPE_INT_RGB (global constant), [118](#)

TYPE_MASK_CLASS (global constant), [4](#)

TYPE_MASK_FLAGS (global constant), [4](#)

TYPE_MASK_VARIATION (global constant), [4](#)

TYPE_NULL (global constant), [4](#)

TYPE_NUMBER_FLAG_DECIMAL (global constant), [4](#)

TYPE_NUMBER_FLAG_SIGNED (global constant), [4](#)

TYPE_NUMBER_VARIATION_NORMAL (global constant), [4](#)

TYPE_NUMBER_VARIATION_PASSWORD (global constant), [4](#)

TYPE_TEXT_FLAG_AUTO_COMPLETE (global constant), [4](#)

TYPE_TEXT_FLAG_AUTO_CORRECT (global constant), [4](#)

TYPE_TEXT_FLAG_CAP_CHARACTERS (global constant), [4](#)

TYPE_TEXT_FLAG_CAP_SENTENCES (global constant), [4](#)

TYPE_TEXT_FLAG_CAP_WORDS (global constant), [4](#)

TYPE_TEXT_FLAG_IME_MULTI_LINE (global constant), [4](#)

TYPE_TEXT_FLAG_MULTI_LINE (global constant), [4](#)

TYPE_TEXT_FLAG_NO_SUGGESTIONS (global constant), [4](#)

TYPE_TEXT_VARIATION_EMAIL_ADDRESS (global constant), [4](#)

TYPE_TEXT_VARIATION_EMAIL_SUBJECT (global constant), [4](#)

TYPE_TEXT_VARIATION_FILTER (global constant), [4](#)

TYPE_TEXT_VARIATION_LONG_MESSAGE (global constant), [4](#)

TYPE_TEXT_VARIATION_NORMAL (global constant), [4](#)

TYPE_TEXT_VARIATION_PASSWORD (global constant), [4](#)

TYPE_TEXT_VARIATION_PERSON_NAME (global constant), [4](#)

TYPE_TEXT_VARIATION_PHONETIC (global constant), [4](#)

TYPE_TEXT_VARIATION_POSTAL_ADDRESS (global constant), [4](#)

TYPE_TEXT_VARIATION_SHORT_MESSAGE (global constant), [4](#)

TYPE_TEXT_VARIATION_URI (global constant), [4](#)

TYPE_TEXT_VARIATION_VISIBLE_PASSWORD (global constant), [4](#)

TYPE_TEXT_VARIATION_WEB_EDIT_TEXT (global constant), [4](#)

TYPE_TEXT_VARIATION_WEB_EMAIL_ADDRESS (global constant), [4](#)

TYPE_TEXT_VARIATION_WEB_PASSWORD (global constant), [4](#)

TYPE_USHORT_555_RGB (global constant), [119](#)

TYPE_USHORT_565_RGB (global constant), [119](#)

TYPE_USHORT_GRAY (global constant), [119](#)

U

UNICODE_CASE (global constant), [175](#)

UNIX_LINES (global constant), [175](#)

W

WARNING_MESSAGE (global constant), [134](#)

Y

YES_NO_CANCEL_OPTION (global constant), [134](#)

YES_NO_OPTION (global constant), [134](#)

YES_OPTION (global constant), [134](#)