JPHP Documentation

Выпуск 1.0 alpha1

JPHP group

03 April 2016

Оглавление

Глава 1

API (English)

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

1.1 BaseException

class `BaseException`
abstract class

extends: `Exception`

Children

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

1.2 EngineException

class `EngineException`
extends: BaseException

Methods

`EngineException::getErrorType()`

Результат `int`

1.3 ParseException

class `ParseException`
extends: BaseException

1.4 php

1.4.1 android

Android

php\android\Android
Class Android

Methods

---

static startActivity(*$class*)

Параметры

- $class – string

app

Activity

php\android\app\Activity

Methods

__construct()

setContentView(*$view*)

Параметры

- $view – php\android\view\View

getContentView()

Результат php\android\view\View

setTitle(*$title*)

Параметры

- $title – string

findViewById(*$id*)

Параметры

- $id – int

Результат php\android\view\View

isChild()

Результат bool

getParent()

Результат php\android\app\Activity

onCreate()

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*)

Параметры

- `$name` – `string`

Результат `int`

static `string`(*$name*)

Параметры

- `$name` –

Результат `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*`)`

Параметры

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

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

Параметры

- `$event` – `string`
- `$callback` – `callable`

`off(`*$event*`)`

Параметры

- `$event` – `string`

`trigger(`*$event*`)`

Параметры

- `$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*`)`

Параметры

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

Параметры

- `$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*)

Параметры

- $text – string

getText()

Результат string

setInputType(*$type*)

Параметры

- $type – int - see phpandroidtextInputType constants

getInputType()

Результат int

Toast

php\android\widget\Toast
Class Toast

Constants

constant LENGTH_SHORT

constant LENGTH_LONG

Methods

show()

cancel()

setText(*$text*)

Параметры

- $text – string

setDuration(*$duration*)

Параметры

- `$duration` – `int`

`getDuration()`

Результат `int`

static `makeText(`*$text*, *$duration = ::*`)`
Show text

Параметры

- `$text` – `string`
- `$duration` – `int`

Результат php\android\widget\Toast

1.4.2 compress

ArchiveEntry

`php\compress\ArchiveEntry`
jphp-compress-ext

Class ArchiveEntry

Methods

`getName()`

Результат `string`

`isDirectory()`

Результат `bool`

`getSize()`

Результат `int`

`setSize(`*$size*`)`

Параметры

- `$size` – `int`

`getLastModifiedDate()`

Результат `int`

ArchiveInputStream

`php\compress\ArchiveInputStream`
extends: php\io\MiscStream

jphp-compress-ext

Class ArchiveInputStream for reading archive

Methods

`__construct(`*$format*, *$source*`)`

Параметры

- `$format` – `string` - zip, tar, jar
- `$source` – php\io\File, php\io\Stream

`nextEntry()`

Результат php\compress\ArchiveEntry

ArchiveOutputStream

`php\compress\ArchiveOutputStream`
extends: php\io\MiscStream

jphp-compress-ext

Class ArchiveOutputStream for creating archives

Methods

`__construct(`*$format*, *$source*`)`

Параметры

- `$format` – `string` - zip, tar, jar, ar
- `$source` – php\io\File, php\io\Stream

`createEntry(`*$file*, *$name*`)`

Параметры

- `$file` – `string`
- `$name` – `string`

Результат php\compress\ArchiveEntry

`addFile(`*$file*, *$name*`)`

Параметры

- `$file` – `string`
- `$name` – `string`

Результат php\compress\ArchiveEntry

`addEntry(`*$entry*`)`

Параметры

- `$entry` – php\compress\ArchiveEntry

`canAddEntry(`*$entry*`)`

Параметры

- `$entry` – php\compress\ArchiveEntry

`closeEntry()`

1.4.3 concurrent

Future

php\concurrent\Future
Class Future

Methods

__construct()
private

isCancelled()

Результат bool

isDone()

Результат bool

cancel(*$mayInterruptIfRunning*)

Параметры

- $mayInterruptIfRunning – bool

Результат bool

get(*$timeout = null*)
throws php\concurrent\\Exception

Параметры

- $timeout – null, int - - in milliseconds

Результат mixed

TimeoutException

php\concurrent\TimeoutException
extends: php\lang\JavaException

Class TimeoutException

1.4.4 format

JsonProcessor

php\format\JsonProcessor
extends: php\format\Processor

Class JsonProcessor

Constants

---

constant SERIALIZE_PRETTY_PRINT

constant DESERIALIZE_AS_ARRAYS

Methods

`__construct(`*$flags = 0*`)`

Параметры

- `$flags` – `int`

`parse(`*$json*`)`
throws php\format\ProcessorException

Параметры

- `$json` – `string`, php\io\Stream

Результат `mixed`

`format(`*$value*`)`
throws php\format\ProcessorException

Параметры

- `$value` – `mixed`

Результат `string`

`formatTo(`*$value*, *$output*`)`
throws php\format\ProcessorException

Параметры

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

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

Параметры

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

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

Параметры

- `$className` – `string`
- `$handler` – `callable`

Processor

`php\format\Processor`
abstract class

Children

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

Methods

`format(`*$value*`)`
abstract

Параметры

- `$value` –

`formatTo(`*$value*, *$output*`)`
abstract

Параметры

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

`parse(`*$string*`)`
abstract

Параметры

- `$string` –

ProcessorException

`php\format\ProcessorException`
extends: `Exception`

Class ProcessorException

1.4.5 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()`

Результат php\gdx\Graphics

`getFiles()`

Результат php\gdx\Files

`getInput()`

Результат php\gdx\Input

`getAudio()`

Результат php\gdx\Audio

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

Параметры

- `$tag` – `string`
- `$message` – `string`

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

Параметры

- `$tag` – `string`
- `$message` – `string`

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

Параметры

- `$tag` – `string`
- `$message` – `string`

`setLogLevel(`*$level*`)`

Параметры

- `$level` – `int`

`getLogLevel()`

Результат `int`

`getType()`

Результат `string` - Android, Desktop, HeadlessDesktop, Applet, WebGL, iOS

`getVersion()`

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

`getJavaHeap()`

Результат `int` the Java heap memory use in bytes

`getNativeHeap()`

Результат `int` the Native heap memory use in bytes

`halt()`

`getClipboard()`

Результат php\gdx\Clipboard

ApplicationListener

php\gdx\ApplicationListener

Methods

---

create()

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

Параметры

- $width –
- $height –

render()

pause()

resume()

dispose()

assets

AssetManager

php\gdx\assets\AssetManager

Methods

__construct(*$resolver*)

Параметры

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

get(*$fileName*)

Параметры

- $fileName – string

Результат php\gdx\graphics\Texture, php\gdx\graphics\Pixmap, php\gdx\audio\Music, php\gdx\audio\Sound

loadTexture(*$fileName*)

Параметры

- $fileName – string

loadPixmap(*$fileName*)

Параметры

- $fileName – string

loadMusic(*$fileName*)

Параметры

- $fileName – string

loadSound(*$fileName*)

Параметры

- $fileName – string

unload(*$fileName*)

Параметры

- $fileName – string

isLoaded(*$fileName*)

Параметры

- $fileName – string

containsAsset(*$asset*)

Параметры

- $asset – object, mixed

getAssetFileName(*$asset*)

Параметры

- $asset – object, mixed

disposeDependencies(*$fileName*)

Параметры

- $fileName – string

update(*$millis*)

Параметры

- $millis – int - (optional)

finishLoading()

getLoadedAssets()

Результат int

getQueuedAssets()

Результат int

getProgress()

Результат float

dispose()

clear()

getReferenceCount(*$fileName*)

Параметры

- $fileName – string

Результат int

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

Параметры

- `$fileName` – `string`
- `$refCount` – `int`

`getDiagnostics()`

Результат `string`

audio

AudioDevice

`php\gdx\audio\AudioDevice`
Class AudioDevice

Methods

`__construct()`
private

`isMono()`

Результат `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.

Параметры

- `$samples` – `array`
- `$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.

Параметры

- `$samples` – `array`
- `$offset` – `int`
- `$numSamples` – `int`

`getLatency()`

Результат `int`

`dispose()`

`setVolume(`*$volume*`)`
Sets the volume in the range [0,1].

Параметры

- `$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.

Параметры

- `$samples` – `array`
- `$offset` – `int`
- `$numSamples` – `int`

`dispose()`

Music

`php\gdx\audio\Music`
Class Music

Methods

`__construct()`
private

`play()`

`pause()`

`stop()`

`isPlaying()`

Результат `bool` whether this music stream is playing

`setLooping(`*$isLooping*`)`

Параметры

- `$isLooping` – `bool`

`isLooping()`

Результат `bool`

`setVolume(`*$volume*`)`

Параметры

- `$volume` – `double`

`getVolume()`

Результат `double`

`setPan(`*$pan*, *$volume*`)`
Sets the panning and volume of this music stream.

Параметры

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

`getPosition()`
Returns the playback position in milliseconds.

Результат `double`

`dispose()`

`setOnCompletionListener(`*$listener*`)`
Register a callback to be invoked when the end of a music stream has been reached during playback.

Параметры

- `$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.

Параметры

- `$volume` – `double` - (optional)
- `$pitch` – `double` - (optional)
- `$pan` – `double` - (optional)

Результат `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.

Параметры

- `$volume` – `double` - (optional)
- `$pitch` – `double` - (optional)
- `$pan` – `double` - (optional)

Результат `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.

Параметры

- `$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.

Параметры

- `$soundId` – `int` - (optional)

`resume(`*$soundId*`)`

Параметры

- `$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

Параметры

- `$soundId` – `int`
- `$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.

Параметры

- `$soundId` – `int`
- `$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.

Параметры

- `$soundId` – `int`
- `$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.

Параметры

- `$soundId` – `int`
- `$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

Параметры

- `$samplingRate` – `int`
- `$isMono` – `boolean`

Результат 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

Параметры

- `$samplingRate` – `int`
- `$isMono` – `bool`

Результат 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

Параметры

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

Результат 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

Параметры

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

Результат 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

Результат `string` the clipboard content or null

`setContent(`*$content*`)`
Sets the content of the system clipboard.

Параметры

- `$content` – `string`

files

FileHandle

`php\gdx\files\FileHandle`
Class FileHandle

Methods

`__construct(`*$path*`)`

Параметры

- `$path` – `string`, php\io\File

`path()`

Результат `string`

`name()`

Результат `string`

`extension()`

Результат `string`

`nameWithoutExtension()`

Результат `string`

`pathWithoutExtension()`

Результат `string`

`type()`

Результат `string`

`file()`

Результат php\io\File

`read()`

Результат php\io\Stream

`readString(`*$charset*`)`

Параметры

- `$charset` – `string` - (optional)

Результат `string`

`readBytes()`

Результат `string` binary

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

Параметры

- `$append` – `bool`
- `$bufferSize` – `int` - (optional)

Результат php\io\Stream

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

Параметры

- `$string` – `string`
- `$append` – `bool`
- `$charset` – `string` - (optional)

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

Параметры

- `$binaryString` – `string`
- `$append` – `bool`

`getList(`*$suffix*`)`

Параметры

- `$suffix` – `string` - (optional)

Результат php\gdx\files\FileHandle[]

`isDirectory()`

Результат `bool`

`child(`*$name*`)`

Параметры

- `$name` – `string`

Результат php\gdx\files\FileHandle

`sibling(`*$name*`)`

Параметры

- `$name` –

Результат php\gdx\files\FileHandle

`parent()`

Результат php\gdx\files\FileHandle

`mkdirs()`
throws `php\gdx\files\\Exception`

`exists()`

Результат `bool`

`delete()`

Результат `bool`

`deleteDirectory()`

Результат `bool`

`emptyDirectory(`*$preserveTree*`)`

Параметры

- `$preserveTree` – `bool` - (optional)

`copyTo(`*$fileHandle*`)`

Параметры

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

`moveTo(`*$fileHandle*`)`

Параметры

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

`length()`

Результат `int`

`lastModified()`

Результат `int`

static `tempFile(`*$suffix*`)`

Параметры

- `$suffix` – `string`

Результат php\gdx\files\FileHandle

static `tempDirectory(`*$suffix*`)`

Параметры

- `$suffix` – `string`

Результат php\gdx\files\FileHandle

Files

php\gdx\Files
Class Files

Methods

---

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

Параметры

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

Результат php\gdx\files\FileHandle

classpath(*$path*)

Параметры

- $path – string

Результат php\gdx\files\FileHandle

internal(*$path*)

Параметры

- $path –

Результат php\gdx\files\FileHandle

external(*$path*)

Параметры

- $path –

Результат php\gdx\files\FileHandle

absolute(*$path*)

Параметры

- $path –

Результат php\gdx\files\FileHandle

local(*$path*)

Параметры

- $path –

Результат php\gdx\files\FileHandle

getExternalStoragePath()

Результат string

isExternalStorageAvailable()

Результат bool

`getLocalStoragePath()`

Результат `string`

`isLocalStorageAvailable()`

Результат `string`

Gdx

`php\gdx\Gdx`

Methods

---

static `app`

Результат php\gdx\Application

static `files`

Результат php\gdx\Files

static `graphics`

Результат php\gdx\Graphics

static `input`

Результат php\gdx\Input

static `audio`

Результат 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()`

Результат `int`

`getHeight()`

Результат `int`

`getBitsPerPixel()`

Результат `int`

`getRefreshRate()`

Результат `int`

`__toString()`

Результат `string`

Pixmap

php\gdx\graphics\Pixmap
Class Pixmap

Methods

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

Параметры

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

static `ofFile(`*$fileHandle*`)`

Параметры

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

Результат `string`

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

Параметры

- `$redOrColor` – `double`, `int`
- `$g` – `double` - (optional)

- `$b` – `double` - (optional)
- `$a` – `double` - (optional)

`fill()`

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

Параметры

- `$x` – `int`
- `$y` – `int`
- `$x2` – `int`
- `$y2` – `int`

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

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

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

Параметры

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

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

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

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

Параметры

- `$x` – `int`
- `$y` – `int`
- `$radius` – `int`

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

Параметры

- `$x` – `int`
- `$y` – `int`
- `$radius` – `int`

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

Параметры

- `$x1` – `int`
- `$y1` – `int`
- `$x2` – `int`
- `$y2` – `int`
- `$x3` – `int`
- `$y3` – `int`

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

Параметры

- `$x` – `int`
- `$y` – `int`

Результат `int` The pixel color in RGBA8888 format.

`getWidth()`

Результат `int`

`getHeight()`

Результат `int`

`dispose()`

`drawPixel(`*$x*, *$y*, *$color*`)`

Параметры

- `$x` – `int`
- `$y` – `int`
- `$color` – `int` - (optional)

`getGLFormat()`

Результат `int` one of GL_ALPHA, GL_RGB, GL_RGBA, GL_LUMINANCE, or GL_LUMINANCE_ALPHA.

`getGLInternalFormat()`

Результат `int` one of GL_ALPHA, GL_RGB, GL_RGBA, GL_LUMINANCE, or GL_LUMINANCE_ALPHA.

`getGLType()`

Результат `int` one of GL_UNSIGNED_BYTE, GL_UNSIGNED_SHORT_5_6_5, GL_UNSIGNED_SHORT_4_4_4_4

`getFormat()`

Результат `string`

`getBlending()`

Результат `string` None, SourceOver

static `setBlending(`*$blending*`)`
Sets the type of Blending to be used for all operations. Default is 'SourceOver'

Параметры

- `$blending` – `string`

static `setFilter(`*$filter*`)`
Filters to be used with Pixmap.drawPixmap(Pixmap, int, int, int, int, int, int, int, int).

Параметры

- `$filter` – `string` - NearestNeighbour, BiLinear

Sprite

`php\gdx\graphics\Sprite`
Class Sprite

Methods

`__construct(`*$texture*, *$width*, *$height*, *$x*, *$y*`)`

Параметры

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

`setBounds(`*$x*, *$y*, *$width*, *$height*`)`

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

`setSize(`*$width*, *$height*`)`

Параметры

- `$width` – `int`
- `$height` – `int`

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

Параметры

- `$x` – `int`

- $y – int

setX(*$x*)

Параметры

- $x – int

getX()

Результат int

setY(*$y*)

Параметры

- $y – int

getY()

Результат int

translateX(*$xAmount*)

Параметры

- $xAmount – double

translateY(*$yAmount*)

Параметры

- $yAmount – double

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

Параметры

- $xAmount – double
- $yAmount – double

setAlpha(*$alpha*)

Параметры

- $alpha – double

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

Параметры

- $originX – int
- $originY – int

setOriginCenter()

setRotation(*$degrees*)

Параметры

- $degrees – double

getRotation()

Результат double

rotate(*$degrees*)

Параметры

- `$degrees` – `double`

`rotate90(`*$clockwise*`)`

Параметры

- `$clockwise` – `bool`

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

Параметры

- `$scaleXY` – `double`
- `$scaleY` – `double` - (optional)

`scale(`*$amount*`)`

Параметры

- `$amount` – `double`

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

Параметры

- `$x` – `int`
- `$y` – `int`

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

Параметры

- `$xAmount` – `double`
- `$yAmount` – `double`

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

Параметры

- `$batch` – php\gdx\graphics\SpriteBatch
- `$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*`)`

Параметры

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

- `$useMipMaps` – `bool` - (optional)
- `$format` – `string` - (optional)

Результат php\gdx\graphics\Texture

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

Параметры

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

`getWidth`()

Результат `int`

`getHeight`()

Результат `int`

`getDepth`()

Результат `int`

`isManaged`()

Результат `bool`

Graphics

`php\gdx\Graphics`
Class Graphics

Methods

`isGL30Available`()

Результат `bool`

`getWidth`()

Результат `int`

`getHeight`()

Результат `int`

`getDeltaTime`()

Результат `double`

`getRawDeltaTime`()

Результат `double`

`getFramesPerSecond`()

Результат `int`

`getDensity`()

Результат `float`

`supportsDisplayModeChange()`

Результат `bool`

`getDisplayModes()`

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

`getDesktopDisplayMode()`

Результат php\gdx\graphics\DisplayMode

`setDisplayMode(`*$widthOrDisplayMode*, *$height*, *$fullscreen*`)`

Параметры

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

`setTitle(`*$title*`)`

Параметры

- `$title` – `string`

`setVSync(`*$vsync*`)`

Параметры

- `$vsync` – `bool`

`supportsExtension(`*$extension*`)`

Параметры

- `$extension` – `string`

Результат `bool`

`setContinuousRendering(`*$value*`)`

Параметры

- `$value` – `bool`

`isContinuousRendering()`

Результат `bool`

`requestRendering()`

`isFullscreen()`

Результат `bool`

Input

php\gdx\Input
final class

Methods

`__construct()`
private

`getAccelerometerX()`

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

`getAccelerometerY()`

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

`getAccelerometerZ()`

Результат `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.

Параметры

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

Результат `int`

`getDeltaX(`*$pointer*`)`

Параметры

- `$pointer` – `int` - (optional) the pointer id.

Результат `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.

Параметры

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

Результат `int`

`getDeltaY(`*$pointer*`)`

Параметры

- `$pointer` – `int` - (optional) the pointer id.

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

`isTouched(`*$pointer*`)`

Параметры

- `$pointer` – `int` - (optional)

Результат `bool` whether the screen is currently touched.

`justTouched()`

Результат `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.

Параметры

- `$button` – `int`

Результат `bool`

`isKeyPressed(`*$key*`)`
Returns whether the key is pressed.

Параметры

- `$key` – `int`

`setOnscreenKeyboardVisible(`*$visible*`)`
Sets the on-screen keyboard visible if available.

Параметры

- `$visible` – `bool`

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

Параметры

- `$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 earths center.

Результат `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.

Результат `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.

Результат `double`

`getCurrentEventTime()`

Результат `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.

Параметры

- `$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.

Параметры

- `$catchMenu` – `bool`

`getRotation()`

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

`getNativeOrientation()`

Результат `string` the native orientation of the device.

`setCursorCatched(`*$catched*`)`
Only viable on the desktop. Will confine the mouse cursor location to the window and hide the mouse cursor.

Параметры

- `$catched` – `bool` - whether to catch or not to catch the mouse cursor

`isCursorCatched()`

Результат `bool` whether the mouse cursor is catched.

`setCursorPosition(`*$x*, *$y*`)`
Only viable on the desktop. Will set the mouse cursor location to the given window coordinates (origin top-left corner).

Параметры

- `$x` – `int`
- `$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.

Параметры

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

LwjglApplication

`php\gdx\LwjglApplication`
extends: php\gdx\Application

Methods

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

Параметры

- `$listener` – php\gdx\ApplicationListener
- `$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*`)`

Параметры

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

static `getDesktopDisplayMode`

Результат php\gdx\graphics\DisplayMode

static `getDisplayModes`

Результат `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).

Параметры

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

math

Polygon

`php\gdx\math\Polygon`
Class Polygon

Methods

`__construct(`*$vertices*`)`

Параметры

- `$vertices` – `array` - (optional)

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

Результат `float[]`

`getTransformedVertices()`

Результат `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.

Параметры

- `$originX` – `float`
- `$originY` – `float`

`setPosition(`*$x*, *$y*`)`
Sets the polygon's position within the world.

Параметры

- `$x` – `float`
- `$y` – `float`

`setVertices(`*$vertices*`)`

Параметры

- `$vertices` – `array`

`translate(`*$x*, *$y*`)`
Translates the polygon's position by the specified horizontal and vertical amounts.

Параметры

- `$x` – `float`
- `$y` – `float`

`setRotation(`*$degrees*`)`
Sets the polygon to be rotated by the supplied degrees.

Параметры

- `$degrees` – `float`

`rotate(`*$degrees*`)`
Applies additional rotation to the polygon by the supplied degrees.

Параметры

- `$degrees` – `float`

`setScale(`*$scaleX*, *$scaleY*`)`
Sets the amount of scaling to be applied to the polygon.

Параметры

- `$scaleX` – `float`
- `$scaleY` – `float`

`scale(`*$amount*`)`
Applies additional scaling to the polygon by the supplied amount.

Параметры

- `$amount` – `float`

`dirty()`

`area()`
Returns the area contained within the polygon.

Результат `float`

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

Параметры

- `$x` – `float`
- `$y` – `float`

Результат `bool`

`getX()`

Результат `float`

`getY()`

Результат `float`

`getOriginX()`

Результат `float`

`getOriginY()`

Результат `float`

`getRotation()`

Результат `float`

`getScaleX()`

Результат `float`

`getScaleY()`

Результат `float`

Polyline

`php\gdx\math\Polyline`
Class Polyline

Methods

`__construct(`*$vertices*`)`

Параметры

- `$vertices` – `array` - (optional)

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

Результат `float[]`

`getTransformedVertices()`

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

`getLength()`

Результат `float` Returns the euclidean length of the polyline without scaling

`getScaledLength()`

Результат `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.

Параметры

- `$originX` – `float`
- `$originY` – `float`

`setPosition(`*$x*, *$y*`)`
Sets the polyline's position within the world.

Параметры

- `$x` – `float`
- `$y` – `float`

`setVertices(`*$vertices*`)`

Параметры

- `$vertices` – `array`

`translate(`*$x*, *$y*`)`
Translates the polyline's position by the specified horizontal and vertical amounts.

Параметры

- `$x` – `float`
- `$y` – `float`

`setRotation(`*$degrees*`)`
Sets the polyline to be rotated by the supplied degrees.

Параметры

- `$degrees` – `float`

`rotate(`*$degrees*`)`
Applies additional rotation to the polyline by the supplied degrees.

Параметры

- `$degrees` – `float`

`setScale(`*$scaleX*, *$scaleY*`)`
Sets the amount of scaling to be applied to the polyline.

Параметры

- `$scaleX` – `float`
- `$scaleY` – `float`

`scale(`*$amount*`)`
Applies additional scaling to the polyline by the supplied amount.

Параметры

- `$amount` – `float`

`getX()`

Результат `float`

`getY()`

Результат `float`

`getOriginX()`

Результат `float`

`getOriginY()`

Результат `float`

`getRotation()`

Результат `float`

`getScaleX()`

Результат `float`

`getScaleY()`

Результат `float`

Vector2

`php\gdx\math\Vector2`
Class Vector2

Methods

`__construct(`*$x*, *$y*`)`

Параметры

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

`x()`

Результат `float`

`y()`

Результат `float`

`len()`

Результат `float` The euclidean length

`len2()`

Результат `float` The squared euclidean length

`set(`*$x*, *$y*`)`
Sets this vector from the given vector or x, y

Параметры

- `$x` – `float`, php\gdx\math\Vector2

- `$y` – `float` - (optional)

Результат php\gdx\math\Vector2

`sub(`*$x*, *$y*`)`
Subtracts the given vector from this vector.

Параметры

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

Результат php\gdx\math\Vector2

`nor()`
Normalizes this vector. Does nothing if it is zero.

Результат php\gdx\math\Vector2

`add(`*$x*, *$y*`)`
Adds the given vector to this vector

Параметры

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

`dot(`*$ox*, *$oy*`)`

Параметры

- `$ox` – `float`, php\gdx\math\Vector2
- `$oy` – `float` - (optional)

Результат `float` The dot product between this and the other vector

`scl(`*$x*, *$y*`)`
Scales this vector by a scalar

Параметры

- `$x` – `float`
- `$y` – `float` - (optional)

Результат php\gdx\math\Vector2

`mulAdd(`*$vec*, *$scalar*`)`
First scale a supplied vector, then add it to this vector.

Параметры

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

Результат php\gdx\math\Vector2

`dst(`*$x*, *$y*`)`

Параметры

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

Результат `float` the distance between this and the other vector

`dst2(`*$x*, *$y*`)`

Параметры

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

Результат `float` the squared distance between this and the other vector

`limit(`*$limit*`)`

Параметры

- `$limit` – `float`

Результат php\gdx\math\Vector2

`clamp(`*$min*, *$max*`)`

Параметры

- `$min` – `float`
- `$max` – `float`

Результат php\gdx\math\Vector2

`crs(`*$x*, *$y*`)`

Параметры

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

Результат `float`

`angle()`

Результат `float`

`getAngleRad()`

Результат `float`

`setAngle(`*$degrees*`)`

Параметры

- `$degrees` – `float`

Результат php\gdx\math\Vector2

`setAngleRad(`*$radians*`)`

Параметры

- `$radians` – `float`

Результат php\gdx\math\Vector2

`rotate(`*$degrees*`)`

Параметры

- `$degrees` – `float`

Результат php\gdx\math\Vector2

`rotateRad(`*$radians*`)`

Параметры

- `$radians` – `float`

Результат php\gdx\math\Vector2

`rotate90(`*$dir*`)`

Параметры

- `$dir` – `int`

Результат 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.

Параметры

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

Результат php\gdx\math\Vector2

`epsilonEquals(`*$other*, *$epsilon*`)`

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

Параметры

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

Результат `bool`

`isUnit(`*$margin*`)`

Whether this vector is a unit length vector

Параметры

- `$margin` – `float` - (optional)

Результат `bool`

`isZero(`*$margin*`)`

Параметры

- `$margin` – `float` - (optional)

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

Параметры

- `$other` – php\gdx\math\Vector2
- `$epsilon` – `float` - (optional)

Результат `bool`

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

Параметры

- `$other` – php\gdx\math\Vector2
- `$epsilon` – `float` - (optional)

Результат `bool`

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

Параметры

- `$other` – php\gdx\math\Vector2
- `$epsilon` – `float` - (optional)

Результат `bool`

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

Параметры

- `$vector` – php\gdx\math\Vector2
- `$epsilon` – `float` - (optional)

`hasSameDirection(`*$vector*`)`

Параметры

- `$vector` – php\gdx\math\Vector2

Результат `bool`

`hasOppositeDirection(`*$vector*`)`

Параметры

- `$vector` – php\gdx\math\Vector2

Результат `bool`

1.4.6 io

File

`php\io\File`
Class File

Constants

---

constant `PATH_SEPARATOR`

constant `DIRECTORY_SEPARATOR`

constant `PATH_NAME_CASE_INSENSITIVE`

Methods

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

Параметры

- `$path` – `string`
- `$child` – `null`, `string`

`exists()`

Результат `bool`

`canExecute()`

Результат `bool`

`canWrite()`

Результат `bool`

`canRead()`

Результат `bool`

`getName()`

Результат `string`

`getAbsolutePath()`

Результат `string`

`getCanonicalPath()`
throws php\io\IOException

Результат `string`

`getParent()`

Результат `string`

`getPath()`

Результат `string`

`getAbsoluteFile()`

Результат php\io\File

`getCanonicalFile()`
throws php\io\IOException

Результат php\io\File

`getParentFile()`

Результат php\io\File

`mkdir()`

Результат `bool`

`mkdirs()`

Результат `bool`

`isFile()`

Результат `bool`

`isDirectory()`

Результат `bool`

`isAbsolute()`

Результат `bool`

`isHidden()`

Результат `bool`

`matches(`*$pattern*`)`

Параметры

- `$pattern` – `string` - the "glob" and "regex" syntaxes, and may support others.

Результат `bool`

`delete()`

Результат `bool`

`deleteOnExit()`

Результат `void`

`createNewFile(`*$withDirs = false*`)`

Параметры

- `$withDirs` – `bool`

Результат `bool`

`lastModified()`

Результат `int`

`length()`

Результат `int`

`crc32()`

Результат `int`, `null` null if not exists or io exception

`hash(`*$algorithm = 'MD5'*`)`

Параметры

- `$algorithm` – `string`

Результат `string`, `null` if not exists or io exception

`renameTo(`*$newName*`)`

Параметры

- `$newName` – `string`

Результат `bool`

`setExecutable(`*$value, $ownerOnly = true*`)`

Параметры

- `$value` – `bool`
- `$ownerOnly` – `bool`

Результат `bool`

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

Параметры

- `$value` – `bool`
- `$ownerOnly` – `bool`

Результат `bool`

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

Параметры

- `$value` – `bool`
- `$ownerOnly` – `bool`

Результат `bool`

`setReadOnly()`

Результат `bool`

`setLastModified(`*$time*`)`

Параметры

- `$time` – `int`

Результат `bool`

`compareTo(`*$file*`)`

Параметры

- `$file` – `string`, php\io\File

Результат `int`

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

Параметры

- `$filter` – `callable`

Результат `string[]`

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

Параметры

- `$filter` – `callable` - (File $directory, $name)

Результат php\io\File[]

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

Параметры

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

Результат 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.

Результат php\io\File[]

static `of(`*$path*`)`

Параметры

- `$path` – `string`

Результат php\io\File

FileStream

php\io\FileStream
extends: php\io\Stream

Methods

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

Параметры

- `$length` – `int` - - count of bytes

Результат `mixed`

`readFully()`
throws php\io\IOException

Результат `mixed`

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

Параметры

- `$value` – `string`
- `$length` – `null`, `int`

Результат `int`

`eof()`

Результат `bool`

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

Параметры

- `$position` – `int`

Результат `mixed`

`getPosition()`
throws php\io\IOException

Результат `int`

`close()`

Результат `mixed`

`length()`

Результат `int`

`getFilePointer()`
throws php\io\IOException

Результат `int`

`truncate(`*$size*`)`
throws php\io\IOException

Параметры

- `$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\compress\ArchiveInputStream
- **class** php\compress\ArchiveOutputStream
- **class** php\io\MemoryStream

Class MiscStream

Methods

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

Параметры

- `$length` – `int` - - count of bytes

Результат `mixed`

`readFully()`
throws php\io\IOException

Результат `mixed`

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

Параметры

- `$value` – `string`
- `$length` – `null`, `int`

Результат `int`

`eof()`

Результат `bool`

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

Параметры

- `$position` – `int`

Результат `mixed`

`getPosition()`
throws php\io\IOException

Результат `int`

`close()`

Результат `mixed`

`length()`

Результат `int`

`flush()`
throws php\io\IOException

ResourceStream

`php\io\ResourceStream`
extends: php\io\Stream

Class ResourceStream

Methods

`__construct(`*$path*`)`

Параметры

- `$path` – `string`

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

Параметры

- `$length` – `int` - - count of bytes

Результат `mixed`

`readFully()`
throws php\io\IOException

Результат `mixed`

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

Параметры

- `$value` – `string`
- `$length` – `null`, `int`

Результат `int`

`eof()`

Результат `bool`

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

Параметры

- `$position` – `int`

Результат `mixed`

`getPosition()`
throws php\io\IOException

Результат `int`

`close()`

Результат `mixed`

`toExternalForm()`

Результат `string`

static `getResources(`*$name*`)`

Параметры

- `$name` – `string`

Результат 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()`

Результат `string`

`getMode()`

Результат `string`

`read(`*$length*`)`
abstract

throws php\io\IOException

Параметры

- `$length` – `int` - - count of bytes

Результат `mixed`

`readFully()`
abstract

throws php\io\IOException

Результат `mixed`

`write(`*$value*, *$length = null*`)`
abstract

throws php\io\IOException

Параметры

- `$value` – `string`
- `$length` – `null`, `int`

Результат `int`

`eof()`
abstract

Результат `bool`

`seek(`*$position*`)`
abstract

throws php\io\IOException

Параметры

- `$position` – `int`

Результат `mixed`

`getPosition()`
abstract

throws php\io\IOException

Результат `int`

`close()`
abstract

Результат `mixed`

`__construct(`*$path*, *$mode = null*`)`

Параметры

- `$path` – `string`
- `$mode` – `null`, `string`

Результат php\io\Stream

`setContext(`*$context*`)`

Параметры

- `$context` –

Результат `void`

`getContext()`

Результат `mixed`

static `of(`*$path*, *$mode = 'r'*`)`
throws php\io\IOException

Параметры

- `$path` – `string`
- `$mode` – `string`

Результат 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

Параметры

- `$path` – `string`
- `$mode` – `string`

Результат `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

Параметры

- `$path` – `string`
- `$data` – `string`
- `$mode` – `string`

static `tryAccess`(*$path*, *$onAccess*, *$mode = 'r'*)

Open a stream and close it after calling $onAccess automatically.

throws php\io\IOException

Параметры

- `$path` – `string`
- `$onAccess` – `callable` - (Stream $stream)
- `$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

Параметры

- `$path` – `string`

Результат `bool`

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

Параметры

- `$protocol` – `string`
- `$className` – `string`

static `unregister`(*$protocol*)

Параметры

- `$protocol` –

1.4.7 jsoup

Connection

`php\jsoup\Connection`

abstract class

Constants

---

constant `METHOD_POST`

constant `METHOD_GET`

Methods

`data`(*$data*)

Параметры

- `$data` – `array`

Результат php\jsoup\Connection

`cookies(`*$data*`)`

Параметры

- `$data` – `array`

Результат php\jsoup\Connection

`headers(`*$data*`)`

Параметры

- `$data` – `array`

Результат php\jsoup\Connection

`header(`*$name*, *$value*`)`

Параметры

- `$name` – `string`
- `$value` – `string`

Результат php\jsoup\Connection

`url(`*$url*`)`

Параметры

- `$url` – `string`

Результат php\jsoup\Connection

`method(`*$method*`)`

Параметры

- `$method` – `string` - POST or GET

Результат php\jsoup\Connection

`userAgent(`*$userAgent*`)`

Параметры

- `$userAgent` – `string`

Результат php\jsoup\Connection

`maxBodySize(`*$bytes*`)`

Параметры

- `$bytes` – `int`

Результат php\jsoup\Connection

`timeout(`*$millis*`)`

Параметры

- `$millis` – `int`

Результат php\jsoup\Connection

`referrer(`*$referrer*`)`

Параметры

- `$referrer` – `string`

Результат php\jsoup\Connection

`followRedirects(`*$enable*`)`

Параметры

- `$enable` – `bool`

Результат php\jsoup\Connection

`ignoreHttpErrors(`*$enable*`)`

Параметры

- `$enable` – `bool`

Результат php\jsoup\Connection

`ignoreContentType(`*$enable*`)`

Параметры

- `$enable` – `bool`

Результат php\jsoup\Connection

`execute()`

Результат php\jsoup\ConnectionResponse

`get()`

Результат php\jsoup\Document

`post()`

Результат php\jsoup\Document

`request()`

Результат php\jsoup\ConnectionRequest

`response()`

Результат php\jsoup\ConnectionResponse

ConnectionRequest

`php\jsoup\ConnectionRequest`
abstract class

Methods

`timeout(`*$millis*`)`
Setter and getter for timeout.

Параметры

- `$millis` – `int` - (optional)

Результат `int`, php\jsoup\ConnectionRequest

`maxBodySize(`*$bytes*`)`
Setter and getter for max of body size.

Параметры

- `$bytes` – `int` - (optional)

Результат `int`, php\jsoup\ConnectionRequest

`followRedirects(`*$enable*`)`
Setter and getter.

Параметры

- `$enable` – `bool` - (optional)

Результат `bool`, php\jsoup\ConnectionRequest

`ignoreHttpErrors(`*$enable*`)`

Параметры

- `$enable` – `bool` - (optional)

Результат `bool`, php\jsoup\ConnectionRequest

`ignoreContentType(`*$enable*`)`

Параметры

- `$enable` – `bool` - (optional)

Результат `bool`, php\jsoup\ConnectionRequest

ConnectionResponse

`php\jsoup\ConnectionResponse`
abstract class

Methods

`headers()`

Результат `array`

`cookies()`

Результат `array`

`statusCode()`
Get the status code of the response.

Результат `int`

`statusMessage()`
Get the status message of the response.

Результат `string`

`charset()`

Результат `string`

`body()`

Результат `string`

`bodyAsBytes()`

Результат `string` binary string

`contentType()`
Get the response content type (e.g. "text/html");

Результат `string`

`parse()`

Результат php\jsoup\Document

Document

`php\jsoup\Document`
abstract class

Methods

`location()`

Результат `string`

`title(`*$value*`)`

Параметры

- `$value` – `string` - (optional)

Результат `string`

`head()`

Результат php\jsoup\Element

`body()`

Результат php\jsoup\Element

`select(`*$cssQuery*`)`

Параметры

- `$cssQuery` – `string`

Результат php\jsoup\Elements

Element

`php\jsoup\Element`
abstract class

Methods

`html(`*$html*`)`

Параметры

- `$html` – `string` - (optional)

Результат `string`

`text()`

Результат `string`

`nodeName()`

Результат `string`

`tagName(`*$tagName*`)`

Параметры

- `$tagName` – `string` - (optional)

Результат `string`

`isBlock()`

Результат `bool`

`id()`

Результат `string`

`attr(`*$attributeKey*, *$attributeValue*`)`

Параметры

- `$attributeKey` –
- `$attributeValue` –

  – (optional)

Результат `php\jsoup\$this`

`dataset()`

Результат `array`

`parent()`

Результат php\jsoup\Element

`parents()`

Результат php\jsoup\Elements

`child(`*$index*`)`

Параметры

- `$index` – `int`

Результат php\jsoup\Element

`children()`

Результат php\jsoup\Elements

`select(`*$cssQuery*`)`

Параметры

- `$cssQuery` – `string`

Результат php\jsoup\Elements

Elements

php\jsoup\Elements
abstract class

implements: Iterator

Methods

`text()`

Результат string

`hasText()`

Результат bool

`html(`*$html*`)`

Параметры

- $html – string - (optional)

Результат string

`outerHtml()`

Результат string

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

Параметры

- $attributeKey – string
- $value –

  – (optional)

Результат string, php\jsoup\$this

`hasAttr(`*$attributeKey*`)`

Параметры

- $attributeKey – string

Результат bool

`removeAttr(`*$attributeKey*`)`

Параметры

- $attributeKey – string

Результат php\jsoup\Elements

`addClass(`*$class*`)`

Параметры

- $class – string

Результат php\jsoup\Elements

`removeClass(`*$class*`)`

Параметры

- `$class` – `string`

Результат php\jsoup\Elements

`hasClass(`*$class*`)`

Параметры

- `$class` – `string`

Результат `bool`

`toggleClass(`*$class*`)`

Параметры

- `$class` – `string`

Результат `bool`

`val(`*$value*`)`

Параметры

- `$value` – `string` - (optional)

Результат `php\jsoup\$this`

`prepend(`*$html*`)`

Параметры

- `$html` – `string`

Результат php\jsoup\Elements

`append(`*$html*`)`

Параметры

- `$html` – `string`

Результат php\jsoup\Elements

`before(`*$html*`)`

Параметры

- `$html` – `string`

Результат php\jsoup\Elements

`after(`*$html*`)`

Параметры

- `$html` – `string`

Результат php\jsoup\Elements

`select(`*$query*`)`

Параметры

- `$query` – `string`

Результат php\jsoup\Elements

`first()`

Результат php\jsoup\Element

`last()`

Результат php\jsoup\Element

`not(`*$query*`)`

Параметры

- `$query` – `string`

Результат php\jsoup\Elements

`is(`*$query*`)`

Параметры

- `$query` – `string`

Результат `bool`

`parents()`

Результат php\jsoup\Elements

Jsoup

`php\jsoup\Jsoup`
final class

Methods

`__construct()`
private

static `connect(`*$url*`)`

Параметры

- `$url` – `string`

Результат php\jsoup\Connection

static `parse(`*$source*, *$encoding*, *$baseUri*`)`

Параметры

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

Результат php\jsoup\Document

static `parseText(`*$text*, *$baseUri*`)`

Параметры

- `$text` – `string`
- `$baseUri` – `string` - (optional)

Результат php\jsoup\Document

1.4.8 lang

ClassLoader

php\lang\ClassLoader
    abstract class

Children

- **class** php\util\LauncherClassLoader

Methods

`loadClass(`*$name*`)`
    abstract

Параметры

- `$name` – `string`

`register()`

`unregister()`

Environment

php\lang\Environment
    Class Environment

Constants

---

constant `CONCURRENT`

constant `HOT_RELOAD`

Methods

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

Параметры

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

`registerSourceMap(`*$sourceMap*`)`

Параметры

- `$sourceMap` – php\lang\SourceMap

`unregisterSourceMap(`*$sourceMap*`)`

Параметры

- `$sourceMap` – php\lang\SourceMap

`execute(`*$runnable*`)`
Executes $runnable in the environment

Параметры

- `$runnable` – `callable` - - in new environment

Результат `mixed`

`importClass(`*$className*`)`
Imports the $className to the environment

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

Параметры

- `$className` – `string`

`exportClass(`*$className*`)`
Exports the $className from th environment

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

Параметры

- `$className` – `string`

`importFunction(`*$functionName*`)`
Imports the $functionName to the environment

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

Параметры

- `$functionName` – `string`

`exportFunction(`*$functionName*`)`
Exports the $functionName from the environment

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

Параметры

- `$functionName` – `string`

`importAutoLoaders()`

`defineConstant(`*$name*, *$value*, *$caseSensitive = true*`)`
throws `php\lang\\Exception` - if constant already registered or value is not scalar type

Параметры

- `$name` – `string`
- `$value` – `mixed` - - scalar value
- `$caseSensitive` – `bool`

`onMessage(`*$callback*`)`
Handles messages that sent to the environment

Параметры

- `$callback` – `callable`

`onOutput(`*$callback*`)`

Параметры

- `$callback` – `callable`

`sendMessage(`*$message*`)`
Send message to the environment

Параметры

- `$message` – `mixed`

Результат `mixed`

`findModule(`*$path*`)`

Параметры

- `$path` – `string`

Результат php\lang\Module, `null`

static `current`
Get environment of current execution

Результат 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*`)`

Параметры

- `$callback` – `callable`

`callArray(`*$args*`)`
Call with array arguments

Параметры

- `$args` – `array`

Результат `mixed`

`call()`
Call the current callback

Результат `int`, `mixed`

`__invoke()`

`canAccess()`
Check access to invoke the method at a moment

Результат `bool`

`getDescription()`
Returns description of the method - name + argument info

Результат `string`

`getArgumentCount()`
Returns argument count of the method

Результат `int`

`isClosure()`
Checks it is a closure

Результат `bool`

`isNamedFunction()`
Checks it is a named function

Результат `bool`

`isStaticCall()`
Checks it is a static call

Результат `bool`

`isDynamicCall()`
Checks it is a dynamic call

Результат `bool`

static `of(`*$callback*`)`

Параметры

- `$callback` – `mixed`, `callable`

Результат 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

Параметры

- `$className` – `string` - - full name of java class

`isStatic()`

Результат `bool`

`isFinal()`

Результат `bool`

`isAbstract()`

Результат `bool`

`isInterface()`

Результат `bool`

`isEnum()`

Результат `bool`

`isAnnotation()`

Результат `bool`

`isArray()`

Результат `bool`

`isPrimitive()`

Результат `bool`

`isAnonymousClass()`

Результат `bool`

`isMemberClass()`

Результат `bool`

`getName()`

Результат `string`

`getSimpleName()`

Результат `string`

`getCanonicalName()`

Результат `string`

`getSuperClass()`

Результат php\lang\JavaClass, `null`

`getModifiers()`

Результат `int`

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

Параметры

- `$annotationClassName` – `string`

Результат `bool`

`getInterfaces()`

Результат php\lang\JavaClass[]

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

Параметры

- `$name` – `string`
- `$types` – `array`

Результат php\lang\JavaMethod

`getDeclaredMethods()`

Результат php\lang\JavaMethod[]

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

Параметры

- `$name` –

Результат php\lang\JavaField

`getDeclaredFields()`

Результат php\lang\JavaField[]

`newInstance()`
throws php\lang\JavaException

Результат php\lang\JavaObject

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

Параметры

- `$types` – `array`
- `$arguments` – `array`

Результат php\lang\JavaObject

`isAssignableFrom(`*$class*`)`

Параметры

- `$class` – php\lang\JavaClass

Результат `bool`

`isSubClass(`*$className*`)`
throws php\lang\JavaException

Параметры

- `$className` – `string`

Результат `bool`

`getEnumConstants()`

Результат php\lang\JavaObject[]

`getResource(`*$name*`)`

Параметры

- `$name` – `string`

Результат `string`, `null` - filename

static `primitive(`*$name*`)`

Параметры

- `$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

Результат `bool`

`isNullPointerException()`
Check exception instance of java.lang.NullPointerException

Результат `bool`

`isIllegalArgumentException()`
Check exception instance of java.lang.IllegalArgumentException

Результат bool

isNumberFormatException()
Check exception instance of java.lang.NumberFormatException

Результат bool

getExceptionClass()

Результат php\lang\JavaClass

getJavaException()

Результат php\lang\JavaObject

printJVMStackTrace()

JavaField

php\lang\JavaField
final class

extends: php\lang\JavaReflection

Methods

---

get(*$object = null*)
throws php\lang\JavaException

Параметры

- $object – php\lang\JavaObject

Результат mixed

set(*$object = null*, *$value*)
throws php\lang\JavaException

Параметры

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

isStatic()

Результат bool

isFinal()

Результат bool

isPublic()

Результат bool

isProtected()

Результат bool

isPrivate()

Результат bool

isTransient()

Результат `bool`

`isVolatile()`

Результат `bool`

`getModifiers()`

Результат `int`

`getName()`

Результат `string`

`getDeclaringClass()`

Результат php\lang\JavaClass

JavaMethod

`php\lang\JavaMethod`

final class

extends: php\lang\JavaReflection

Methods

`invoke(`*$object = null*`)`

Invoke method

Параметры

- `$object` – php\lang\JavaObject

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

Параметры

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

`getName()`

Результат `string`

`isStatic()`

Результат `bool`

`isFinal()`

Результат `bool`

`isAbstract()`

Результат `bool`

`isPublic()`

Результат `bool`

`isProtected()`

Результат `bool`

`isPrivate()`

Результат bool

isNative()

Результат bool

isSynchronized()

Результат bool

isVarArgs()

Результат bool

getDeclaringClass()

Результат php\lang\JavaClass

getReturnedType()

Результат php\lang\JavaClass

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

Параметры

- $annotationClassName – string

Результат bool

getParameterTypes()

Результат php\lang\JavaClass[]

getParameterCount()

Результат int

JavaObject

php\lang\JavaObject
final class

Methods

---

getClass()
Get class of object

Результат php\lang\JavaClass

getClassName()
Get name of class of object

Результат 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

Параметры

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

getName()

Результат string

call(*$variables = null*)
Include module and return result

Параметры

- $variables – array

Результат mixed

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

Параметры

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

NotImplementedException

php\lang\NotImplementedException
extends: Exception

Class NotImplementedException

NumberFormatException

php\lang\NumberFormatException
extends: php\lang\JavaException

Class NumberFormatException

Process

php\lang\Process
Class Process

Methods

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

Параметры

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

start()
throws php\lang\IllegalStateException

Результат 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

Результат php\lang\Process

getExitValue()
Returns the exit value for the subprocess.

throws php\lang\IllegalStateException

Результат 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

Результат 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

Результат 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

Результат php\io\Stream

SourceMap

`php\lang\SourceMap`
Class SourceMap

Methods

`__construct(`*$moduleName*`)`

Параметры

- `$moduleName` – `string`

`getModuleName()`

Результат `string`

`getSourceLine(`*$compiledLine*`)`

Параметры

- `$compiledLine` – `int`

Результат `int` -1 if not found

`getCompiledLine(`*$sourceLine*`)`

Параметры

- `$sourceLine` – `int`

Результат `int` -1 if not found

`insertLines(`*$inserts*, *$allCountLines*`)`

Параметры

- `$inserts` – `array` - int[][2] [[line, lineCount], [line, lineCount], ...]
- `$allCountLines` – `int` - original source line count.

`addLine(`*$sourceLine*, *$compiledLine*`)`

Параметры

- `$sourceLine` – `int`
- `$compiledLine` – `int`

`clear()`

`toArray()`

Результат `array`

System

php\lang\System
final class

Methods

__construct()
private

static halt(*$status*)
Exit from program with status globally

Параметры

- $status – int

static gc

static getEnv

Результат string[]

static getProperty(*$name*, *$def = ''*)
Gets a system property by name

Параметры

- $name –
- $def – string

Результат string

Thread

php\lang\Thread
Class Thread

Constants

---

constant MAX_PRIORITY

constant MIN_PRIORITY

constant NORM_PRIORITY

Methods

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

Параметры

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

getId()

Результат `int`

`getName()`

Результат `string`

`setName(`*$value*`)`

Параметры

- `$value` – `string`

`getGroup()`

Результат php\lang\ThreadGroup

`isDaemon()`

Результат `bool`

`setDaemon(`*$value*`)`

Параметры

- `$value` – `bool`

`isInterrupted()`

Результат `bool`

`isAlive()`

Результат `bool`

`start()`

`run()`

`interrupt()`

`join(`*$millis = 0*, *$nanos = 0*`)`
Waits at most $millis milliseconds plus $nanos nanoseconds for this thread to die.

Параметры

- `$millis` – `int`
- `$nanos` – `int`

static `doYield`

static `sleep(`*$millis*, *$nanos = 0*`)`
Causes the currently executing thread to sleep (temporarily cease execution)

Параметры

- `$millis` – `int`
- `$nanos` – `int`

static `getActiveCount`

Результат `int`

static `current`
Get current thread

Результат php\lang\Thread

ThreadGroup

php\lang\ThreadGroup
Class ThreadGroup

Methods

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

Параметры

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

getName()

Результат string

getParent()

Результат php\lang\ThreadGroup, null

getActiveCount()

Результат int

getActiveGroupCount()

Результат int

isDaemon()

Результат bool

setDaemon(*$value*)

Параметры

- $value – bool

isDestroyed()

Результат bool

getMaxPriority()

Результат int

setMaxPriority(*$value*)

Параметры

- $value – int

destroy()

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

throws php\lang\JavaException

interrupt()

ThreadPool

`php\lang\ThreadPool`
Class ThreadPool

Methods

`__construct()`
private

`isScheduled()`
Is Scheduled ?

Результат `bool`

`isShutdown()`
Is Shutdown?

Результат `bool`

`isTerminated()`

Результат `bool`

`execute(`*$runnable*, *$env = null*`)`
Execute some $runnable via the Executor Service

Параметры

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

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

Параметры

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

Результат php\concurrent\Future

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

Параметры

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

Результат 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\lang\\Exception`

Параметры

- `$timeout` – `int` - - in milliseconds

Результат `bool`

static `create`(*$coreSize*, *$maxSize*, *$keepAliveTime = 0*)

Параметры

- `$coreSize` – `int` - the number of threads to keep in the pool, even if they are idle
- `$maxSize` – `int` - the maximum number of threads to allow in the pool
- `$keepAliveTime` – `int` - in millis

Результат php\lang\ThreadPool

static `createFixed`(*$max*)

Параметры

- `$max` – `int`

Результат php\lang\ThreadPool

static `createCached`

Результат php\lang\ThreadPool

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

Результат php\lang\ThreadPool

static `createScheduled`(*$corePoolSize*)
Creates a thread pool that can schedule commands to run after a given delay, or to execute periodically.

Параметры

- `$corePoolSize` – `int`

Результат php\lang\ThreadPool

1.4.9 lib

arr

`php\lib\arr`
Library for working with collections - arrays, iterators, etc.

Methods

`__construct`()
private

static `count`(*$collection*)
Returns element count of the collection

> **Предупреждение:** for iterators it will iterate all elements to return the result

Параметры

- `$collection` – `array`, `Countable`, `Iterator`

Результат `int` element count

static `has`(*$collection*, *$value*, *$strict = false*)

Параметры

- `$collection` – `array`, `Traversable`
- `$value` – `mixed`
- `$strict` – `bool`

Результат `bool`

static `toArray`(*$collection*, *$withKeys = false*)
Converts $collection to array

Параметры

- `$collection` – `array`, `Iterator`
- `$withKeys` – `bool`

Результат `array`

static `of`(*$collection*, *$withKeys = false*)
Alias of toArray()

Параметры

- `$collection` – `array`, `Iterator`
- `$withKeys` – `bool`, `php\lib\false`

Результат `array`

static `toList`(*$collection*)
Example: items::toList(['x' => 10, 20], 30, ['x' => 50, 60]) -> [10, 20, 30, 50, 60]

Параметры

- `$collection` –

Результат `array`

static `keys`(*$collection*)
Returns all keys of collection

Параметры

- `$collection` – `array`, `Iterator`

Результат `array`

static `combine`(*$keys*, *$values*)
Combines two collections to array.

Параметры

- `$keys` – `array`, `Iterator`
- `$values` – `array`, `Iterator`

Результат `array`, `null` returns null if size of arrays is not equals.

static `map`(*$collection*, *$callback*)

Параметры

- `$collection` – `array`, `Iterator`
- `$callback` – `callable`

static `flatten`(*$collection*, *$maxLevel = -1*)
Returns a new array that is a one-dimensional flattening of this collection (recursively). That is, for every element that is an collection, extract its elements into the new array. If the optional $maxLevel argument > -1 the level of recursion to flatten.

Параметры

- `$collection` – `array`, `Iterator`
- `$maxLevel` – `int`

Результат `array`

static `sort`(*$collection*, *$comparator = null*, *$saveKeys = false*)
Sorts the specified list into ascending order

Параметры

- `$collection` – `array`, `Iterator`
- `$comparator` – `callable` - ($o1, $o2) -> int where -1 smaller, 0 equal, 1 greater
- `$saveKeys` – `bool`

Результат `array`

static `sortByKeys`(*$collection*, *$comparator = null*, *$saveKeys = false*)
Sorts the specified list into ascending order by keys

Параметры

- `$collection` – `array`, `Iterator`
- `$comparator` – `callable` - ($key1, $key2)
- `$saveKeys` – `bool`

Результат `array`

static `peak`(*$array*)
Returns the last element of array.

Параметры

- `$array` –

Результат `mixed` last value of array

static `push`(*$array*, *$values*)

Параметры

- `$array` – `array`, `ArrayAccess`
- `$values` –

static `pop`(*$array*)

Параметры

- `$array` – `array`

Результат `mixed`

static `shift`(*$array*)

Параметры

- `$array` – `array`

Результат `mixed`

static `unshift`(*$array*, *$values*)

Параметры

- `$array` – `array`
- `$values` –

static `first`(*$collection*)

Параметры

- `$collection` – `Traversable`, `array`

Результат `mixed`

static `firstKey`(*$collection*)

Параметры

- `$collection` – `Traversable`, `array`

Результат `string`, `int`, `null`

static `reverse`(*$array*)

Параметры

- `$array` – `array`

Результат `array`

bin

`php\lib\bin`

Methods

`__construct`()
private

static `of`(*$string*)

Параметры

- `$string` – `string`, `array`

Результат `string` - binary string

fs

`php\lib\fs`
File System class.

Class fs

Methods

static `separator`
Return the local filesystem's name-separator character.

Результат `string`

static `pathSeparator`
Return the local filesystem's path-separator character.

Результат `string`

static `valid(`*$name*`)`
Validate file name.

Параметры

- `$name` –

Результат `bool`

static `abs(`*$path*`)`
Returns absolute real path.

Параметры

- `$path` –

Результат `string`

static `name(`*$path*`)`

Параметры

- `$path` –

Результат `string`

static `nameNoExt(`*$path*`)`

Параметры

- `$path` –

Результат `string`

static `pathNoExt(`*$path*`)`
Returns path without extension.

Параметры

- `$path` – `string`

Результат `string`

static `ext(`*$path*`)`
Returns extension of path.

Параметры

- `$path` –

Результат `string`

static `hasExt(`*$path*, *$extensions = null*, *$ignoreCase = true*`)`
Check that $path has an extension from the extension set.

Параметры

- `$path` – `string`
- `$extensions` – `string`, `array`
- `$ignoreCase` – `bool`

Результат `bool`

static `parent`(*$path*)
Returns parent directory.

Параметры

- `$path` –

Результат `string`

static `ensureParent`(*$path*)
Checks parent of path and if it is not exists, tries to create parent directory. See makeDir().

Параметры

- `$path` – `string`

Результат `bool`

static `normalize`(*$path*)
Normalizes file path for current OS.

Параметры

- `$path` –

Результат `string`

static `exists`(*$path*)

Параметры

- `$path` –

Результат `string`

static `size`(*$path*)
Returns size of file in bytes.

Параметры

- `$path` –

Результат `int`

static `isFile`(*$path*)

Параметры

- `$path` –

Результат `bool`

static `isDir`(*$path*)

Параметры

- `$path` –

Результат `bool`

static `isHidden`(*$path*)

Параметры

- `$path` –

Результат `bool`

static `time`(*$path*)
Returns last modification time of file or directory.

Параметры

- `$path` –

Результат `int`

static `makeDir`(*$path*)
Creates empty directory (mkdirs) if not exists.

Параметры

- `$path` –

Результат `bool`

static `makeFile`(*$path*)
Creates empty file, if file already exists then rewrite it.

Параметры

- `$path` –

Результат `bool`

static `delete`(*$path*)
Deletes file or empty directory.

Параметры

- `$path` –

Результат `bool`

static `clean`(*$path*, *$checker = null*)
Deletes all files in path. This method does not delete the $path directory. Returns array with error, success and skip file list.

Параметры

- `$path` – `string`
- `$checker` – `callable` - (File $file, $depth) optional, must return true to delete the file.

Результат `array` [success => [], error => [], skip = []]

static `scan`(*$path*, *$onProgress*, *$maxDepth = 0*, *$subIsFirst = false*)

Параметры

- `$path` – `string`
- `$onProgress` – `callable` - (File $file, $depth)
- `$maxDepth` – `int` - if 0 then unlimited.
- `$subIsFirst` – `bool`

static `hash`(*$source*, *$algo = 'MD5'*)
Calculates hash of file or stream.

Параметры

- `$source` – `string`, php\io\Stream
- `$algo` – `string` - MD5, MD2, SHA-1, SHA-256, SHA-512

Результат `string`

static `copy`(*$source*, *$dest*, *$onProgress = null*)
Copies $source stream to $dest stream.

Параметры

- `$source` – `string`, php\io\File, php\io\Stream
- `$dest` – `string`, php\io\File, php\io\Stream
- `$onProgress` – `callable` - ($copiedBytes)

Результат `int` copied bytes.

static `get`(*$source*, *$charset = null*, *$mode = 'r'*)
Reads fully data from source and returns it as binary string.

Параметры

- `$source` – `string`
- `$charset` – `null`, `string` - UTF-8, windows-1251, etc.
- `$mode` – `string`

Результат `string`

num

`php\lib\num`
Utils for numbers

Class num

Methods

`__construct`()
private

static `compare`(*$num1*, *$num2*)
Compare two numbers

Примечание: it can be used as comparator for number sorting

Параметры

- `$num1` – `int`, `double`
- `$num2` – `int`, `double`

Результат `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.

Параметры

- `$number` – `int`

Результат `string`

static `toOctal`(*$number*)
Returns a string representation of the $number argument as an unsigned integer in base 8.

Параметры

- `$number` – `int`

Результат `string`

static `toHex`(*$number*)
Returns a string representation of the $number argument as an unsigned integer in base 16.

Параметры

- `$number` – `int`

Результат `string`

static `toString`(*$number*, *$radix*)
Returns a string representation of the first argument in the radix specified by the second argument.

Параметры

- `$number` – `int`
- `$radix` – `int`

Результат `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.

Параметры

- `$number` – `int`

Результат `int`

static `decode`(*$string*)
Decodes a string into a integer. Accepts decimal, hexadecimal, and octal numbers

Параметры

- `$string` – `string`

Результат `string` or false if invalid number format

static `format`(*$number*, *$pattern*, *$decSep = '.'*, *$groupSep = ','*)

Параметры

- `$number` – `int`, `double`
- `$pattern` –
- `$decSep` – `string`
- `$groupSep` – `string`

Результат `string`

reflect

`php\lib\reflect`
Class reflect

Methods

`__construct()`
private

static `typeOf(`*$object*, *$isLowerCase = false*`)`

Параметры

- `$object` – `object`
- `$isLowerCase` – `bool` - (optional)

Результат `php\lib\false`, `string`

static `typeModule(`*$typeName*`)`

Параметры

- `$typeName` – `string`

Результат php\lang\Module, `null`

static `functionModule(`*$funcName*`)`

Параметры

- `$funcName` – `string`

Результат php\lang\Module, `null`

static `newInstance(`*$className*, *$args = null*, *$withConstruct = true*`)`

Параметры

- `$className` – `string`
- `$args` – `array`
- `$withConstruct` – `bool`

Результат `object`

str

`php\lib\str`
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.

Параметры

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

Результат `int` - returns -1 if not found

static `posIgnoreCase`(*$string*, *$search*, *$fromIndex = 0*)
The same method as `pos()` only with ignoring case characters

Параметры

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

Результат `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`.

Параметры

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

Результат `int` - returns -1 if not found

static `lastPosIgnoreCase`(*$string*, *$search*, *$fromIndex = null*)
The same method as `lastPos()` only with ignoring case characters

Параметры

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

Результат `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`.

Параметры

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

Результат `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

Параметры

- `$string1` – `string` - - first string
- `$string2` – `string` - - second string

Результат `int`

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

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

Параметры

- `$string1` – `string`
- `$string2` – `string`

Результат `int`

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

Checks that the strings are equal with ignoring case characters

Параметры

- `$string1` – `string`
- `$string2` – `string`

Результат `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)

Параметры

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

Результат `bool`

static `endsWith`(*$string*, *$suffix*)

Tests if this string ends with the specified suffix.

Параметры

- `$string` – `string`
- `$suffix` – `string`

Результат `bool`

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

Параметры

- `$string` – `string`

Результат `string`

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

Параметры

- `$string` – `string`

Результат `string`

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

Параметры

- `$string` – `string`

Результат `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".

Параметры

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

Результат `string`

static `repeat`(*$string*, *$amount*)
Return s a new string consisting of the original `$string` repeated

Параметры

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

Результат `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*`)`

Параметры

- `$string` – `string`

Результат `string`

static `shuffle(`*$string*`)`
Returns a randomized string based on chars in $string

Параметры

- `$string` – `string`

Результат `string`

static `random(`*$length = 16*, *$set = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM0123456789'*

Параметры

- `$length` – `int`
- `$set` – `string`

Результат `string`

static `split(`*$string*, *$separator*, *$limit = 0*`)`
The method like `explode()` in Zend PHP

Параметры

- `$string` – `string`
- `$separator` – `string`
- `$limit` – `int`

Результат `array`

static `join(`*$iterable*, *$separator*, *$limit = 0*`)`
The method like `implode()` in Zend PHP

Параметры

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

Результат `string`

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

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

Параметры

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

Результат `string` binary string

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

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

Параметры

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

Результат `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

Параметры

- `$string` – `string`
- `$bigNumbers` – `bool`

Результат `bool`

static `isLower`(*$string*)

Параметры

- `$string` – `string`

Результат `bool`

static `isUpper`(*$string*)

Параметры

- `$string` –

Результат `bool`

static `lowerFirst`(*$string*)

Параметры

- `$string` – `string`

Результат `string`

static `upperFirst`(*$string*)

Параметры

- `$string` – `string`

Результат `string`

static `format`(*$string*, *$args*)

Параметры

- `$string` – `string`
- `$args` –

Результат `string`

static `contains`(*$string*, *$search*)

Параметры

- `$string` – `string`
- `$search` – `string`

Результат `bool`

static `count`(*$string*, *$subString*, *$offset = 0*)

Параметры

- `$string` – `string`
- `$subString` – `string`
- `$offset` – `int`

Результат `int`

static `uuid`(*$value = null*)

Параметры

- `$value` – `null`, `string`

Результат `string` uuid of $value if it is not null, else random uuid

static `hash`(*$string*, *$algorithm = 'SHA-1'*)
throws `php\lib\\Exception` if the algorithm is not supported

Параметры

- `$string` – `string`
- `$algorithm` – `string` - MD5, SHA-1, SHA-256, etc.

Результат `string`

1.4.10 mail

Email

php\mail\Email
Class Email

Methods

---

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

Параметры

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

Результат php\mail\$this

setCharset(*$charset*)

Параметры

- $charset – string

Результат php\mail\$this

setSubject(*$subject*)

Параметры

- $subject – string

Результат php\mail\$this

setTo(*$addresses*)

Параметры

- $addresses – array

Результат php\mail\$this

setCc(*$addresses*)

Параметры

- $addresses – array

Результат php\mail\$this

setBcc(*$addresses*)

Параметры

- $addresses – array

Результат php\mail\$this

setBounceAddress(*$email*)

Параметры

- $email – array

Результат php\mail\$this

`setHeaders(`*$headers*`)`

Параметры

- `$headers` – `array`

Результат `php\mail\$this`

`setMessage(`*$message*`)`

Параметры

- `$message` – `string`

Результат `php\mail\$this`

`setHtmlMessage(`*$message*`)`

Параметры

- `$message` – `string`

Результат `php\mail\$this`

`setTextMessage(`*$message*`)`

Параметры

- `$message` – `string`

Результат `php\mail\$this`

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

Параметры

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

Результат `php\mail\$this`

`send(`*$backend*`)`

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

Параметры

- `$backend` – php\mail\EmailBackend

Результат `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

Результат `string`

`setHostName(`*$hostName*`)`
protected

Параметры

- `$hostName` – `string`

`getSmtpPort()`
protected

Результат `string`

`setSmtpPort(`*$smtpPort*`)`
protected

Параметры

- `$smtpPort` – `string`

`getSslSmtpPort()`
protected

Результат `string`

`setSslSmtpPort(`*$sslSmtpPort*`)`
protected

Параметры

- `$sslSmtpPort` – `string`

`isSendPartial()`
protected

Результат `boolean`

`setSendPartial(`*$sendPartial*`)`
protected

Параметры

- `$sendPartial` – `boolean`

`getSocketTimeout()`
protected

Результат `int`

`setSocketTimeout(`*$socketTimeout*`)`
protected

Параметры

- `$socketTimeout` – `int`

`getSocketConnectionTimeout()`
protected

Результат `int`

`setSocketConnectionTimeout(`*$socketConnectionTimeout*`)`
protected

Параметры

- `$socketConnectionTimeout` – `int`

`isSslOnConnect()`
protected

Результат `boolean`

`setSslOnConnect(`*$sslOnConnect*`)`
protected

Параметры

- `$sslOnConnect` – `boolean`

`isSslCheckServerIdentity()`
protected

Результат `boolean`

`setSslCheckServerIdentity(`*$sslCheckServerIdentity*`)`
protected

Параметры

- `$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.

Параметры

- `$login` – `string`
- `$password` – `string`

`clearAuthentication()`

1.4.11 net

NetStream

`php\net\NetStream`
extends: php\io\Stream

http, ftp protocols

Class NetStream

Methods

`read(`*$length*`)`

Параметры

- `$length` –

`readFully(`*$bufferSize = 4096*`)`

Параметры

- `$bufferSize` – `int`

Результат `mixed`, `void`

`readFullyWithCallback(`*$bufferSize*, *$callback*`)`

Параметры

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

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

Параметры

- `$value` –
- `$length` –

`eof()`

`seek(`*$position*`)`

Параметры

- `$position` –

`getPosition()`

`close()`

`getUrl()`

Результат php\net\URL

`setProxy(`*$proxy*`)`

Параметры

- `$proxy` – php\net\Proxy

`getProxy()`

Результат php\net\Proxy

`getUrlConnection()`

Результат php\net\URLConnection

Proxy

`php\net\Proxy`
Class Proxy

Methods

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

Параметры

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

`address()`

Результат `string` host with port

`type()`

Результат `string` DIRECT, HTTP or SOCKS

ServerSocket

`php\net\ServerSocket`
Class SocketServer

Methods

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

Параметры

- `$port` – `int`

- `$backLog` – `int`

`accept()`
throws php\io\IOException

Результат php\net\Socket

`bind(`*$hostname*, *$port*, *$backLog = 50*`)`
throws php\net\SocketException

Параметры

- `$hostname` – `string`
- `$port` – `int`
- `$backLog` – `int`

`close()`
throws php\io\IOException

`isClosed()`

Результат `bool`

`isBound()`
Returns the binding state of the ServerSocket.

Результат `bool`

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

throws php\net\SocketException

Параметры

- `$timeout` – `int`

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

throws php\net\SocketException

Параметры

- `$on` – `bool`

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

Параметры

- `$size` – `int`

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

! Not implemented yet for TCP/IP

Параметры

- `$connectTime` – `int`
- `$latency` – `int`
- `$bandWidth` – `int`

static `findAvailableLocalPort`

Результат `int`

Socket

`php\net\Socket`
Class Socket

Methods

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

Параметры

- `$host` – `null`, `string`
- `$port` – `null`, `int`

`getOutput()`
throws php\io\IOException

Результат php\io\MiscStream

`getInput()`
throws php\io\IOException

Результат php\io\MiscStream

`getLocalAddress()`

Результат `string`

`getAddress()`

Результат `string`

`getLocalPort()`

Результат `int`

`getPort()`

Результат `int`

`close()`
throws php\io\IOException

`shutdownInput()`
throws php\io\IOException

`shutdownOutput()`
throws php\io\IOException

`isConnected()`

Результат `bool`

`isClosed()`

Результат `bool`

`isBound()`

Результат `bool`

`isInputShutdown()`

Результат `bool`

`isOutputShutdown()`

Результат `bool`

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

Параметры

- `$hostname` – `string`
- `$port` – `int`
- `$timeout` – `null`, `int`

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

throws php\net\SocketException

Параметры

- `$hostname` – `string`
- `$port` – `int`

`bindDefault()`

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

throws php\net\SocketException

Параметры

- `$timeout` – `int`

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

Параметры

- `$on` – `bool`
- `$linger` – `int`

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

throws php\net\SocketException

Параметры

- `$on` – `bool`

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

Параметры

- `$size` – `int`

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

Параметры

- `$on` – `bool`

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

Параметры

- `$on` – `bool`

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

Параметры

- `$on` – `bool`

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

Параметры

- `$size` – `int`

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

Параметры

- `$tc` – `int`

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

! Not implemented yet for TCP/IP

Параметры

- `$connectTime` – `int`
- `$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

Параметры

- `$data` – `int`

SocketException

`php\net\SocketException`
extends: php\lang\JavaException

Class SocketException

URL

php\net\URL
Class URL

Methods

__construct(*$uri*)

Параметры

- $uri – string

openConnection(*$proxy*)

Параметры

- $proxy – php\net\Proxy - (optional)

Результат php\net\URLConnection

getAuthority()
Gets the authority part of this URL

Результат string

getPort()

Результат 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.

Результат int

getProtocol()
Gets the protocol name of this URL

Результат string

getHost()

Результат string

getFile()
Gets the file name of this <code>URL</code>. 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.

Результат string

getPath()

Результат string

getQuery()

Результат string

getRef()
Gets the anchor (also known as the "reference") of this URL

Результат string

`sameFile(`*$url*`)`
Compares two URLs, excluding the fragment component.

Параметры

- `$url` – php\net\URL

Результат `bool`

`toString()`

Результат `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.

Результат `string`

`openStream()`

Результат php\io\Stream

`__toString()`

Результат `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

Параметры

- `$parent` – php\net\URLConnection

`connect()`

`getHeaderField(`*$name*`)`

Параметры

- `$name` – `string`

`getHeaderFields()`

Результат array

getInputStream()

Результат php\io\Stream

getErrorStream()

Результат php\io\Stream

getOutputStream()

Результат php\io\Stream

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

Параметры

- $name – string
- $value – string

getRequestProperty(*$name*)

Параметры

- $name – string

getRequestProperties()

Результат 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.

Параметры

- $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 <code>getContentType</code> method.

Параметры

- $stream – php\io\Stream

Результат string

static guessContentTypeFromName(*$name*)

Параметры

- $name – string

static create(*$url*, *$proxy = null*)

Параметры

- $url – string
- $proxy – php\net\Proxy

Результат php\net\URLConnection

1.4.12 orientdb

ODatabase

`php\orientdb\ODatabase`
Class ODatabase

Properties

---

property `name`
`string`

read-only

property `url`
`string`

read-only

property `size`
`int`

read-only

property `status`
`string`

read-only

OPEN, CLOSED, IMPORTING

property `type`
`string`

Methods

`__construct(`*$uri*`)`

Параметры

- `$uri` – `string`

`setUser(`*$name*`)`

Параметры

- `$name` – `string`

`setUserAndPassword(`*$name*, *$password*`)`

Параметры

- `$name` – `string`
- `$password` – `string`

`exists()`

Результат `bool`

`create()`

`drop()`

`open(`*$username*, *$password*`)`
Open database

Параметры

- `$username` – `string`
- `$password` – `string`

`close()`

`reload()`

`freeze()`

`release()`

`begin()`

`commit(`*$force = false*`)`

Параметры

- `$force` – `bool`

`rollback(`*$force = false*`)`

Параметры

- `$force` – `bool`

`getDocuments(`*$className*`)`

Параметры

- `$className` –

Результат php\util\Flow of ODocument

`query(`*$query*, *$limit = -1*`)`

Параметры

- `$query` – `string` - like sql.
- `$limit` – `int`

Результат php\util\Flow of ODocument

`command(`*$query*`)`

Параметры

- `$query` – `string`

Результат `int`

ODocument

`php\orientdb\ODocument`
Class ODocument

Properties

property `id`
`string`

read-only

Methods

`__construct(`*$className*`)`

Параметры

- `$className` – `string`

`isNew()`

Результат `bool`

`resetId()`

`save()`

`delete()`

`clear()`

`reset()`

`undo(`*$field*`)`
Undo changes.

Параметры

- `$field` –
  – (optional)

`load()`

`reload()`

`toJson()`

Результат `string`

`fromJson(`*$json*`)`

Параметры

- `$json` – `string`

`__set(`*$name*, *$value*`)`

Параметры

- `$name` –
- `$value` –

`__get(`*$name*`)`

Параметры

- `$name` –

`__isset(`*$name*`)`

Параметры

- `$name` –

`__unset(`*$name*`)`

Параметры

- `$name` –

`__clone()`

1.4.13 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

Параметры

- `$name` – `string`

Результат `string`

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

Параметры

- `$sql` – `string`
- `$arguments` – `array`

Результат 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`()

Результат `array`

`getSchemas`()

Результат `array`

`getMetaData`()

Результат `array`

SqlConnectionPool

`php\sql\SqlConnectionPool`
Class SqlConnectionPool

Methods

`__construct`(*$parent*)
protected

Параметры

- `$parent` – php\sql\SqlConnectionPool

`getConnection`()

Результат php\sql\SqlConnection

`setUser`(*$username*)

Параметры

- `$username` – `string`

Результат `php\sql\$this`

`setPassword(`*$password*`)`

Параметры

- `$password` – `string`

Результат `php\sql\$this`

`setMaxPoolSize(`*$value*`)`

Параметры

- `$value` – `int`

Результат `php\sql\$this`

`setIdleTimeout(`*$millis*`)`

Параметры

- `$millis` – `int`

Результат `php\sql\$this`

`setMaxLifetime(`*$millis*`)`

Параметры

- `$millis` – `int`

Результат `php\sql\$this`

`setMinimumIdle(`*$millis*`)`

Параметры

- `$millis` – `int`

Результат `php\sql\$this`

SqlDriverManager

`php\sql\SqlDriverManager`
Class DriverManager

Methods

---

static `install(`*$driverName*`)`
throws php\sql\SqlException if cannot install or find driver.

Параметры

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

static `getConnection(`*$url*, *$options*`)`
throws php\sql\SqlException if a database access error occurs

Параметры

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

Результат php\sql\SqlConnection

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

Параметры

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

Результат 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

Результат `bool`

`isFirst`()
throws php\sql\SqlException

Результат `bool`

`delete`()
Deletes current row.

throws php\sql\SqlException

`isDeleted`()
throws php\sql\SqlException

Результат `bool`

`refresh`()
throws php\sql\SqlException

`get`(*$column*)
throws php\sql\SqlException

Параметры

- `$column` – `string`

Результат `mixed`, php\io\Stream, php\time\Time

`toArray`(*$assoc = true*)
throws php\sql\SqlException

Параметры

- `$assoc` – `bool`

Результат `array`

SqlStatement

`php\sql\SqlStatement`
abstract class

implements: `Iterator`

Methods

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

Параметры

- `$index` – `int`
- `$value` – `mixed`

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

Параметры

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

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

Параметры

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

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

Параметры

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

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

Параметры

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

`fetch()`
throws php\sql\SqlException

Результат php\sql\SqlResult

`update()`
throws php\sql\SqlException

Результат `int`

`getLastInsertId()`

Результат mixed

getGeneratedKeys()

Результат php\sql\SqlResult

current()

Результат php\sql\SqlResult

next()

key()

valid()

rewind()

1.4.14 swing

Border

php\swing\Border
Class Border

Methods

---

isOpaque()

Результат bool

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

Параметры

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

Результат php\swing\Border

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

Параметры

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

Результат php\swing\Border

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

Параметры

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

Результат php\swing\Border

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

Параметры

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

Результат php\swing\Border

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

Параметры

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

Результат php\swing\Border

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

Параметры

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

Результат php\swing\Border

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

Параметры

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

Результат 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*`)`

Параметры

- `$rgb` – `int`
- `$hasAlpha` – `bool`

`darker()`
Creates a new Color that is a darker version of this

Результат php\swing\Color

`brighter()`
Creates a new Color that is a brighter version of this

Результат php\swing\Color

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

Параметры

- `$r` – `int` - - 0 .. 255
- `$g` – `int` - - 0 .. 255
- `$b` – `int` - - 0 .. 255
- `$alpha` – `int` - - 0 .. 255

Результат php\swing\Color

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

Параметры

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

Результат php\swing\Color

static `decode(`*$nm*`)`
Decode color from a hex string (#RGB, 0xRGB, etc...)

Параметры

- `$nm` –

Результат 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()`

Результат `bool`

`isDeselected()`

Результат `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

MouseWheelEvent

`php\swing\event\MouseWheelEvent`
extends: php\swing\event\MouseEvent

Events:

- mouseWheel

Class MouseWheelEvent

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*`)`

Параметры

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

`isBold()`

Результат `bool`

`isItalic()`

Результат `bool`

`isPlain()`

Результат `bool`

`isTransformed()`

Результат `bool`

`getBaselineFor(`*$symbol*`)`

Параметры

- `$symbol` – `string` - - one char

Результат `int`

`canDisplay(`*$symbol*`)`

Параметры

- `$symbol` – `string` - - one char

Результат `bool`

`canDisplayUpTo(`*$string*`)`
Indicates whether or not this Font can display a specified String.

Параметры

- `$string` – `string`

Результат `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

Параметры

- `$str` – `string`

Результат php\swing\Font

static `create(`*$source*, *$trueType = true*`)`
Create new font by using Stream or File

throws php\io\IOException

throws `php\swing\\Exception`

Параметры

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

Результат php\swing\Font

static `get(`*$name*`)`
Get font by name

Параметры

- `$name` – `string`

Результат 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

Параметры

- `$str` –

Результат `int`

`getTextHeight()`
Return height of one line text with current font

Результат `int`

`setPaintMode()`

`drawLine(`*$x1*, *$y1*, *$x2*, *$y2*`)`
Draw line

Параметры

- `$x1` – `int`
- `$y1` – `int`
- `$x2` – `int`
- `$y2` – `int`

`drawRect(`*$x*, *$y*, *$width*, *$height*`)`
Draw rect

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

`fillRect(`*$x*, *$y*, *$width*, *$height*`)`

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

`draw3DRect(`*$x*, *$y*, *$width*, *$height*, *$raised*`)`
Draws a 3-D highlighted outline of the specified rectangle.

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

- $raised –

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

Параметры

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

drawOval(*$x*, *$y*, *$width*, *$height*)
Draw oval

Параметры

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

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

Параметры

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

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

Параметры

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

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

Параметры

- $x – int
- $y – int
- $width – int
- $height – int
- $startAngle – int

- `$arcAngle` – `int`

`drawPolygon(`*$xy*`)`

Параметры

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

`fillPolygon(`*$xy*`)`

Параметры

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

`drawPolyline(`*$xy*`)`

Параметры

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

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

Параметры

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

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

Параметры

- `$text` – `string`
- `$x` – `int`
- `$y` – `int`

`clipRect(`*$x*, *$y*, *$width*, *$height*`)`
Intersects the current clip with the specified rectangle.

Параметры

- `$x` – `int`
- `$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.

Параметры

- `$x` – `int`
- `$y` – `int`
- `$width` – `int`
- `$height` – `int`

`setXORMode(`*$color*`)`

Параметры

- `$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.

Параметры

- `$x` – `int`
- `$y` – `int`

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

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

Параметры

- `$x` – `int`
- `$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

Параметры

- `$x` – `int`, `null`
- `$y` – `int`, `null`
- `$w` – `int`, `null`
- `$h` – `int`, `null`

Результат 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 = ::*)

Параметры

- $width – int
- $height – int
- $type – int

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

Параметры

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

Результат php\swing\Image

getRGB(*$x*, *$y*)

Параметры

- $x – int
- $y – int

Результат `int`

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

Параметры

- `$x` – `int`
- `$y` – `int`
- `$rgb` – `int` - - color

`getGraphics()`

Результат php\swing\Graphics

`getProperty(`*$name*`)`

Параметры

- `$name` –

Результат php\lang\JavaObject, `null`

static `read(`*$stream*`)`
throws php\io\IOException

Параметры

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

Результат php\swing\Image

static `write(`*$image*, *$format*, *$stream*`)`
throws php\io\IOException

Параметры

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

Scope

`php\swing\Scope`
Class Scope

Methods

---

static `getDefault`

Результат php\swing\Scope

ScopeValue

`php\swing\ScopeValue`
final class

Methods

`__construct()`
private

`getValue()`

Результат `mixed`

`setValue(`*$value*`)`

Параметры

- `$value` –

`bind(`*$object*, *$property*`)`

Параметры

- `$object` – `object`
- `$property` – `string`

static `of(`*$name*`)`

Параметры

- `$name` – `string`

Результат php\swing\ScopeValue

SwingUtilities

`php\swing\SwingUtilities`
final class

Methods

`__construct()`
private

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

Результат `array`

static `setExceptionHandler(`*$handler*`)`

Параметры

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

static `invokeLater(`*$runner*`)`

Параметры

- `$runner` – `callable`

SwingWorker

`php\swing\SwingWorker`
abstract class

Methods

`doInBackground()`
abstract

protected

Результат `mixed`

`get(`*$timeout = -1*`)`
throws php\concurrent\TimeoutException

throws php\lang\InterruptedException

Параметры

- `$timeout` – `int`

Результат `mixed`

`getProgress()`

Результат `int`

`setProgress(`*$val*`)`
protected

Параметры

- `$val` – `int`

`publish(`*$values*`)`
protected

Параметры

- `$values` – `array`

`process(`*$values*`)`
protected

Параметры

- `$values` – `array`

`isDone()`

Результат `bool`

`isCanceled()`

Результат `bool`

`getState()`

Результат `string` PENDING, STARTED, DONE

`cancel(`*$mayInterruptIfRunning*`)`

Параметры

- `$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*)

Параметры

- `$delay` – `int`
- `$callback` – `callable`

`start`()

`stop`()

`restart`()

`isRunning`()

Результат `bool`

tree

TreeModel

php\swing\tree\TreeModel
Class TreeModel

Properties

---

property `root`
php\swing\tree\TreeNode

Methods

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

Параметры

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

`nodeChanged`(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

`nodeStructureChanged`(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

`removeNodeFromParent`(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

`reload`(*$node = null*)

Параметры

- `$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*`)`

Параметры

- `$object` – `mixed`
- `$allowsChildren` – `bool`

`isRoot()`

Результат `bool`

`isLeaf()`

Результат `bool`

`getRoot()`

Результат php\swing\tree\TreeNode

`isNodeChild(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`isNodeAncestor(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`isNodeDescendant(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`isNodeRelated(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`isNodeSibling(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`getNextNode()`

Результат php\swing\tree\TreeNode, `null`

`getNextLeaf()`

Результат php\swing\tree\TreeNode, `null`

`getNextSibling()`

Результат php\swing\tree\TreeNode, `null`

`getPreviousNode()`

Результат php\swing\tree\TreeNode, `null`

`getPreviousLeaf()`

Результат php\swing\tree\TreeNode, `null`

`getPreviousSibling()`

Результат php\swing\tree\TreeNode, `null`

`getFirstChild()`

Результат php\swing\tree\TreeNode, `null`

`getFirstLeaf()`

Результат php\swing\tree\TreeNode, `null`

`getLastChild()`

Результат php\swing\tree\TreeNode, `null`

`getLastLeaf()`

Результат php\swing\tree\TreeNode, `null`

`add(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

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

Параметры

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

`insertAfter(`*$child*, *$node*`)`
throws >

Параметры

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

`insertBefore(`*$child*, *$node*`)`
throws >

Параметры

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

`remove(`*$child*`)`

Параметры

- `$child` – php\swing\tree\TreeNode

`removeByIndex(`*$childIndex*`)`

Параметры

- `$childIndex` – `int`

`removeAllChildren()`

`removeFromParent()`

`getIndex(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `int`

`getChild(`*$index*`)`

Параметры

- `$index` – `int`

Результат php\swing\tree\TreeNode, `null`

`getChildCount()`

Результат `int`

`duplicate()`

Результат 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\int|string`

Direction

property `iconTextGap`
`int`

property `borderPainted`
`bool`

property `focusPainted`
`bool`

property `rolloverEnabled`
`bool`

property `contentAreaFilled`
`bool`

property `buttonGroup`
`string`

Methods

`setIcon(`*$icon*`)`

Параметры

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

`setDisabledIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

`setSelectedIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

`setPressedIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

`setRolloverIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

`setDisabledSelectedIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

`setRolloverSelectedIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

`doClick(`*$pressTime = 68*`)`

Параметры

- `$pressTime` – `int` - the time to "hold down" the button, in milliseconds

static `getButtons(`*$buttonGroup*`)`

Параметры

- `$buttonGroup` – `string`

Результат php\swing\UIAbstractIButton[]

static `getSelectedButtons(`*$buttonGroup*`)`

Параметры

- `$buttonGroup` – `string`

Результат php\swing\UIAbstractIButton[]

UIButton

`php\swing\UIButton`
extends: php\swing\UIAbstractIButton

Class UIButton

UICheckbox

`php\swing\UICheckbox`
extends: php\swing\UIAbstractIButton

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*)

Параметры

- $items – array

`getItem(`*$index*`)`

Параметры

- `$index` – `int`

Результат `string`, `null`

`addItem(`*$item*`)`

Параметры

- `$item` – `string`

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

Параметры

- `$index` – `int`
- `$item` – `string`

`removeItem(`*$index*`)`

Параметры

- `$index` – `int`

`removeAllItems()`

`onCellRender(`*$callback = null*`)`

Параметры

- `$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\UIAbstractIButton
- **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

Параметры

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

`setLayout(`*$type*`)`

Параметры

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

`remove(`*$component*`)`

Параметры

- `$component` – php\swing\UIElement

`removeByIndex(`*$index*`)`
throws >

Параметры

- `$index` – `int`

`removeAll()`

`getComponentCount()`

Результат `int`

`getComponent(`*$index*`)`
throws >

Параметры

- `$index` – `int`

Результат php\swing\UIElement

`getComponents()`

Результат php\swing\UIElement[]

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

Параметры

- `$group` – `string`

Результат php\swing\UIElement, `NULL`

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

Параметры

- `$group` – `string`

Результат 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*)

Параметры

- $owner – php\swing\UIWindow

showModal()

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

Параметры

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

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

Параметры

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

Результат int

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

Параметры

- $message – string
- $initialValue – string

Результат string

UIEdit

php\swing\UIEdit
extends: php\swing\UITextElement

Class UIEdit

Properties

---

property `columns`
`int`

property `scrollOffset`
`int`

UIEditorArea

`php\swing\UIEditorArea`
extends: php\swing\UITextElement

Class UIEditorArea

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

Результат 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

Параметры

- `$name` –

  – * name of event

- `$callback` – `callable`

- `$group` – `string`

`off(`*$name*, *$group = NULL*`)`
Remove all event callbacks (if group == null), or only group

Параметры

- `$name` –
- `$group` – `null`, `string`

Результат `bool`

`trigger(`*$name*`)`
Trigger callback by event name

Параметры

- `$name` –

`addAllowedEventType(`*$name*`)`
protected

Параметры

- `$name` – `string`

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

Параметры

- `$name` – `string`
- `$callback` – `callable`

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

Параметры

- `$keyString` – `string`
- `$actionName` – `string`

`hasFocus()`

Результат `bool`

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

Параметры

- `$component` – php\swing\UIElement
- `$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.

Параметры

- `$x` –
- `$y` –

Результат php\swing\UIElement

`printOne(`*$canvas*`)`
Prints this component.

Параметры

- `$canvas` – php\swing\Graphics

`printAll(`*$canvas*`)`
Prints this component and all of its subcomponents.

Параметры

- `$canvas` – php\swing\Graphics

`paintOne(`*$canvas*`)`
Paints this component.

Параметры

- `$canvas` – php\swing\Graphics

`paintAll(`*$canvas*`)`
Paints this component and all of its subcomponents.

Параметры

- `$canvas` – php\swing\Graphics

`updateUI()`

`invalidate()`

`repaint()`

`revalidate()`

`repaintRegion(`*$x*, *$y*, *$w*, *$h*`)`

Параметры

- `$x` – `int`
- `$y` – `int`
- `$w` – `int`
- `$h` – `int`

`grabFocus()`

`getTextWidth(`*$str*`)`
Return width of str for drawText + current font

Параметры

- `$str` –

Результат `int`

`getTextHeight()`
Return height of one line text with current font

Результат `int`

`show()`

`hide()`

`toggle()`

`removeSelf()`

static `getByUid(`*$uid*`)`
Get component by unique id

Параметры

- `$uid` – `string`

Результат 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*`)`

Параметры

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

Результат `bool`

`showSaveDialog(`*$parent = null*`)`

Параметры

- `$parent` – php\swing\UIWindow

Результат `bool`

`showOpenDialog(`*$parent = null*`)`

Параметры

- `$parent` – php\swing\UIWindow

Результат `bool`

`addChoosableFilter(`*$filter*, *$description*`)`

Параметры

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

`addChoosableExtensions(`*$extensions*, *$description*, *$showDirectories = true*`)`

Параметры

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

`resetChoosableFilters()`

`isTraversable(`*$file*`)`

Параметры

- `$file` – `string`, php\io\File

Результат `bool`

`ensureFileIsVisible(`*$file*`)`

Параметры

- `$file` – `string`, php\io\File

`approveSelection()`

`cancelSelection()`

`changeToParentDirectory()`

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

Параметры

- `$field` – `string` - - name (string), description (string), icon (Image), traversable (bool)
- `$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*`)`

Параметры

- `$action` – `int`

`setIconImage(`*$image = null*`)`

Параметры

- `$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*`)`

Параметры

- `$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*`)`

Параметры

- `$panel` – php\swing\UIDesktopPanel

`setContent(`*$content*`)`

Параметры

- `$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\int|string`

Direction

property `iconTextGap`
`int`

Methods

`setLabelFor(`*$component*`)`

Параметры

- `$component` – php\swing\UIElement

`setIcon(`*$icon*`)`

Параметры

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

`setDisabledIcon(`*$icon*`)`

Параметры

- `$icon` – php\swing\Image, `string`

UIListbox

`php\swing\UIListbox`
extends: php\swing\UIContainer

Class UIListbox

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*)

Параметры

- $items – array

getItem(*$index*)

Параметры

- $index – int

Результат string, null

addItem(*$item*)

Параметры

- $item – string

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

Параметры

- $index – int
- $item – string

removeItem(*$index*)

Параметры

- $index – int

removeAllItems()

`onCellRender(`*$callback = null*`)`

Параметры

- `$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

Параметры

- `$name` –

  – * name of theme

static `getSystemLookAndFeel`
Get name of OS native look and feel

Результат `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*`)`

Параметры

- `$pos` – `int`

Результат php\swing\UIMenuItem

`addSeparator()`

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

Параметры

- `$x` – `int`
- `$y` – `int`

`isTopLevelMenu()`

Результат `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*`)`

Параметры

- `$width` – `int`
- `$height` – `int`

`setSelected(`*$component*`)`

Параметры

- `$component` – php\swing\UIElement

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

Параметры

- `$invoker` – php\swing\UIElement
- `$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*`)`

Параметры

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

Результат php\swing\UIElement

`onRead(`*$handle = NULL*`)`

Параметры

- `$handle` – `callable` - (UIElement $el, $var)

`onTranslate(`*$handle = NULL*`)`

Параметры

- `$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*`)`

Параметры

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

Результат php\swing\text\Style

`getStyle(`*$name*`)`

Параметры

- `$name` – `string`

Результат php\swing\text\Style

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

Параметры

- `$text` – `string`
- `$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*)

Параметры

- `$height` – `int`
- `$row` – `null`, `int`

`getRowHeight`(*$row = null*)

Параметры

- `$row` – `null`, `int`

Результат `int`

`setValueAt`(*$value*, *$row*, *$column*)

Параметры

- `$value` – `string`, `null`
- `$row` – `int`
- `$column` – `int`

`getValueAt`(*$row*, *$column*)

Параметры

- `$row` – `int`
- `$column` – `int`

Результат `string`, `null`

`columnAtPoint`(*$x*, *$y*)

Параметры

- `$x` – `int`
- `$y` – `int`

Результат `int`

`rowAtPoint`(*$x*, *$y*)

Параметры

- `$x` – `int`
- `$y` – `int`

Результат `int`

`editCellAt(`*$row*, *$column*`)`

Параметры

- `$row` – `int`
- `$column` – `int`

Результат `bool`

`addColumnSelectionInterval(`*$index0*, *$index1*`)`

Параметры

- `$index0` – `int`
- `$index1` – `int`

`addRowSelectionInterval(`*$index0*, *$index1*`)`

Параметры

- `$index0` – `int`
- `$index1` – `int`

`getColumnName(`*$column*`)`

Параметры

- `$column` – `int`

Результат `string`

`setEditingColumn(`*$column*`)`

Параметры

- `$column` – `int`

`setEditingRow(`*$row*`)`

Параметры

- `$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*`)`

Параметры

- `$title` – `string`
- `$component` – php\swing\UIElement
- `$icon` – php\swing\Image

`getTitleAt(`*$index*`)`

Параметры

- `$index` – `int`

Результат `string`

`setTitleAt(`*$index*, *$value*`)`

Параметры

- `$index` – `int`
- `$value` – `string`

`getToolTipTextAt(`*$index*`)`

Параметры

- `$index` – `int`

Результат `string`

`setToolTipTextAt(`*$index*, *$value*`)`

Параметры

- `$index` – `int`
- `$value` – `string`

`getTabIconAt(`*$index*`)`

Параметры

- `$index` – `int`

Результат php\swing\Image

`setTabIconAt(`*$index*, *$image*`)`

Параметры

- `$index` – `int`

- `$image` – php\swing\Image

`getTabComponentAt(`*$index*`)`

Параметры

- `$index` – `int`

Результат php\swing\UIElement

`setTabComponentAt(`*$index*, *$component*`)`

Параметры

- `$index` – `int`
- `$component` – php\swing\UIElement

`removeTabAt(`*$index*`)`

Параметры

- `$index` – `int`

`removeAll()`

`indexAtPosition(`*$x*, *$y*`)`

Параметры

- `$x` – `int`
- `$y` – `int`

Результат `int`

`isEnabledAt(`*$index*`)`

Параметры

- `$index` – `int`

Результат `bool`

`setEnabledAt(`*$index*, *$enabled*`)`

Параметры

- `$index` – `int`
- `$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*`)`

Параметры

- `$selStart` – `int`
- `$selEnd` – `int`

`selectAll()`

`replaceSelection(`*$content*`)`

Параметры

- `$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.

Результат `bool`

UIToggleButton

`php\swing\UIToggleButton`
extends: php\swing\UIAbstractIButton

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*`)`

Параметры

- `$w` – `null`, `int`
- `$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*`)`

Параметры

- `$renderer` – `callable`

addSelectionNode(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

removeSelectionNode(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

expandNode(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

expandRow(*$row*)

Параметры

- `$row` – `int`

collapseNode(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

collapseRow(*$row*)

Параметры

- `$row` – `int`

expandNodeAll(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

collapseNodeAll(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

isExpandedRow(*$row*)

Параметры

- `$row` – `int`

Результат `bool`

isExpandedNode(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

isNodeSelected(*$node*)

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`isNodeEditable(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`isVisible(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`hasBeenExpanded(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

Результат `bool`

`fireTreeExpanded(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

`fireTreeCollapsed(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

`fireTreeWillExpand(`*$node*`)`
throws >

Параметры

- `$node` – php\swing\tree\TreeNode

`fireTreeWillCollapse(`*$node*`)`
throws >

Параметры

- `$node` – php\swing\tree\TreeNode

`makeVisible(`*$node*`)`

Параметры

- `$node` – php\swing\tree\TreeNode

`scrollToNode(`*$node*`)`

Параметры

- `$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()

Результат bool

isAlwaysOnTopSupported()

Результат bool

moveToCenter()

1.4.15 time

Time

php\time\Time
Class Time, Immutable

Methods

`__construct`(*$date*, *$timezone = null*)

Параметры

- `$date` – `int` - unix long timestamp (in millis)
- `$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.

Результат `int`

`getTimeZone`()
Get timezone of the time object

Результат php\time\TimeZone

`year`()
Get the current year

Результат `int`

`month`()
Get the current month of the year, 1 - Jan, 12 - Dec

Результат `int`

`week`()
Get week of year

Результат `int`

`weekOfMonth`()
Get week of month

Результат `int`

`day`()
Get day of year

Результат `int`

`dayOfMonth`()
Get day of month

Результат `int`

`dayOfWeek`()
Get day of week

Результат `int`

`dayOfWeekInMonth`()

Результат `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.

Результат `int`

`hourOfDay()`
Get hour of the day

Результат `int`

`minute()`
Get minute of the hour

Результат `int`

`second()`
Get second of the minute

Результат `int`

`millisecond()`
Get millisecond of the second

Результат `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.

Параметры

- `$time` – php\time\Time

Результат `int`

`withTimeZone(`*$timeZone*`)`

Параметры

- `$timeZone` – php\time\TimeZone

Результат php\time\Time

`add(`*$args*`)`
Get a new time + $args

use negative values to minus

Параметры

- `$args` – `array` - [millis, sec, min, hour, day, month, year]

Результат php\time\Time

`replace(`*$args*`)`
Clones the current datetime and replaces some fields to new values $args

Параметры

- `$args` – `array` - [millis, sec, min, hour, day, month, year]

Результат 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

Параметры

- `$format` – `string` - date time format

Результат `string`

`__toString()`
Format the time to yyyy-MM-dd'T'HH:mm:ss

Результат `string`

`__clone()`
private

static `now(`*$timeZone = null*`)`
Returns now time object (date + time)

Параметры

- `$timeZone` – php\time\TimeZone

Результат php\time\Time

static `today(`*$timeZone = null*`)`
Returns today date (without time)

Параметры

- `$timeZone` – php\time\TimeZone

Результат 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
```

Параметры

- `$args` – `array` - [millis, sec, min, hour, day, month, year]
- `$timeZone` – php\time\TimeZone - if null then it uses the default timezone

Результат php\time\Time

static `seconds`
Returns the current time in seconds (like the `millis()` method only in seconds)

Результат `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.

Результат `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

Результат `int`

TimeFormat

`php\time\TimeFormat`
Class TimeFormat, Immutable

Methods

`__construct`(*$format*, *$locale = null*, *$formatSymbols = null*)

Параметры

- `$format` – `string`
- `$locale` – php\util\Locale - if `null` then it uses the default locale
- `$formatSymbols` – `array` - [months => [...], short_months, eras, weekdays, short_weekdays, local_pattern_chars]

`format`(*$time*)

Параметры

- `$time` – php\time\Time

Результат `string`

`parse(`*$string*, *$timeZone = null*`)`

Параметры

- `$string` – `string`
- `$timeZone` – php\time\TimeZone

Результат 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*`)`

Параметры

- `$rawOffset` – `int`
- `$ID` – `string`
- `$options` – `array`

Результат php\time\TimeZone

`getId()`
Get id of the timezone

Результат `string`

`getRawOffset()`
Get raw offset of the timezone

Результат `string`

`__clone()`
private

static `UTC`
Returns UTC Time zone

Результат php\time\TimeZone

static `of(`*$ID*`)`

Параметры

- `$ID` – `string` - code of timezone, e.g.: 'UTC'

Результат php\time\TimeZone

static `setDefault`(*$zone*, *$globally = false*)
Set default time zone for Time objects, by default - the default timezone is UTC

Параметры

- `$zone` – php\time\TimeZone
- `$globally` – `bool`

static `getDefault`(*$globally = false*)
Get default timezone

Параметры

- `$globally` – `bool` - if `false` - only for the current environment

Результат php\time\TimeZone

static `getAvailableIDs`(*$rawOffset = null*)
Returns all available ids of timezones

Параметры

- `$rawOffset` – `int`, `null`

Результат `string[]`

1.4.16 util

Configuration

`php\util\Configuration`
Class Configuration

Methods

`__construct`(*$source = null*, *$encoding = 'UTF-8'*)

Параметры

- `$source` – `string`, php\io\Stream
- `$encoding` – `string`

`has`(*$key*)

Параметры

- `$key` – `string`

Результат `bool`

`get`(*$key*, *$def = null*)

Параметры

- `$key` – `string`
- `$def` – `null`, `string`

Результат `string`

`getArray`(`$key, $def = []`)

Параметры

- $key – string
- $def – array

Результат string[]

getBoolean(*$key*, *$def = false*)

Параметры

- $key – string
- $def – bool

Результат bool

getNumber(*$key*, *$def = 0*)

Параметры

- $key – string
- $def – int, float

Результат int, float

getInteger(*$key*, *$def = 0*)

Параметры

- $key – string
- $def – int

Результат int

set(*$key*, *$value*)

Параметры

- $key – string
- $value – string, array

Результат string old value

put(*$values*)

Параметры

- $values – array, Traversable

clear()

load(*$in*, *$encoding = 'UTF-8'*)

Параметры

- $in – string, php\io\Stream
- $encoding – string

save(*$out*, *$encoding = 'UTF-8'*)

Параметры

- $out – string, php\io\Stream
- $encoding – string

toArray()

Результат `array`

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

Параметры

- `$collection` – `array`, `Iterator`

`withKeys()`

Enables to save keys for the next operation

Результат php\util\Flow

`onlyKeys(`*$keys*, *$ignoreCase = false*`)`

Параметры

- `$keys` – `array`, `Traversable`
- `$ignoreCase` – `bool`

Результат 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

Параметры

- `$collection` – `array`, `Iterator`

Результат php\util\Flow

`find(`*$filter = null*`)`

Finds elements by using the $filter callback, elements - for each iteration that returns `true`

Параметры

- `$filter` – `callable`

Результат php\util\Flow

`findOne(`*$filter = null*`)`

Finds the first element by using the $filter callback, when $filter will return the first `true`

Параметры

- `$filter` – `callable`

Результат `mixed`

`findValue(`*$value*, *$strict = false*`)`

Параметры

- `$value` –
- `$strict` – `bool`

Результат `int`, `null`, `string` return null if not found, else - key of value

`group(`*$callback*`)`

Параметры

- `$callback` – `callable`

Результат php\util\Flow

`each(`*$callback*`)`
Iterates elements. It will break if $callback returns `false` strongly

Параметры

- `$callback` – `callable` - ($el[, $key]): bool

Результат `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

Параметры

- `$sliceSize` – `int`
- `$callback` – `callable` - (array $items): bool
- `$withKeys` – `bool`

Результат `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

Параметры

- `$callback` – `callable` - ($el[, $key])

Результат php\util\Flow

`keys()`
Create a new flow by using the keys of the current flow

Результат php\util\Flow

`skip(`*$n*`)`
Skips $n elements in the current collection

Параметры

- `$n` – `int` - skip count

Результат php\util\Flow

`limit(`*$count*`)`
Limits collection with $count

Параметры

- `$count` – `int` - count of limit

Результат php\util\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 });

Параметры

- `$callback` – `callable` - ($result, $el[, $key])

Результат `int`

`sort(`*$comparator = null*`)`
Sort the last result of the flow, also see: `php\\lib\\items::sort()`

Примечание: use the `withKeys()` method to save keys

Параметры

- `$comparator` – `callable` - ($o1, $o2) -> int, where -1 smaller, 0 equal, 1 greater

Результат `array`

`sortByKeys(`*$comparator = null*`)`
The same method as `sort()` only based on keys insteadof values

Примечание: use the `withKeys()` method to save keys

Параметры

- `$comparator` – `callable` - ($key1, $key2) -> int

Результат `array`

`toArray()`
Convert elements to an array

Примечание: use the `withKeys()` method to save keys

Результат `array`

`toString(`*$separator*`)`
Join elements to a string similar to `implode()` in PHP

Параметры

- `$separator` – `string`

Результат `string`

`count()`

Результат `int`

`current()`

Результат `mixed`

`next()`

Результат `void`

`key()`

Результат `mixed`

`valid()`

Результат `bool`

`rewind()`

Результат `void`

`__clone()`
private

static `ofEmpty`

Результат php\util\Flow

static `of(`*$collection*`)`
Creates a new flow for an array of Iterator

Параметры

- `$collection` – `array`, `Traversable`

Результат php\util\Flow

static `ofRange(`*$from*, *$to*, *$step = 1*`)`
Creates a new flow for a number range

Параметры

- `$from` – `int`
- `$to` – `int`
- `$step` – `int`

Результат php\util\Flow

static `ofString(`*$string*, *$chunkSize = 1*`)`
Creates a new flow for the string

Параметры

- `$string` – `string`
- `$chunkSize` – `int` - how many characters to combine for one item ?

Результат php\util\Flow

static `ofStream(`*$stream*, *$chunkSize = 1*`)`
Creates a new flow for the Stream object

Параметры

- `$stream` – php\io\Stream - stream object
- `$chunkSize` – `int` - size for `Stream.read($size)` method

Результат php\util\Flow

LauncherClassLoader

`php\util\LauncherClassLoader`
extends: php\lang\ClassLoader

Class LauncherClassLoader

Methods

`loadClass(`*$name*`)`

Параметры

- `$name` – `string`

Locale

`php\util\Locale`
Class Locale, Immutable

Methods

`__construct(`*$lang*, *$country = ''*, *$variant = ''*`)`

Параметры

- `$lang` – `string`
- `$country` – `string`
- `$variant` – `string`

`getLanguage()`

Результат `string`

`getDisplayLanguage(`*$locale = null*`)`

Параметры

- `$locale` – php\util\Locale

Результат `string`

`getCountry()`

Результат `string`

`getDisplayCountry(`*$locale = null*`)`

Параметры

- `$locale` – php\util\Locale

Результат `string`

`getVariant()`

Результат `string`

`getDisplayVariant(`*$locale = null*`)`

Параметры

- `$locale` – php\util\Locale

Результат `string`

`getISO3Country()`

Результат `string`

`getISO3Language()`

Результат `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

Результат `string`

`__clone()`

private

static `ENGLISH`

Результат php\util\Locale

static `US`

Результат php\util\Locale

static `UK`

Результат php\util\Locale

static `CANADA`

Результат php\util\Locale

static `CANADA_FRENCH`

Результат php\util\Locale

static `FRENCH`

Результат php\util\Locale

static `FRANCE`

Результат php\util\Locale

static `ITALIAN`

Результат php\util\Locale

static `ITALY`

Результат php\util\Locale

static `GERMAN`

Результат php\util\Locale

static `GERMANY`

Результат php\util\Locale

static `JAPAN`

Результат php\util\Locale

static `JAPANESE`

Результат php\util\Locale

static `KOREA`

Результат php\util\Locale

static `KOREAN`

Результат php\util\Locale

static `CHINA`

Результат php\util\Locale

static `CHINESE`

Результат php\util\Locale

static `TAIWAN`

Результат php\util\Locale

static `RUSSIAN`

Результат php\util\Locale

static `RUSSIA`

Результат php\util\Locale

static `ROOT`

Результат php\util\Locale

static `getDefault`(*$globally = false*)
Get default locale (if globally = false - only for the current environment)

Параметры

- `$globally` – `bool`

Результат php\util\Locale

static `setDefault`(*$locale*, *$globally = false*)
Set default locale

Параметры

- `$locale` – php\util\Locale
- `$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

Результат php\util\Locale[] An array of installed locales.

Promise

`php\util\Promise`

Methods

`__construct()`
private

static `create(`*$initialValue*, *$callback*`)`

Параметры

- `$initialValue` – `mixed` - (optional)
- `$callback` – `callable` - ($value) - assign callback

Результат `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

Результат `string`

`getFlags()`
Get the current flags

Результат `int`

static `of`(*$pattern*, *$flag = 0*)
Creates a new Regex of regex with $string and $flag

throws php\util\RegexException

Параметры

- `$pattern` – `string` - regular expression
- `$flag` – `int` - Regex::CASE_INSENSITIVE and other constants

Результат php\util\Regex

`matches`()
Attempts to match the entire region against the pattern.

Результат `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

Параметры

- `$start` – `int`, `null`

Результат `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

Параметры

- `$replacement` – `string`

Результат `string`

`replaceFirst`(*$replacement*)
Replaces the first subsequence of the input sequence that matches the pattern with the given replacement string.

throws php\util\RegexException

Параметры

- `$replacement` – `string`

Результат `string`

`replaceGroup`(*$group*, *$replacement*)
throws php\util\RegexException

Параметры

- `$group` – `int`
- `$replacement` – `string`

Результат `string`

`replaceWithCallback(`*$callback*`)`
throws php\util\RegexException

Параметры

- `$callback` – `callable` - (Regex $pattern) -> string

Результат `string`

`with(`*$string*`)`
Duplicates this pattern with a new $string

Параметры

- `$string` – `string`

Результат php\util\Regex

`withFlags(`*$flags*`)`
Clone this object with the new $flags

Параметры

- `$flags` – `int`

Результат php\util\Regex

`group(`*$group = null*`)`
Returns the input subsequence captured by the given group during the previous match operation.

throws php\util\RegexException

Параметры

- `$group` – `null`, `int`

Результат `string`

`getGroupCount()`
Returns the number of capturing groups in this matcher's pattern.

Результат `int`

`start(`*$group = null*`)`
Returns the start index of the previous match.

throws php\util\RegexException

Параметры

- `$group` – `null`, `int`

Результат `int`

`end(`*$group = null*`)`
Returns the offset after the last character matched.

throws php\util\RegexException

Параметры

- `$group` – `null`, `int`

Результат `int`

`hitEnd()`
Returns true if the end of input was hit by the search engine in the last match operation performed by this matcher.

Результат `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.

Результат `bool`

`lookingAt()`
Attempts to match the input sequence, starting at the beginning of the region, against the pattern.

Результат `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

Параметры

- `$start` – `int`
- `$end` – `int`

Результат 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).

Результат `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).

Результат `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.

Параметры

- `$string` – `null`, `string` - The new input character sequence

Результат `php\util\$this`

`current()`

Результат `null`, `string`

`next()`

`key()`

Результат `int`

`valid()`

Результат `bool`

`rewind()`

`__clone()`
private

static `match(`*$pattern*, *$string*, *$flags = 0*`)`
Tells whether or not this string matches the given regular expression. See also java.lang.String.matches()

Параметры

- `$pattern` – `string` - regular expression
- `$string` – `string`
- `$flags` – `int`

Результат `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

Параметры

- `$pattern` – `string` - the delimiting regular expression
- `$string` – `string`
- `$limit` – `int` - the result threshold

Результат `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.

Параметры

- `$string` – `string` - The string to be literalized

Результат `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.

Параметры

- `$string` – `string`

Результат `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

Параметры

- `$source` – `string`, php\io\Stream
- `$charset` – `string`, `null` - e.g.: UTF-8, windows-1251, etc., only for Stream objects

`hasNext(`*$pattern = null*`)`

Параметры

- `$pattern` – php\util\Regex

Результат `bool`

`next(`*$pattern = null*`)`

Параметры

- `$pattern` – php\util\Regex

Результат `string`, `null` null if doesn’t has the next pattern

`nextLine()`

Результат `string`, `null` null if doesn’t has the next line

`hasNextLine()`

Результат `bool`

`nextInt(`*$radix = null*`)`

Параметры

- `$radix` – `null`, `int` - if null then uses the default radix

Результат `int`, `null` null if doesn't has the next int

`hasNextInt(`*$radix = null*`)`

Параметры

- `$radix` – `null`, `int` - if null then uses the default radix

Результат `bool`

`nextDouble()`

Результат `float`, `null` null if does not has the next double

`hasNextDouble()`

Результат `bool`

`skip(`*$pattern*`)`

Параметры

- `$pattern` – php\util\Regex

Результат `bool` `true` on success, `false` on fail

`useDelimiter(`*$delimiter*`)`

Параметры

- `$delimiter` – php\util\Regex

Результат php\util\Scanner

`useLocale(`*$locale*`)`

Параметры

- `$locale` – php\util\Locale

Результат php\util\Scanner

`useRadix(`*$value*`)`

Параметры

- `$value` – `int`

Результат php\util\Scanner

`getIOException()`
Get the last io exception (if does not occur then returns `null`)

Результат php\io\IOException, `null`

`reset()`

`current()`
Uses the result of the last called `next()` method

Результат `string`

`key()`

Результат `int`

`valid()`

Результат `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

Параметры

- `$name` – `string`
- `$creator` – `callable` - returns init value

Результат php\util\SharedValue

static `reset(`*$name*`)`
Removes the value by $name.

Параметры

- `$name` – `php\util\String`

Результат 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

Результат `bool`

`count()`
abstract

Результат `int`

`clear()`
abstract

Remove all elements.

Результат `void`

SharedMap

`php\util\SharedMap`
extends: php\util\SharedCollection

Class SharedMap

Methods

`__construct(`*$array*`)`

Параметры

- `$array` – `array`, `Traversable` - (optional)

`get(`*$key*, *$default = null*`)`

Параметры

- `$key` – `string`
- `$default` – `mixed`

Результат `mixed`

`getOrCreate(`*$key*, *$createCallback*`)`

Параметры

- `$key` – `string`
- `$createCallback` – `callable`

Результат `mixed`

`has(`*$key*`)`

Параметры

- `$key` – `string`

Результат `bool`

`count()`

Результат `int`

`set(`*$key*, *$value*, *$override = true*`)`

Параметры

- `$key` – `string`
- `$value` – `mixed`
- `$override` – `bool`

Результат `mixed` previous value

`remove(`*$key*`)`

Параметры

- `$key` – `string`

Результат `mixed`

`clear()`

`isEmpty()`

Результат `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

Параметры

- `$callback` – `callable` - (SharedValue $this)

Результат `mixed` result of execution of $callback

SharedQueue

`php\util\SharedQueue`
extends: php\util\SharedCollection

Class SharedQueue

Methods

`__construct(`*$array*`)`

Параметры

- `$array` – `array`, `Traversable` - (optional)

`isEmpty()`

Результат `bool`

`count()`

Результат `int`

`clear()`
Remove all elements.

Результат `void`

`add(`*$value*`)`

Параметры

- `$value` – `mixed`

Результат `bool`

`peek()`

Результат `mixed`

`poll()`
Retrieves and removes the head of this queue.

Результат `mixed`

SharedStack

`php\util\SharedStack`
extends: php\util\SharedCollection

Class SharedStack

Methods

`__construct(`*$array*`)`

Параметры

- `$array` – `array`, `Traversable` - (optional)

`push(`*$value*`)`

Параметры

- `$value` – `mixed`

Результат `mixed` peek value

`pop()`

Результат `mixed` peek value

`peek()`

Результат `mixed`

`count()`

Результат `int`

`clear()`

`isEmpty()`

Результат `bool`

SharedValue

`php\util\SharedValue`
extends: php\util\SharedMemory

Class SharedValue

Methods

`__construct(`*$value*`)`

Параметры

- `$value` – `mixed` - (optional)

`get()`

Результат `mixed`

`set(`*$value*, *$override = true*`)`

Параметры

- `$value` – `mixed`
- `$override` – `bool`

Результат `mixed`

`remove()`

Результат `mixed`

`isEmpty()`

Результат `bool`

`getAndSet(`*$updateCallback*`)`

Параметры

- `$updateCallback` – `callable` - ($oldValue) returns a new value

Результат `mixed`

`setAndGet(`*$updateCallback*`)`

Параметры

- `$updateCallback` – `callable` - ($oldValue) returns a new value

Результат `mixed`

1.4.17 webserver

WebRequest

`php\webserver\WebRequest`
Class WebRequest

Properties

property `method`
`string`

read-only

property `scheme`
`string`

read-only

property `pathInfo`
`string`

read-only

property `servletPath`
`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

Параметры

- `$parent` – php\webserver\WebRequest

`getBody()`

Результат `string`

`getBodyStream()`

Результат php\io\Stream

`getMethod()`
protected

Результат `string`

`getPathInfo()`
protected

Результат `string`

`getAuthType()`
protected

Результат `string`

`getQueryString()`
protected

Результат `string`

`getUrl()`
protected

Результат `string`

`getScheme()`
protected

Результат `string`

`getPort()`
protected

Результат `int`

`getIp()`
protected

Результат `string`

static `current`

Результат 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

Параметры

- $parent – php\webserver\WebResponse

setHeader(*$name*, *$value*)

Параметры

- $name – string
- $value – string

getHeader(*$name*)

Параметры

- $name – string

Результат string

getHeaders(*$name*)

Параметры

- $name – string

Результат string[]

getHeaderNames()

Результат string[]

addHeader(*$name*, *$value*)

Параметры

- $name – string
- $value – string

redirect(*$location*, *$httpStatus = 301*)

Параметры

- $location – string
- $httpStatus – int

encodeRedirectURL(*$url*)

Параметры

- $url – string

writeToBody(*$content*)

Параметры

- `$content` – `string`

`setContentLength(`*$length*`)`

Параметры

- `$length` – `int`

`addCookie(`*$cookie*`)`

Параметры

- `$cookie` – `array` - [name, value, maxAge, path, domain, httpOnly, secure, comment]

`getStatus()`
protected

Результат `int`

`setStatus(`*$status*`)`
protected

Параметры

- `$status` – `int`

`getContentType()`
protected

Результат `string`

`setContentType(`*$contentType*`)`
protected

Параметры

- `$contentType` – `string`

`getCharacterEncoding()`
protected

Результат `string`

`setCharacterEncoding(`*$characterEncoding*`)`
protected

Параметры

- `$characterEncoding` – `string`

`getBufferSize()`
protected

Результат `int`

`setBufferSize(`*$bufferSize*`)`
protected

Параметры

- `$bufferSize` – `int`

static `current`

Результат 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*`)`

Параметры

- `$onRequest` – `callable`

`run()`

`addStaticHandler(`*$handler*`)`

Параметры

- `$handler` – `array` - [path, location, cache, cachePeriod, gzip]

Результат php\webserver\WebServer

`getId()`
protected

Результат `string`

`getPort()`
protected

Результат `int`

`setPort(`*$port*`)`
protected

Параметры

- `$port` – `int`

`isIsolated()`
protected

Результат boolean

setIsolated(*$isolated*)
protected

Параметры

- $isolated – boolean

isImportAutoloaders()
protected

Результат boolean

setImportAutoloaders(*$importAutoloaders*)
protected

Параметры

- $importAutoloaders – boolean

isHotReload()
protected

Результат boolean

setHotReload(*$hotReload*)
protected

Параметры

- $hotReload – boolean

__clone()
private

static current

Результат php\webserver\WebServer

1.4.18 xml

DomDocument

php\xml\DomDocument
abstract class

extends: php\xml\DomElement

Methods

---

getDocumentElement()

Результат php\xml\DomDocument

getElementById(*$id*)

Параметры

- $id – string

Результат php\xml\DomElement

getInputEncoding()

Результат string

getXmlEncoding()

Результат string

getXmlVersion()

Результат string

getXmlStandalone()

Результат bool

setXmlStandalone(*$value*)

Параметры

- $value – bool

getStrictErrorChecking()

Результат bool

setStrictErrorChecking(*$value*)

Параметры

- $value – bool

getDocumentURI()

Результат string

setDocumentURI(*$value*)

Параметры

- $value – string

createElement(*$tagName*, *$model*)

Параметры

- $tagName – string
- $model – Traversable, array - (optional)

Результат php\xml\DomElement

createElementNS(*$namespaceURI*, *$qualifiedName*)

Параметры

- $namespaceURI – string
- $qualifiedName – string

Результат php\xml\DomElement

createProcessingInstruction(*$name*, *$value*)

Параметры

- $name – string
- $value – string

Результат php\xml\DomNode

importNode(*$importedNode*, *$deep*)

Параметры

- `$importedNode` – php\xml\DomNode
- `$deep` – `bool`

Результат php\xml\DomNode

`adoptNode(`*$source*`)`

Параметры

- `$source` – php\xml\DomNode

Результат php\xml\DomNode

`renameNode(`*$node*, *$namespaceURI*, *$qualifiedName*`)`

Параметры

- `$node` – php\xml\DomNode
- `$namespaceURI` – `string`
- `$qualifiedName` – `string`

`normalizeDocument()`

DomElement

`php\xml\DomElement`

abstract class

extends: php\xml\DomNode

Children

- **abstract class** php\xml\DomDocument

Methods

`__get(`*$name*`)`

Параметры

- `$name` – `string`

Результат `string` Value of attribute by $name

`__set(`*$name*, *$value*`)`
Set attribute value

Параметры

- `$name` – `string`
- `$value` – `string`

`__unset(`*$name*`)`
Remove attribute by name

Параметры

- `$name` – `string`

`__isset(`*$name*`)`
Check attribute exists by name

Параметры

- `$name` –

Результат `bool`

`getTagName()`

Результат `string`

`getAttribute(`*$name*`)`

Параметры

- `$name` – `string`

Результат `string`

`hasAttribute(`*$name*`)`

Параметры

- `$name` – `string`

Результат `bool`

`hasAttributeNS(`*$namespaceURI*, *$localName*`)`

Параметры

- `$namespaceURI` – `string`
- `$localName` – `string`

Результат `bool`

`setAttribute(`*$name*, *$value*`)`

Параметры

- `$name` – `string`
- `$value` – `string`

`setAttributes(`*$attributes*`)`

Параметры

- `$attributes` – `array`, `Traversable`

`getAttributes()`

Результат `array`

`removeAttribute(`*$name*`)`

Параметры

- `$name` – `string`

`getElementsByTagName(`*$name*`)`

Параметры

- `$name` – `string`

Результат php\xml\DomNodeList

`getElementsByTagNameNS(`*$namespaceURI*, *$localName*`)`

Параметры

- `$namespaceURI` – `string`
- `$localName` – `string`

Результат php\xml\DomNodeList

`getAttributeNS(`*$namespaceURI*, *$localName*`)`

Параметры

- `$namespaceURI` – `string`
- `$localName` – `string`

Результат `string`

`setAttributeNS(`*$namespaceURI*, *$qualifiedName*, *$value*`)`

Параметры

- `$namespaceURI` – `string`
- `$qualifiedName` – `string`
- `$value` – `string`

`removeAttributeNS(`*$namespaceURI*, *$localName*`)`

Параметры

- `$namespaceURI` – `string`
- `$localName` – `string`

`setIdAttribute(`*$name*, *$isId*`)`

Параметры

- `$name` – `string`
- `$isId` – `bool`

`setIdAttributeNS(`*$namespaceURI*, *$localName*, *$isId*`)`

Параметры

- `$namespaceURI` – `string`
- `$localName` – `string`
- `$isId` – `string`

DomNode

`php\xml\DomNode`
abstract class

Children

- **abstract class** php\xml\DomElement

Methods

`get(`*$xpathExpression*`)`

Параметры

- `$xpathExpression` – `string`

Результат `string`

`find(`*$xpathExpression*`)`

Параметры

- `$xpathExpression` – `string`

Результат php\xml\DomNode

`findAll(`*$xpathExpression*`)`

Параметры

- `$xpathExpression` – `string`

Результат php\xml\DomNodeList

`getBaseURI()`

Результат `string`

`getNamespaceURI()`

Результат `string`

`getLocalName()`

Результат `string`

`getNodeType()`

Результат `int`

`getNodeName()`

Результат `string`

`getNodeValue()`

Результат `string`

`getPrefix()`

Результат `string`

`getTextContent()`

Результат `string`

`getFirstChild()`

Результат php\xml\DomNode

`getLastChild()`

Результат php\xml\DomNode

`getNextSibling()`

Результат php\xml\DomNode

getPreviousSibling()

Результат php\xml\DomNode

getParentNode()

Результат php\xml\DomNode

getOwnerDocument()

Результат php\xml\DomDocument

hasAttributes()

Результат bool

hasChildNodes()

Результат bool

isDefaultNamespace(*$namespace*)

Параметры

- $namespace – string

isEqualNode(*$node*)

Параметры

- $node – php\xml\DomNode

Результат bool

isSameNode(*$node*)

Параметры

- $node – php\xml\DomNode

Результат bool

isSupported(*$feature*, *$version*)

Параметры

- $feature – string
- $version – string

Результат bool

lookupNamespaceURI(*$prefix*)

Параметры

- $prefix – string

Результат string

lookupPrefix(*$namespaceURI*)

Параметры

- $namespaceURI – string

Результат string

normalize()

setTextContent(*$content*)

Параметры

- `$content` – `string`

`setPrefix(`*$prefix*`)`

Параметры

- `$prefix` – `string`

`cloneNode(`*$deep*`)`

Параметры

- `$deep` – `bool`

Результат php\xml\DomNode

`appendChild(`*$node*`)`

Параметры

- `$node` – php\xml\DomNode

Результат `php\xml\$this`

`removeChild(`*$node*`)`

Параметры

- `$node` – php\xml\DomNode

Результат `php\xml\$this`

`replaceChild(`*$newNode*, *$oldNode*`)`

Параметры

- `$newNode` – php\xml\DomNode
- `$oldNode` – php\xml\DomNode

Результат `php\xml\$this`

`insertBefore(`*$newNode*, *$refNode*`)`

Параметры

- `$newNode` – php\xml\DomNode
- `$refNode` – php\xml\DomNode

Результат `php\xml\$this`

`toModel()`

Результат `array`

DomNodeList

`php\xml\DomNodeList`
abstract class

implements: `ArrayAccess`, `Iterator`, `Countable`

XmlProcessor

`php\xml\XmlProcessor`
extends: php\format\Processor

Class XmlProcessor

Methods

`format(`*$value*`)`
throws php\format\ProcessorException

Параметры

- `$value` – php\xml\DomDocument

Результат `string` xml

`formatTo(`*$value*, *$output*`)`
throws php\format\ProcessorException

Параметры

- `$value` – php\xml\DomDocument
- `$output` – php\io\Stream

`parse(`*$string*`)`
throws php\format\ProcessorException

Параметры

- `$string` – php\io\Stream, `string` - stream of string of xml

Результат php\xml\DomDocument

`createDocument()`

Результат 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.

Примечание: Supports: JDK 1.6+ (OpenJDK, Oracle), Linux, Windows, MacOS, etc.

Symbols

B

C

D

E

F

G

H

I

L

M

N

O

P

W

Y