

# SofortLib

1.5.5

Wed May 8 2013 06:03:26



# Chapter 1

## Main Page

Base class for SOFORT XML-API This class implements basic http authentication and a xml-parser for parsing response messages

Requires libcurl and openssl

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

**Date:**

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

**Version**

[SofortLib](#) 1.5.4

**Id:**

sofortLib.php 5773 2012-11-23 16:15:47Z dehn

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))



## Chapter 2

# Namespace Index

### 2.1 Package List

Here are the packages with brief descriptions (if available):

[sofortLib](#) . . . . . 11



## Chapter 3

# Class Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ArrayToXml . . . . .	13
ArrayToXmlException . . . . .	14
PnagArticle . . . . .	15
PnagCustomer . . . . .	18
PnagInvoice . . . . .	19
SofortElement . . . . .	39
SofortTag . . . . .	112
SofortHtmlTag . . . . .	39
SofortText . . . . .	113
SofortLib . . . . .	40
SofortLib_CancelSr . . . . .	47
SofortLib_ClassicNotification . . . . .	49
SofortLib_ConfirmSr . . . . .	53
SofortLib_Debit . . . . .	57
SofortLib_EditSr . . . . .	62
SofortLib_iDeal_Banks . . . . .	65
SofortLib_Multipay . . . . .	69
SofortLib_Notification . . . . .	84
SofortLib_Refund . . . . .	86
SofortLib_SofortueberweisungClassic . . . . .	89
SofortLib_iDealClassic . . . . .	66
SofortLib_TransactionData . . . . .	94
XmlToArray . . . . .	114
XmlToArrayException . . . . .	116
XmlToArrayNode . . . . .	116





# Chapter 4

## Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">ArrayToXml</a>	13
<a href="#">ArrayToXmlException</a>	14
<a href="#">PnagArticle</a>	15
<a href="#">PnagCustomer</a>	18
<a href="#">PnagInvoice</a>	19
<a href="#">SofortElement</a>	39
<a href="#">SofortHtmlTag</a>	39
<a href="#">SofortLib</a>	40
<a href="#">SofortLib_CancelSr</a>	47
<a href="#">SofortLib_ClassicNotification</a>	49
<a href="#">SofortLib_ConfirmSr</a>	53
<a href="#">SofortLib_Debit</a>	57
<a href="#">SofortLib_EditSr</a>	62
<a href="#">SofortLib_iDeal_Banks</a>	65
<a href="#">SofortLib_iDealClassic</a>	66
<a href="#">SofortLib_Multipay</a>	69
<a href="#">SofortLib_Notification</a>	84
<a href="#">SofortLib_Refund</a>	86
<a href="#">SofortLib_SofortueberweisungClassic</a>	89
<a href="#">SofortLib_TransactionData</a>	94
<a href="#">SofortTag</a>	112
<a href="#">SofortText</a>	113
<a href="#">XmlToArray</a>	114
<a href="#">XmlToArrayException</a>	116
<a href="#">XmlToArrayNode</a>	116



## Chapter 5

# Namespace Documentation

### 5.1 Package sofortLib

#### 5.1.1 Detailed Description

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

**Date:**

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

**Version**

**Id:**

class.invoice.inc.php 5773 2012-11-23 16:15:47Z dehn

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))



# Chapter 6

## Class Documentation

### 6.1 ArrayToXml Class Reference

#### Public Member Functions

- [\\_\\_construct](#) (array \$input, \$maxDepth=10, \$trim=true)
- [toXml](#) (\$version= '1.0', \$encoding= 'UTF-8')

#### Static Public Member Functions

- static [render](#) (array \$input, array \$options=array())

#### 6.1.1 Detailed Description

##### Author

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SOFORT AG Released under the GNU General Public License (Version 2) [<http://www.gnu.org/licenses/gpl-2.0.html>] Array To XML conversion

#### 6.1.2 Constructor & Destructor Documentation

##### 6.1.2.1 `__construct ( array $ input, $ maxDepth = 10, $ trim = true )`

Loads array into XML representation.

##### Parameters

*array* \$input

*int* \$maxDepth

**Exceptions**[\*ArrayToXmlException\*](#)**6.1.3 Member Function Documentation****6.1.3.1 static render ( array \$ input, array \$ options = *array()* )  
[static]**

static entry point. Options are:

- version: (default 1.0) version string to put in xml prolog
- encoding: (default UTF-8) use the specified encoding
- trim: (default true) Trim values
- depth: (default 10) Maximum depth to parse the given array, throws exception when exceeded

**Parameters**

*array* \$input the input array

*array* \$options set additional options to pass to [XmlToArray](#)

**Exceptions**[\*ArrayToXmlException\*](#)**6.1.3.2 toXml ( \$ version = '1.0', \$ encoding = 'UTF-8' )**

Returns parsed array as XML structure Pass both params as null to exclude prolog  
<?xml version="version" encoding="encoding" ?>

**Parameters**

*string* \$version

*string* \$encoding

The documentation for this class was generated from the following file:

- array\_to\_xml.php

**6.2 ArrayToXmlException Class Reference****6.2.1 Detailed Description****Author**

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SOFORT AG Released under the GNU General Public License (Version 2)

[\[http://www.gnu.org/licenses/gpl-2.0.html\]](http://www.gnu.org/licenses/gpl-2.0.html) Array To XML Exception

The documentation for this class was generated from the following file:

- array\_to\_xml\_exception.php

## 6.3 PnagArticle Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$itemId, \$productNumber, \$productType, \$title, \$description, \$quantity, \$unitPrice, \$tax)
- [getItemId](#) ()
- [getQuantity](#) ()
- [setQuantity](#) (\$quantity)
- [getUnitPrice](#) ()
- [setUnitPrice](#) (\$unitPrice)
- [getTitle](#) ()
- [getTax](#) ()
- [setTax](#) (\$value)
- [setProductNumber](#) (\$productNumber)
- [getProductNumber](#) ()
- [setDescription](#) (\$description)
- [getDescription](#) ()

### 6.3.1 Detailed Description

Data object that encapsulates article's data

**Date:**

2012-09-05 14:27:56 +0200 (Mi, 05 Sep 2012)

\$ID\$

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 [\\_\\_construct](#) ( *\$ itemId*, *\$ productNumber*, *\$ productType*, *\$ title*, *\$ description*, *\$ quantity*, *\$ unitPrice*, *\$ tax* )

Constructor for [PnagArticle](#)

**Parameters**

*\$itemId* int

*\$productNumber* string

*\$productType* string

*\$title* string

*\$description* string

*\$quantity* int

*\$unitPrice* float

*\$tax* float

### 6.3.3 Member Function Documentation

#### 6.3.3.1 getDescription ( )

Getter for description

#### 6.3.3.2 getItemId ( )

Getter for item id

#### 6.3.3.3 getProductNumber ( )

Getter for product number

#### 6.3.3.4 getQuantity ( )

Getter for quantity

#### 6.3.3.5 getTax ( )

Getter for tax value

#### 6.3.3.6 getTitle ( )

Getter for title

#### 6.3.3.7 getUnitPrice ( )

Getter for unit price



**6.3.3.8 setDescription ( \$ *description* )**

Setter for description

**Parameters**

*string* \$description

**6.3.3.9 setProductNumber ( \$ *productNumber* )**

Setter for product number

**Parameters**

*string* \$productNumber

**6.3.3.10 setQuantity ( \$ *quantity* )**

Setter for quantity

**Parameters**

*int* \$quantity

**6.3.3.11 setTax ( \$ *value* )**

Setter for tax value

**Parameters**

*float* \$value

**6.3.3.12 setUnitPrice ( \$ *unitPrice* )**

Setter for unit price

**Parameters**

*float* \$unitPrice

The documentation for this class was generated from the following file:

- pnag\_article.php

## 6.4 PnagCustomer Class Reference

### Public Member Functions

- [PnagCustomer](#) (\$name= "", \$lastname= "", \$firstname= "", \$company= "", \$csID= "", \$vatId= "", \$shopId= "", \$Id= "", \$cIP= "", \$streetAddress= "", \$suburb= "", \$city= "", \$postcode= "", \$state= "", \$country= "", \$formatId= "", \$telephone= "", \$emailAddress= "")

#### 6.4.1 Detailed Description

Data object that encapsulates user's data

##### Date:

2012-09-05 14:27:56 +0200 (Mi, 05 Sep 2012)

\$ID\$

#### 6.4.2 Member Function Documentation

**6.4.2.1 PnagCustomer** ( *\$ name = "", \$ lastname = "", \$ firstname = "", \$ company = "", \$ csID = "", \$ vatId = "", \$ shopId = "", \$ Id = "", \$ cIP = "", \$ streetAddress = "", \$ suburb = "", \$ city = "", \$ postcode = "", \$ state = "", \$ country = "", \$ formatId = "", \$ telephone = "", \$ emailAddress = ""* )

Set the customer's credentials

##### Parameters

*\$name* string

*\$lastname* string

*\$firstname* string

*\$company* string

*\$csID* string customer id in shop

*\$vatId* string - customer's VAT ID

*\$shopId* - shop's ID

*\$Id*

*\$cIP*

*\$streetAddress* string

*\$suburb* string

*\$city* string

*\$postcode* string

*\$state* string

*\$country* string  
*\$formatId* string  
*\$telephone* string  
*\$emailAddress* string

The documentation for this class was generated from the following file:

- pnag\_customer.php

## 6.5 PnagInvoice Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey, \$transactionId= '')
- [getConstantById](#) (\$id)
- [getConstantByName](#) (\$name)
- [setBitmask](#) (\$status, \$statusReason, \$invoiceStatus)
- [setState](#) (\$state, \$callback= '')
- [getState](#) ()
- [setTransactionId](#) (\$transactionId)
- [setSofortLibMultipay](#) (\$SofortLib\_Multipay)
- [setSofortLibTransactionData](#) (\$SofortLib\_TransactionData)
- [setSofortLibEditSr](#) (\$SofortLib\_EditSr)
- [setSofortLibCancelSr](#) (\$SofortLib\_CancelSr)
- [refreshTransactionData](#) ()
- [cancelInvoice](#) (\$transactionId= '', \$creditNoteNumber= '')
- [confirmInvoice](#) (\$transactionId= '', \$invoiceNumber= '', \$customerNumber= '', \$orderNumber= '')
- [updateInvoice](#) (\$transactionId, \$items, \$comment, \$invoiceNumber= '', \$customerNumber= '', \$orderNumber= '')
- [updateOrderNumber](#) (\$transactionId, \$orderNumber)
- [addItemToInvoice](#) (\$itemId, \$productNumber, \$title, \$unitPrice, \$productType=0, \$description= '', \$quantity=1, \$tax=19)
- [removeItemfromInvoice](#) (\$itemId)
- [updateInvoiceItem](#) (\$itemId, \$quantity, \$unitPrice)
- [getItemAmount](#) (\$itemId)
- [addShippingAddress](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE', \$nameAdditive= '', \$streetAdditive= '', \$companyName= '')
- [addShippingAddresss](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE', \$nameAdditive= '', \$streetAdditive= '')
- [addInvoiceAddress](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE', \$nameAdditive= '', \$streetAdditive= '', \$companyName= '')
- [setOrderId](#) (\$arg)

- [setCustomerId](#) (\$arg)
- [setPhoneNumberCustomer](#) (\$arg)
- [setEmailCustomer](#) (\$arg)
- [addUserVariable](#) (\$arg)
- [setNotificationUrl](#) (\$arg)
- [setAbortUrl](#) (\$arg)
- [setSuccessUrl](#) (\$arg)
- [setTimeoutUrl](#) (\$arg)
- [setTimeout](#) (\$arg)
- [setReason](#) (\$reason1, \$reason2= '')
- [setAmount](#) (\$arg, \$currency= 'EUR')
- [getAmount](#) ()
- [setAmountRefunded](#) (\$arg)
- [getAmountRefunded](#) ()
- [setSofortrechnung](#) ()
- [setDebitorVatNumber](#) (\$vatNumber)
- [getPaymentUrl](#) ()
- [getInvoiceNumber](#) ()
- [getCustomerNumber](#) ()
- [getOrderNumber](#) ()
- [getInvoiceTye](#) ()
- [getTransactionId](#) ()
- [checkout](#) ()
- [getTransactionInfo](#) ()
- [getInvoice](#) ()
- [handleDownload](#) (\$method= 'socket')
- [getInvoiceDownloadMethod](#) ()
- [getHttpStatusCode](#) ()
- [setInvoiceUrl](#) (\$invoiceUrl)
- [getInvoiceUrl](#) ()
- [setStatus](#) (\$status)
- [setStatusReason](#) (\$statusReason)
- [setStatusOfInvoice](#) (\$invoiceStatus= '')
- [setLanguageCode](#) (\$languageCode= '')
- [setTransaction](#) (\$transaction)
- [setTime](#) (\$time)
- [setVersion](#) (\$arg)
- [setPaymentMethod](#) (\$paymentMethod)
- [setInvoiceObjection](#) (\$invoiceObjection)
- [setInvoiceStatus](#) (\$invoiceStatus)
- [getInvoiceObjection](#) ()
- [getStatusOfInvoice](#) ()
- [getInvoiceStatus](#) ()
- [getPaymentMethod](#) ()
- [getStatusReason](#) ()
- [getStatus](#) ()

- [getLanguageCode \(\)](#)
- [getItems \(\)](#)
- [setItems \(\\$items\)](#)
- [getTransactionData \(\)](#)
- [isError \(\)](#)
- [isWarning \(\)](#)
- [getError \(\)](#)
- [getErrors \(\)](#)
- [getErrorCodes \(\\$detailLevel=0\)](#)
- [getWarnings \(\)](#)
- [enableLog \(\)](#)
- [disableLog \(\)](#)
- [log \(\\$message\)](#)
- [logError \(\\$message\)](#)
- [logWarning \(\\$message\)](#)
- [\\_\\_toString \(\)](#)

### 6.5.1 Detailed Description

Abstraction of an invoice Helper class to ease usage of "Rechnung by sofort" Encapsulates Multipay, TransactionData and ConfirmSr to handle everything there is about "Rechnung by sofort"

See also

[SofortLib\\_Multipay](#)  
[SofortLib\\_TransactionData](#)  
[SofortLib\\_ConfirmSr](#)

### 6.5.2 Constructor & Destructor Documentation

#### 6.5.2.1 `__construct ( $ configKey, $ transactionId = " ")`

Constructor for [PnagInvoice](#)

Parameters

*string* \$apiKey  
*string* \$transactionId  
*string* \$apiUrl

### 6.5.3 Member Function Documentation

#### 6.5.3.1 `__toString ( )`

Override toString

**6.5.3.2** `addInvoiceAddress ( $ firstname, $ lastname, $ street, $ streetNumber, $ zipcode, $ city, $ salutation, $ country = 'DE', $ nameAdditive = "", $ streetAdditive = "", $ companyName = "" )`

Wrapper for [SofortLib\\_Multipay.setSofortrechnungInvoiceAddress](#)

See also

[SofortLib\\_Multipay](#)

**Parameters**

*\$firstname*

*\$lastname*

*\$street*

*\$streetNumber*

*\$zipcode*

*\$city*

*\$salutation*

*\$country* (optional, default: DE)

*\$nameAdditive* (optional)

*\$streetAdditive* (optional)

*\$companyName* (optional)

**6.5.3.3** `addItemToInvoice ( $ itemId, $ productNumber, $ title, $ unitPrice, $ productType = 0, $ description = "", $ quantity = 1, $ tax = 19 )`

Wrapper for [SofortLib\\_Multipay.addSofortrechnungItem](#)

See also

[SofortLib\\_Multipay](#)

**Parameters**

*\$itemId*

*\$productNumber*

*\$title*

*\$unit\_price* - float precision 2

See also

[SofortLib\\_Multipay](#) api

**Parameters**

*\$productType*

*\$description*

*\$quantity* - int

*\$tax*

**6.5.3.4** `addShippingAddress ( $ firstname, $ lastname, $ street, $ streetNumber, $ zipcode, $ city, $ salutation, $ country = 'DE', $ nameAdditive = "", $ streetAdditive = "", $ companyName = "" )`

Wrapper for [SofortLib\\_Multipay.setSofortrechnungShippingAddress](#)

See also

[SofortLib\\_Multipay](#)

#### Parameters

*\$firstname*

*\$lastname*

*\$street*

*\$streetNumber*

*\$zipcode*

*\$city*

*\$salutation*

*\$country* (optional, default: DE)

*\$nameAdditive* (optional)

*\$streetAdditive* (optional)

*\$companyName* (optional)

**6.5.3.5** `addShippingAddresss ( $ firstname, $ lastname, $ street, $ streetNumber, $ zipcode, $ city, $ salutation, $ country = 'DE', $ nameAdditive = "", $ streetAdditive = "" )`

Wrapper for [SofortLib\\_Multipay.setSofortrechnungShippingAddress](#)

See also

[SofortLib\\_Multipay](#)

#### Parameters

*\$firstname*

*\$lastname*

*\$street*

*\$streetNumber*

*\$zipcode*

*\$city*

*\$salutation*

*\$country* (optional, default: DE)

*\$nameAdditive* (optional)

*\$streetAdditive* (optional)

#### 6.5.3.6 addUserVariable ( \$ arg )

Wrapper function for [SofortLib\\_Multipay.addUserVariable](#)

##### Parameters

*\$arg*

#### 6.5.3.7 cancelInvoice ( \$ transactionId = "", \$ creditNoteNumber = "" )

Wrapper function for cancelling this invoice via [SofortLib\\_Multipay](#) ([SofortLib](#))

##### Returns

Ambigious boolean/Array

#### 6.5.3.8 checkout ( )

Validate your parameters against API

##### Returns

array - any validationerrors and -warnings send the order to pnag (-> buy your products)  
empty array if ok ELSE array with errors and/or warnings

#### 6.5.3.9 confirmInvoice ( \$ transactionId = "", \$ invoiceNumber = "", \$ customerNumber = "", \$ orderNumber = "" )

Wrapper function for confirming this invoice via [SofortLib\\_Multipay](#) ([SofortLib](#))

##### Parameters

*\$transactionId* - optional parameter for confirming a transaction on the fly  
*\$invoiceNumber* - optional parameter for own invoice number  
*\$customerNumber* - optional parameter for own customer number  
*\$orderNumber* - optional parameter for own order number

##### Returns

Ambigious boolean/Array

#### 6.5.3.10 disableLog ( )

Disable logging for all encapsed [SofortLib](#) components

##### Returns

boolean



**6.5.3.11 enableLog ( )**

Enabling logging for all encapsed [SofortLib](#) components

**Returns**

boolean

**6.5.3.12 getAmount ( )**

current total amount of the given order-articles

**Returns**

float - sum (price, total) of all articles

**6.5.3.13 getAmountRefunded ( )**

Getter for amount refunded

**6.5.3.14 getConstantById ( \$ id )**

Getter for a class constant

**Parameters**

*int* \$id

**Returns**

string

**6.5.3.15 getConstantByName ( \$ name )**

Getter for a class constant

**Parameters**

*string* \$name

**Returns**

int

**6.5.3.16 getCustomerNumber ( )**

Getter for customer's number

**6.5.3.17** `getError ( )`

returns one error (as String!)

**6.5.3.18** `getErrorCodes ( $ detailLevel = 0 )`

Ouputs errors in a more convenient array to let users easily iterate

**Parameters**

*int* \$detailLevel

**6.5.3.19** `getErrors ( )`

collect all errors and returns them

**Returns**

array - all errors

**6.5.3.20** `getHttpResponseCode ( )`

Getter for HTTP Response code

**6.5.3.21** `getInvoice ( )`

Output the resulting invoice as pdf, if possible Function uses file\_get\_contents, if allow\_url\_fopen is allowed in php.ini (might be disabled on shared hosting) As a fall-back, downloading via cURL, when module cURL is available If neither file\_get\_contents nor cURL is available for downloading, a connection via socket is used to download.

**Returns**

boolean

**6.5.3.22** `getInvoiceDownloadMethod ( )`

Getter for invoice's download method

**6.5.3.23** `getInvoiceNumber ( )`

Getter for invoice's number

**6.5.3.24 getInvoiceObjection ( )**

Returns the reason for objecting this invoice

**Returns**

string

**6.5.3.25 getInvoiceStatus ( )**

Uses the statusMask to "calculate" the current invoice's payment status

**See also**

Invoice.\_calcInvoiceStatusCode

**Returns**

int

**6.5.3.26 getInvoiceTye ( )**

Wrapper function for SofortLib\_Multipay.geInvoiceType

**Returns**

(OR or LS)

**6.5.3.27 getInvoiceUrl ( )**

Getter for retrieving the invoice's url

**Returns**

url string

**6.5.3.28 getItemAmount ( \$ *itemId* )**

Getter for an invoice's amount

**Parameters**

*string* \$itemId

**6.5.3.29** `getItems ( )`

Getter for items

**Returns**

array

**6.5.3.30** `getLanguageCode ( )`

Getter for language code

**Returns**

string

**6.5.3.31** `getOrderNumber ( )`

Getter for order's number

**6.5.3.32** `getPaymentMethod ( )`

Getter for payment\_method

**Returns**

string

**6.5.3.33** `getPaymentUrl ( )`

Wrapper function for [SofortLib\\_Multipay.getPaymentUrl](#)

**Returns**

url string

**6.5.3.34** `getState ( )`

Getter for the current state

**Returns**

int \$this->state

**6.5.3.35 getStatus ( )**

Getter for status

**Returns**

string

**6.5.3.36 getStatusOfInvoice ( )**

Instead of calculated status, this method returns the invoice's status (string)

**Returns**

string

**6.5.3.37 getStatusReason ( )**

Getter for status\_reason

**Returns**

string

**6.5.3.38 getTransactionData ( )**

return TransactionData, the invoice is working with NOTICE: if status changed (removeArticle, InvoiceConfirmed etc.) it returns always the FRESH TransactionData from pnag-server

**Returns**

object

**See also**

`$this->refreshTransactionData\(\);`

**6.5.3.39 getTransactionId ( )**

Wrapper function for [SofortLib\\_Multipay.getPaymentUrl](#)

**Returns**

url string

**6.5.3.40** `getTransactionInfo ( )`

Getter for information about transaction

**6.5.3.41** `getWarnings ( )`

collects all warnings and returns them

**Returns**

array

**6.5.3.42** `handleDownload ( $ method = ' socket ' )`

Handle download of invoice

**Parameters**

*string* \$method

**6.5.3.43** `isError ( )`

Check, if errors occurred

**Returns**

boolean

**6.5.3.44** `isWarning ( )`

Check, if warnings occurred

**Returns**

boolean

**6.5.3.45** `log ( $ message )`

Log the given String into log.txt Notice: logging must be enabled -> use [enableLog\(\)](#);

**Parameters**

*string* \$msg - Message to log

**Returns**

bool - true=logged ELSE false=logging failed

**6.5.3.46 logError ( \$ message )**

Log the given String into error\_log.txt Notice: logging must be enabled -> use [enableLog\(\)](#);

**Parameters**

*string* \$msg - Message to log

**Returns**

bool - true=logged ELSE false=logging failed

**6.5.3.47 logWarning ( \$ message )**

Log the given String into warning\_log.txt

**Parameters**

*string* \$msg - Message to log

**Returns**

bool - true=logged ELSE false=logging failed

**6.5.3.48 refreshTransactionData ( )**

Refreshes the TransactionData with the data directly from the pnag-server

**Returns**

boolean

**6.5.3.49 removeItemfromInvoice ( \$ itemId )**

Remove an item from the invoice

**Parameters**

*\$itemId*

**Returns**

boolean

**6.5.3.50 setAbortUrl ( \$ arg )**

Wrapper function for [SofortLib\\_Multipay.setAbortUrl](#)

**Parameters**

*\$arg*

**6.5.3.51 setAmount ( \$ arg, \$ currency = 'EUR' )**

Wrapper function for [SofortLib\\_Multipay.setAmount](#)

**Parameters**

*\$arg* float

*\$currency* string

**6.5.3.52 setAmountRefunded ( \$ arg )**

Setter for amount refunded

**Parameters**

*amount* \$arg

**6.5.3.53 setBitmask ( \$ status, \$ statusReason, \$ invoiceStatus )**

Set the bitmask to a specific state

**Parameters**

*string* \$status

*string* \$status\_reason

*string* \$invoice\_status

**Returns**

string pending - confirm\_invoice - empty -> 4195329

**6.5.3.54 setCustomerId ( \$ arg )**

Wrapper function for [SofortLib\\_Multipay.setSofortrechnungCustomerId](#)

**Parameters**

*\$arg*



**6.5.3.55 setDebitorVatNumber ( \$ *vatNumber* )**

Wrapper function for [SofortLib\\_Multipay.setDebitorVatNumber](#)

**6.5.3.56 setEmailCustomer ( \$ *arg* )**

Wrapper function for [SofortLib\\_Multipay.setEmailCustomer](#)

**Parameters**

*\$arg*

**6.5.3.57 setInvoiceObjection ( \$ *invoiceObjection* )**

Sets the reason for objecting this invoice

**Parameters**

*string* \$invoiceObjection (40-50 chars max.)

**Returns**

object

**6.5.3.58 setInvoiceStatus ( \$ *invoiceStatus* )**

Sets the invoice status

**Parameters**

*string* \$invoiceStatus

**Returns**

object

**6.5.3.59 setInvoiceUrl ( \$ *invoiceUrl* )**

Setter for invoiceUrl

**Parameters**

*\$invoiceUrl*

**Returns**

object

**6.5.3.60 setItems ( \$ items )**

Setter for invoice items, takes an array of [PnagArticle](#) objects

**Parameters**

*array* \$items

**6.5.3.61 setLanguageCode ( \$ languageCode = " )**

Setter for language\_code

**Parameters**

*string* \$language\_code | fallback en

**Returns**

object

**6.5.3.62 setNotificationUrl ( \$ arg )**

Wrapper function for [SofortLib\\_Multipay.setNotificationUrl](#)

**Parameters**

*\$arg*

**6.5.3.63 setOrderId ( \$ arg )**

Wrapper function for [SofortLib\\_Multipay.setSofortrechnungOrderId](#)

**See also**

[SofortLib\\_Multipay](#)

**Parameters**

*\$arg*

**6.5.3.64 setPaymentMethod ( \$ paymentMethod )**

Setter for payment\_method

**Parameters**

*\$paymentMethod*

**Returns**

object

**6.5.3.65 setPhoneNumberCustomer ( \$ arg )**

Wrapper function for [SofortLib\\_Multipay.setPhoneNumberCustomer](#)

**Parameters**

*\$arg*

**6.5.3.66 setReason ( \$ reason1, \$ reason2 = " )**

Wrapper function for [SofortLib\\_Multipay.setReason](#)

**Parameters**

*\$reason1* string

*\$reason2* string

**6.5.3.67 setSofortLibCancelSr ( \$ SofortLib\_CancelSr )**

Setter for [SofortLib\\_CancelSr](#)

**Parameters**

*object* \$SofortLib\_CancelSr

**6.5.3.68 setSofortLibEditSr ( \$ SofortLib\_EditSr )**

Setter for [SofortLib\\_EditSr](#)

**Parameters**

*object* \$SofortLib\_EditSr

**6.5.3.69 setSofortLibMultipay ( \$ SofortLib\_Multipay )**

Setter for [SofortLib\\_Multipay](#)

**Parameters**

*object* \$SofortLib\_Multipay

**6.5.3.70 setSofortLibTransactionData ( \$ SofortLib\_TransactionData )**

Setter for [SofortLib\\_TransactionData](#)

**Parameters**

*object* \$SofortLib\_TransactionData

**6.5.3.71 setSofortrechnung ( )**

Wrapper function for [SofortLib\\_Multipay.setSofortrechnung](#)

**6.5.3.72 setState ( \$ state, \$ callback = " )**

Set the state An optional callback can be registered

**Parameters**

*int* \$state

*function* \$callback

**6.5.3.73 setStatus ( \$ status )**

Setter for status

**Parameters**

*\$status*

**Returns**

object

**6.5.3.74 setStatusOfInvoice ( \$ invoiceStatus = " )**

Setter for invoice\_status

**Parameters**

*string* \$invoice\_status | may be empty

**Returns**

object

**6.5.3.75 setStatusReason ( \$ statusReason )**

Setter for status\_reason

**Parameters**

*\$status\_reason*

**Returns**

object

**6.5.3.76 setSuccessUrl ( \$ *arg* )**

Wrapper function for [SofortLib\\_Multipay.setSuccessUrl](#)

**Parameters**

*\$arg*

**6.5.3.77 setTime ( \$ *time* )**

Setter for variable time

**Parameters**

*\$time* return object

**6.5.3.78 setTimeout ( \$ *arg* )**

Wrapper function for [SofortLib\\_Multipay.setTimeout](#)

**Parameters**

*\$arg*

**6.5.3.79 setTimeoutUrl ( \$ *arg* )**

Wrapper function for [SofortLib\\_Multipay.setTimeoutUrl](#)

**Parameters**

*\$arg*

**6.5.3.80 setTransaction ( \$ *transaction* )**

Setter for transaction

**Parameters**

*\$transaction*

**Returns**

object

**6.5.3.81    setTransactionId ( \$ transactionId )**

Setter for transactionId

**Parameters**

*\$transactionId*

**6.5.3.82    setVersion ( \$ arg )**

Setter for interface version Wrapper for class Multipay to set version according to shop module and it's interface version e.g. 'pn\_xtc\_5.0.0'

**Parameters**

*\$arg* string

**6.5.3.83    updateInvoice ( \$ transactionId, \$ items, \$ comment, \$ invoiceNumber = "", \$ customerNumber = "", \$ orderNumber = "" )**

Wrapper function for removing an article via [SofortLib\\_Multipay](#) ([SofortLib](#))

**Parameters**

*\$transactionId* string

*\$PnagArticels* array

*\$comment* int

return array

**6.5.3.84    updateInvoiceItem ( \$ itemId, \$ quantity, \$ unitPrice )**

Update an invoice's item

**Parameters**

*string* \$itemId

*int* \$quantity

*float* \$unitPrice

**6.5.3.85    updateOrderNumber ( \$ transactionId, \$ orderNumber )**

Wrapper function for updating order number after order has been placed

**Parameters**

*string* \$transactionId

*string* \$orderNumber

The documentation for this class was generated from the following file:

- class.invoice.inc.php

## 6.6 SofortElement Class Reference

Inherited by [SofortTag](#), and [SofortText](#).

### Public Member Functions

- [render](#) ()

#### 6.6.1 Detailed Description

##### Author

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SOFORT AG Released under the GNU General Public License (Version 2) [<http://www.gnu.org/licenses/gpl-2.0.html>] Implementation of a simple element

#### 6.6.2 Member Function Documentation

##### 6.6.2.1 [render](#) ( ) [abstract]

Render the element

Reimplemented in [SofortTag](#), and [SofortText](#).

The documentation for this class was generated from the following file:

- sofort\_element.php

## 6.7 SofortHtmlTag Class Reference

Inherits [SofortTag](#).

### Public Member Functions

- [\\_\\_construct](#) (\$tagname, array \$attributes=array(), \$children=array())

## Protected Member Functions

- [\\_render](#) (\$output, \$attributes)

### 6.7.1 Detailed Description

#### Author

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SOFORT AG Released under the GNU General Public License (Version 2) [<http://www.gnu.org/licenses/gpl-2.0.html>] Implementation of an HTML Tag

### 6.7.2 Constructor & Destructor Documentation

#### 6.7.2.1 `__construct ( $ tagname, array $ attributes = array (), $ children = array () )`

Constructor for [SofortHtmlTag](#)

#### Parameters

*string* \$tagname  
*array* \$attributes  
*array* \$children

Reimplemented from [SofortTag](#).

### 6.7.3 Member Function Documentation

#### 6.7.3.1 `_render ( $ output, $ attributes )` [protected]

Renders the element (override)

#### See also

[SofortTag.\\_render\(\)](#)

Reimplemented from [SofortTag](#).

The documentation for this class was generated from the following file:

- `sofort_html_tag.php`

## 6.8 SofortLib Class Reference

### Public Member Functions

- `__construct` (\$userId= "", \$apiKey= "", \$apiUrl= "")



- [getWarnings](#) (\$paymentMethod= 'all', \$message= '')
- [getErrors](#) (\$paymentMethod= 'all', \$message= '')
- [handleErrors](#) (\$errors)
- [getError](#) (\$paymentMethod= 'all', \$message= '')
- [isWarning](#) (\$paymentMethod= 'all', \$message= '')
- [isError](#) (\$paymentMethod= 'all', \$message= '')
- [setError](#) (\$message, \$pos= 'global', \$errorCode= '-1', \$field= '')
- [deleteAllWarnings](#) ()
- [deleteAllErrors](#) ()
- [error](#) (\$message, \$fatal=false)
- [fatalError](#) (\$message)
- [enableLog](#) ()
- [disableLog](#) ()
- [setLogEnabled](#) ()
- [setLogDisabled](#) ()
- [setLogger](#) (\$SofortLibLogger)
- [logWarning](#) (\$message)
- [logError](#) (\$message)
- [log](#) (\$message)
- [setLogfilePath](#) (\$path)
- [setApiVersion](#) (\$version)

## Static Public Member Functions

- static [debug](#) (\$var=false, \$showHtml=false)

## Protected Member Functions

- [\\_sendMessage](#) (\$message)
- [\\_handleErrors](#) ()
- [\\_parseXml](#) ()
- [\\_getErrorBlock](#) (\$error)

### 6.8.1 Detailed Description

Include any available helper here

Basic PHP Library for communication with multipay API and related products of so-fort.com

#### Author

payment-network.com

## 6.8.2 Constructor & Destructor Documentation

### 6.8.2.1 `__construct ( $ userId = "", $ apiKey = "", $ apiUrl = "" )`

Constructor for [SofortLib](#)

#### Parameters

*string* \$apiKey

## 6.8.3 Member Function Documentation

### 6.8.3.1 `_getErrorBlock ( $ error ) [protected]`

Getter for error block

#### Parameters

*unknown\_type* \$error

### 6.8.3.2 `_handleErrors ( ) [protected]`

Handle Errors occurred

### 6.8.3.3 `_parseXml ( ) [protected]`

parse the XML received or being sent

### 6.8.3.4 `_sendMessage ( $ message ) [protected]`

internal send-method, will check http-errorcode and return body

#### Parameters

*String* \$message message to post

#### Returns

string error or body

### 6.8.3.5 `static debug ( $ var = false, $ showHtml = false ) [static]`

Static debug method

#### Parameters

*mixed* \$var

*boolean* \$showHtml

### 6.8.3.6 deleteAllErrors ( )

delete all errors

### 6.8.3.7 deleteAllWarnings ( )

delete all warnings

### 6.8.3.8 disableLog ( )

Disable logging in object

See also

[SofortLib setLogDisabled](#)

### 6.8.3.9 enableLog ( )

Enable logging in object

See also

[SofortLib setLogEnabled](#)

### 6.8.3.10 error ( \$ message, \$ fatal = *false* )

Setter for an error

#### Parameters

*string* \$message

*boolean* \$fatal

### 6.8.3.11 fatalError ( \$ message )

error while parsing xml

#### Parameters

*unknown\_type* \$message

**6.8.3.12** `getError ( $ paymentMethod = 'a11', $ message = " )`

returns one error message

**See also**

[getErrors\(\)](#) for more detailed errors

**Parameters**

*array* \$message response array

**Returns**

string errormessage ELSE false

**6.8.3.13** `getErrors ( $ paymentMethod = 'a11', $ message = " )`

Getter for errors

**Parameters**

(*optional*) *array* \$message response array

**Returns**

empty array if no error exist ELSE array with error-codes and error-messages

**6.8.3.14** `getWarnings ( $ paymentMethod = 'a11', $ message = " )`

Getter for warnings

**Returns**

empty array if no warnings exists ELSE array with warning-codes and warning-messages

**6.8.3.15** `handleErrors ( $ errors )`

Alter error array and set error message and error code together as one

**Parameters**

*array* \$errors

**6.8.3.16** `isError ( $ paymentMethod = 'a11', $ message = " )`

checks (response)-array for error

**Parameters**

*array* \$message response array

*string* \$paymentMethod - 'all', 'sr', 'su', 'sv', 'sa', 'ls', 'sl', 'sf' (if unknown then it uses "all")

**Returns**

boolean true if errors found (in given payment-method or in 'global') ELSE false

**6.8.3.17** `isWarning ( $ paymentMethod = 'a11', $ message = " )`

checks (response)-array for warnings

**Parameters**

*array* \$message response array

*string* \$paymentMethod - 'all', 'sr', 'su', 'sv', 'sa', 'ls', 'sl', 'sf' (if unknown then it uses "all")

**Returns**

boolean true if warnings found ELSE false

**6.8.3.18** `log ( $ message )`

log the given string into log.txt use \$this->[enableLog\(\)](#); to enable logging before!

**Parameters**

*string* \$message

**6.8.3.19** `logError ( $ message )`

log the given string into error\_log.txt use \$this->[enableLog\(\)](#); to enable logging before!

**Parameters**

*string* \$message

### 6.8.3.20 logWarning ( \$ message )

log the given string into warning\_log.txt use \$this->[enableLog\(\)](#); to enable logging before!

#### Parameters

*string* \$message

### 6.8.3.21 setApiVersion ( \$ version )

sets the api version which should be used

#### Parameters

*float* \$version

### 6.8.3.22 setError ( \$ message, \$ pos = 'global', \$ errorCode = '-1', \$ field = " )

set Errors later use [getError\(\)](#), [getErrors\(\)](#) or [isError\(\)](#) to retrieve them

#### Parameters

*string* \$message - Detailinformationen about the error

*string* \$pos - Position in the errors-array, must be one of: 'global', 'sr', 'su', 'sv', 'sa', 'ls', 'sl', 'sf'

*string* \$errorCode - a number or string to specify the errors in the module

*string* \$field - if \$errorCode deals with a field

### 6.8.3.23 setLogDisabled ( )

Set logging disabled [disableLog\(\)](#);

### 6.8.3.24 setLogEnabled ( )

Set logging enable [enableLog\(\)](#);

### 6.8.3.25 setLogfilePath ( \$ path )

Set the path of the logfile

#### Parameters

*string* \$path

### 6.8.3.26 setLogger ( \$ SofortLibLogger )

Set the SofortLibLogger object

#### Parameters

*object* \$SofortLibLogger

The documentation for this class was generated from the following file:

- `sofortLib.php`

## 6.9 SofortLib\_CancelSr Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey= '')
- [setTransaction](#) (\$transaction, \$invoice=0)
- [setComment](#) (\$comment, \$invoice=0)
- [setCreditNoteNumber](#) (\$creditNoteNumber, \$invoice=0)
- [cancelInvoice](#) (\$transaction= '', \$invoice=0)
- [getInvoiceUrl](#) ()

### Protected Member Functions

- [\\_parseXml](#) ()
- [\\_handleErrors](#) ()

### 6.9.1 Detailed Description

This class is for confirming and changing statuses of invoices

eg: \$confirmObj = new [SofortLib\\_CancelSr](#)('yourapikey');

\$confirmObj->confirmInvoice('1234-456-789654-31321')->sendRequest();

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

#### Date:

2012-05-21 16:53:26 +0200 (Mo, 21 Mai 2012)

#### Version

[SofortLib](#) 1.5.4

**Id:**

sofortLib\_cancel\_sr.inc.php 4191 2012-05-21 14:53:26Z niehoff

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.9.2 Constructor & Destructor Documentation

### 6.9.2.1 `__construct ( $ configKey = " )`

create new confirm object

**Parameters**

*String* \$apikey your API-key

## 6.9.3 Member Function Documentation

### 6.9.3.1 `_handleErrors ( ) [protected]`

Error handling (override) (non-PHPdoc)

**See also**

[SofortLib.\\_handleErrors\(\)](#)

### 6.9.3.2 `_parseXml ( ) [protected]`

Parse the XML (override) (non-PHPdoc)

**See also**

[SofortLib\\_Abstract.\\_parseXml\(\)](#)

### 6.9.3.3 `cancelInvoice ( $ transaction = "", $ invoice = 0 )`

cancel the invoice

**Parameters**

*string* \$transaction the transaction id

**Returns**

[SofortLib\\_ConfirmSr](#)



### 6.9.3.4 `getInvoiceUrl ( )`

after you you changed/confirmed an invoice you can download the new invoice-pdf with this function

#### Returns

string url

### 6.9.3.5 `setComment ( $ comment, $ invoice = 0 )`

set a comment for refunds

#### Parameters

*string* \$arg

### 6.9.3.6 `setCreditNoteNumber ( $ creditNoteNumber, $ invoice = 0 )`

Setter for credit note number

#### Parameters

*string* \$creditNoteNumber

*int* \$invoice

### 6.9.3.7 `setTransaction ( $ transaction, $ invoice = 0 )`

Set the transaction you want to confirm/change

#### Parameters

*String* \$transaction Transaction Id

#### Returns

[SofortLib\\_ConfirmSr](#)

The documentation for this class was generated from the following file:

- `sofortLib_cancel_sr.inc.php`

## 6.10 SofortLib\_ClassicNotification Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$userId, \$projectId, \$password, \$hashFunction= 'sha1')

- [getNotification](#) (\$request)
- [isError](#) ()
- [getError](#) ()
- [getTransaction](#) ()
- [getAmount](#) ()
- [getUserVariable](#) (\$i=0)
- [getCurrency](#) ()
- [getTime](#) ()
- [getStatus](#) ()
- [getStatusReason](#) ()

## Protected Member Functions

- [\\_getHashHexValue](#) (\$data, \$hashFunction= 'sha1')

### 6.10.1 Detailed Description

Instance of this class handles the callback of SOFORT to notify about a status change, the classic way to do so

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

#### Date:

2012-11-30 12:10:48 +0100 (Fri, 30 Nov 2012)

#### Version

[SofortLib](#) 1.5.4

#### Id:

sofortLib\_classic\_notification.inc.php 5809 2012-11-30 11:10:48Z rotsch

#### Author

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

### 6.10.2 Constructor & Destructor Documentation

#### 6.10.2.1 `__construct ( $ userId, $ projectId, $ password, $ hashFunction = 'sha1' )`

Constructor for [SofortLib\\_ClassicNotification](#)

#### Parameters

*int* \$userId

*int* \$projectId  
*string* \$password  
*string* \$hashFunction

### 6.10.3 Member Function Documentation

#### 6.10.3.1 `_getHashHexValue ( $ data, $ hashFunction = ' sha1' )` [protected]

Getter for Hash Hex Value

##### Parameters

*string* \$data string to be hashed

##### Returns

string the hash

#### 6.10.3.2 `getAmount ( )`

Getter for amount

##### Returns

float

#### 6.10.3.3 `getCurrency ( )`

Getter for currency

##### Returns

string

#### 6.10.3.4 `getError ( )`

Get error if occurred

#### 6.10.3.5 `getNotification ( $ request )`

Get the Notification details

##### Parameters

*string* \$request (POST-Data)

**6.10.3.6 getStatus ( )**

Getter for status

**Returns**

string

**6.10.3.7 getStatusReason ( )**

Getter for status reason

**Returns**

string

**6.10.3.8 getTime ( )**

Getter for time return timestamp

**6.10.3.9 getTransaction ( )**

Getter for transactionId

**6.10.3.10 getUserVariable ( \$ i = 0 )**

Getter for user variables

**Parameters**

*int* \$i

**6.10.3.11 isError ( )**

Check if error occurred

**Returns**

boolean

The documentation for this class was generated from the following file:

- sofortLib\_classic\_notification.inc.php

## 6.11 SofortLib\_ConfirmSr Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey= '')
- [setTransaction](#) (\$transaction, \$invoice=0)
- [setInvoiceNumber](#) (\$invoiceNumber, \$invoice=0)
- [setCustomerNumber](#) (\$customerNumber, \$invoice=0)
- [setOrderNumber](#) (\$orderNumber, \$invoice=0)
- [setComment](#) (\$comment)
- [addItem](#) (\$itemId, \$productId, \$productType, \$title, \$description, \$quantity, \$unitPrice, \$tax)
- [removeItem](#) (\$productId, \$quantity=0)
- [updateCart](#) (\$cartItems=array())
- [cancelInvoice](#) (\$transaction= '')
- [confirmInvoice](#) (\$transaction= '')
- [getInvoiceUrl](#) (\$i=0)

### Protected Member Functions

- [\\_parseXml](#) ()
- [\\_handleErrors](#) ()

#### 6.11.1 Detailed Description

This class is for confirming and changing statuses of invoices

eg: \$confirmObj = new [SofortLib\\_ConfirmSr](#)('yourapikey');

\$confirmObj->confirmInvoice('1234-456-789654-31321')->sendRequest();

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

#### Date:

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

#### Version

[SofortLib](#) 1.5.4

#### Id:

sofortLib\_confirm\_sr.inc.php 5773 2012-11-23 16:15:47Z dehn

#### Author

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.11.2 Constructor & Destructor Documentation

### 6.11.2.1 `__construct ( $ configKey = " )`

create new confirm object

#### Parameters

*String* \$apikey your API-key

## 6.11.3 Member Function Documentation

### 6.11.3.1 `_handleErrors ( ) [protected]`

Handle errors if occurred (non-PHPdoc)

#### See also

[SofortLib.\\_handleErrors\(\)](#)

### 6.11.3.2 `_parseXml ( ) [protected]`

Parse the XML (override) (non-PHPdoc)

#### See also

[SofortLib\\_Abstract.\\_parseXml\(\)](#)

### 6.11.3.3 `addItem ( $ itemId, $ productNumber, $ productType, $ title, $ description, $ quantity, $ unitPrice, $ tax )`

add one item to the cart if you want to change the invoice just useable with api version 1.0

#### See also

[SofortLib\\_EditSr](#)

#### Parameters

*string* \$productNumber product number, EAN code, ISBN number or similar

*string* \$title description of this title

*double* \$unit\_price gross price of one item

*int* \$productType product type number see manual

*string* \$description additional description of this item

*int* \$quantity default 1

*int* \$tax tax in percent, default 19

#### 6.11.3.4 `cancelInvoice ( $ transaction = " )`

cancel the invoice just useable with api version 1.0

See also

[SofortLib\\_EditSr](#)

##### Parameters

*string* \$transaction the transaction id

##### Returns

[SofortLib\\_ConfirmSr](#)

#### 6.11.3.5 `confirmInvoice ( $ transaction = " )`

confirm the invoice

##### Parameters

*string* \$transaction the transaction id

##### Returns

[SofortLib\\_ConfirmSr](#)

#### 6.11.3.6 `getInvoiceUrl ( $ i = 0 )`

after you you changed/confirmed an invoice you can download the new invoice-pdf with this function

##### Returns

string url

#### 6.11.3.7 `removeItem ( $ productId, $ quantity = 0 )`

TODO: implement removal of items

See also

[SofortLib\\_EditSr](#)

##### Parameters

*\$productId*

*\$quantity*

**6.11.3.8 setComment ( \$ *comment* )**

set a comment for refunds just useable with api version 1.0

See also

[SofortLib\\_EditSr](#)

**Parameters**

*string* \$arg

**6.11.3.9 setCustomerNumber ( \$ *customerNumber*, \$ *invoice* = 0 )**

Setter for costumer numer

**Parameters**

*string* \$customerNumber

*int* \$invoice

**6.11.3.10 setInvoiceNumber ( \$ *invoiceNumber*, \$ *invoice* = 0 )**

Setter for invoice number

**Parameters**

*String* \$invoiceNumber

*object* \$invoice

**6.11.3.11 setOrderNumber ( \$ *orderNumber*, \$ *invoice* = 0 )**

Setter for order number

**Parameters**

*string* \$orderNumber

*unknown\_type* \$invoice

**6.11.3.12 setTransaction ( \$ *transaction*, \$ *invoice* = 0 )**

Set the transaction you want to confirm

**Parameters**

*String* \$transaction Transaction Id

**Returns**

[SofortLib\\_ConfirmSr](#)



### 6.11.3.13 updateCart ( \$ cartItems = *array*() )

just useable with api version 1.0

See also

[SofortLib\\_EditSr](#)

#### Parameters

*array* \$cartItems

The documentation for this class was generated from the following file:

- sofortLib\_confirm\_sr.inc.php

## 6.12 SofortLib\_Debit Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey= '')
- [sendRequest](#) ()
- [setProjectId](#) (\$id)
- [setDate](#) (\$date= '')
- [setSenderAccount](#) (\$bankCode, \$accountNumber, \$holder)
- [setSenderAccountNumber](#) (\$accountNumber)
- [setSenderBankCode](#) (\$bankCode)
- [setSenderHolder](#) (\$name)
- [setAmount](#) (\$amount)
- [addUserVariable](#) (\$userVariable)
- [addReason](#) (\$reason)
- [setReason](#) (\$reason1, \$reason2= '')
- [getTransactionId](#) ()
- [getReason](#) (\$i=0)
- [getAmount](#) ()
- [getUserVariable](#) (\$i=0)
- [getDate](#) ()
- [isError](#) (\$paymentMethod= 'all', \$message= '')
- [getError](#) (\$paymentMethod= 'all', \$message= '')
- [getResponse](#) ()

### 6.12.1 Detailed Description

class for handling debit/lastschrift

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

**Date:**

2013-02-20 18:28:51 +0100 (Wed, 20 Feb 2013)

**Version**

[SofortLib](#) 1.5.4

**Id:**

sofortLib\_debit.inc.php 5989 2013-02-20 17:28:51Z niehoff

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.12.2 Constructor & Destructor Documentation

### 6.12.2.1 `__construct ( $ configKey = " )`

Constructor for [SofortLib\\_Debit](#)

**Parameters**

*string* \$configKey

## 6.12.3 Member Function Documentation

### 6.12.3.1 `addReason ( $ reason )`

adds another reason (verwendungszweck) only first two can be used, 27 characters each

**Parameters**

*\$reason* string

**Returns**

[SofortLib\\_Debit](#) \$this

### 6.12.3.2 `addUserVariable ( $ userVariable )`

add another user-variable to this transfer this variable could be a customer-number or similar and will help you identify this transfer later

**Parameters**

*String* \$userVariable max 255 characters

**Returns**

[SofortLib\\_Debit](#) \$this

**6.12.3.3** `getAmount ( )`

Getter for amount

**6.12.3.4** `getDate ( )`

Getter for response's date

**6.12.3.5** `getError ( $ paymentMethod = 'a11', $ message = " )`

Get the error occurred (non-PHPdoc)

See also

[SofortLib.getError\(\)](#)

**6.12.3.6** `getReason ( $ i = 0 )`

Getter for payment reason

**Parameters**

*int* \$i

**6.12.3.7** `getResponse ( )`

Getter for response

**6.12.3.8** `getTransactionId ( )`

get Transaction-Id of this Transfer

**Returns**

String transaction-id

**6.12.3.9** `getUserVariable ( $ i = 0 )`

Getter for user variables

**Parameters**

*int* \$i

**6.12.3.10** `isError ( $ paymentMethod = 'a11', $ message = '' )`

Has an error occurred (non-PHPdoc)

**See also**

[SofortLib.isError\(\)](#)

**6.12.3.11** `sendRequest ( )`

send this debitpay and get response

**Returns**

boolean true if transaction was accepted, false otherwise

**6.12.3.12** `setAmount ( $ amount )`

set amount of this transfer needs to be a decimal e.g. 2.24

**Parameters**

*float* \$amount amount of this transfer

**6.12.3.13** `setDate ( $ date = '' )`

sets date of this debitpay automatically called in constructor and set to today

**Parameters**

*String* \$date date in Format Y-m-d (eg: 2011-01-20), default: today

**Returns**

[SofortLib\\_Debit](#) \$this

**6.12.3.14** `setProjectId ( $ id )`

Project id Id of your Sofortlastschrift project

**Parameters**

*int* \$id project id

**Returns**

[SofortLib\\_Debit](#) \$this

**6.12.3.15 setReason ( \$ reason1, \$ reason2 = " )**

set reason (verwendungszweck) of this transfer two lines possible, 27 characters each

**Parameters**

*\$reason1*

*\$reason2*

**Returns**

[SofortLib\\_Debit](#) \$this

**6.12.3.16 setSenderAccount ( \$ bankCode, \$ accountNumber, \$ holder )**

set data of account

**Parameters**

*String* \$bankCode bank code of bank

*String* \$accountNumber account number

*String* \$holder Name/Holder of this account

**Returns**

[SofortLib\\_Debit](#) \$this

**6.12.3.17 setSenderAccountNumber ( \$ accountNumber )**

set data of account

**Parameters**

*String* \$accountNumber account number

**Returns**

[SofortLib\\_Debit](#) \$this

**6.12.3.18 setSenderBankCode ( \$ bankCode )**

set data of account

**Parameters**

*String* \$bankCode bank code of bank

**Returns**

[SofortLib\\_Debit](#) \$this

### 6.12.3.19 setSenderHolder ( \$ name )

set data of account

#### Parameters

*String* \$name Name/Holder of this account

#### Returns

[SofortLib\\_Debit](#) \$this

The documentation for this class was generated from the following file:

- `sofortLib_debit.inc.php`

## 6.13 SofortLib\_EditSr Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey= '')
- [setTransaction](#) (\$transaction, \$invoice=0)
- [setInvoiceNumber](#) (\$invoiceNumber, \$invoice=0)
- [setCustomerNumber](#) (\$customerNumber, \$invoice=0)
- [setOrderNumber](#) (\$orderNumber, \$invoice=0)
- [setComment](#) (\$comment, \$invoice=0)
- [addItem](#) (\$itemId, \$productNumber, \$productType, \$title, \$description, \$quantity, \$unitPrice, \$tax, \$invoice=0)
- [updateCart](#) (\$cartItems=array(), \$invoice=0)
- [getInvoiceUrl](#) ()

### Protected Member Functions

- [\\_parseXml](#) ()
- [\\_handleErrors](#) ()

### 6.13.1 Detailed Description

This class is for confirming and changing statuses of invoices

eg: `$confirmObj = new SofortLib\_EditSr('yourapikey');`

`$confirmObj->confirmInvoice('1234-456-789654-31321')->sendRequest();`

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

**Date:**

2012-05-21 16:53:26 +0200 (Mo, 21 Mai 2012)

**Version**

[SofortLib](#) 1.5.4

**Id:**

sofortLib\_edit\_sr.inc.php 4191 2012-05-21 14:53:26Z niehoff

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.13.2 Constructor & Destructor Documentation

### 6.13.2.1 `__construct ( $ configKey = " )`

Constructor for [SofortLib\\_EditSr](#)

**Parameters**

*String* \$apikey your API-key

## 6.13.3 Member Function Documentation

### 6.13.3.1 `_handleErrors ( ) [protected]`

Error handling (override)

**See also**

[SofortLib.\\_handleErrors\(\)](#)

### 6.13.3.2 `_parseXml ( ) [protected]`

Parse the XML (override)

**See also**

[SofortLib\\_Abstract.\\_parseXml\(\)](#)

### 6.13.3.3 `addItem ( $ itemId, $ productNumber, $ productType, $ title, $ description, $ quantity, $ unitPrice, $ tax, $ invoice = 0 )`

add one item to the cart if you want to change the invoice

**Parameters**

*string* \$itemId itemId  
*double* \$unit\_price gross price of one item  
*int* \$quantity default 1

**6.13.3.4 getInvoiceUrl ( )**

after you changed/confirmed an invoice you can download the new invoice-pdf with this function

**Returns**

string url

**6.13.3.5 setComment ( \$ comment, \$ invoice = 0 )**

set a comment for refunds just useable with api version 1.0

**Parameters**

*string* \$arg

**6.13.3.6 setCustomerNumber ( \$ customerNumber, \$ invoice = 0 )**

Setter for customer's number

**Parameters**

*string* \$customerNumber  
*int* \$invoice

**6.13.3.7 setInvoiceNumber ( \$ invoiceNumber, \$ invoice = 0 )**

Setter for invoice's number

**Parameters**

*string* \$invoiceNumber  
*int* \$invoice

**6.13.3.8 setOrderNumber ( \$ orderNumber, \$ invoice = 0 )**

Setter for order's number

**Parameters**

*string* \$orderNumber  
*int* \$invoice



### 6.13.3.9 setTransaction ( \$ transaction, \$ invoice = 0 )

Setter for transaction

#### Parameters

*string* \$transaction

*int* \$invoice

### 6.13.3.10 updateCart ( \$ cartItems = array (), \$ invoice = 0 )

Update the invoice's cart via passing an array

#### Parameters

*array* \$cartItems

*int* \$invoice

The documentation for this class was generated from the following file:

- sofortLib\_edit\_sr.inc.php

## 6.14 SofortLib\_iDeal\_Banks Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey, \$apiUrl= '')
- [getBanks](#) ()

### Protected Member Functions

- [\\_parseXml](#) ()

### 6.14.1 Detailed Description

This class encapsulates retrieval of listed banks of the Netherlands

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

#### Date:

2013-02-27 11:37:15 +0100 (Wed, 27 Feb 2013)

**Version**

[SofortLib](#) 1.5.4

**Id:**

sofortLib\_ideal\_banks.inc.php 6029 2013-02-27 10:37:15Z rotsch

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.14.2 Constructor & Destructor Documentation

### 6.14.2.1 `__construct ( $ configKey, $ apiUrl = " )`

Constructor for [SofortLib\\_iDeal\\_Banks](#)

**Parameters**

*string* \$configKey

*string* \$apiUrl

## 6.14.3 Member Function Documentation

### 6.14.3.1 `_parseXml ( )` [protected]

Parse the xml (override) (non-PHPdoc)

**See also**

`SofortLib_Abstract._parseXml()`

### 6.14.3.2 `getBanks ( )`

Getter for bank list

The documentation for this class was generated from the following file:

- `sofortLib_ideal_banks.inc.php`

## 6.15 SofortLib\_iDealClassic Class Reference

Inherits [SofortLib\\_SofortueberweisungClassic](#).

## Public Member Functions

- [\\_\\_construct](#) (\$configKey, \$password, \$hashFunction= 'sha1')
- [setSenderCountryId](#) (\$senderCountryId= 'NL')
- [setSenderBankCode](#) (\$senderBankCode)
- [getError](#) ()
- [getRelatedBanks](#) ()

## Protected Member Functions

- [\\_getPaymentDomain](#) ()

### 6.15.1 Detailed Description

iDeal\_Classic extends Sofortueberweisung\_Classic, implementing payment via iDeal Setup a session within iDeal using the classic api You get the so called payment-url after successful configuration Payment is enabled with this url being sent to iDeal

eg: \$sofort = \$sofortLib\_iDealClassic = new [SofortLib\\_iDealClassic](#) (\$configurationKey, \$password, \$hashfunction = 'sha1'); \$sofort->[getRelatedBanks](#)(); //get all iDEAL-Banks \$sofort->[getPaymentUrl](#)(); //returns paymentUrl including (including ...&hash=1234567890&...)

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

#### Date:

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

#### Version

[SofortLib](#) 1.5.4

#### Id:

sofortLib\_ideal\_classic.php 5773 2012-11-23 16:15:47Z dehn

#### Author

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

### 6.15.2 Constructor & Destructor Documentation

#### 6.15.2.1 `__construct ( $ configKey, $ password, $ hashFunction = ' sha1' )`

Constructor for [SofortLib\\_iDealClassic](#)

**Parameters**

*string* \$configKey  
*string* \$password  
*string* \$hashFunction

**6.15.3 Member Function Documentation****6.15.3.1 `_getPaymentDomain ( )` [protected]**

Getter for the payment domain (non-PHPdoc)

**See also**

[SofortLib\\_SofortueberweisungClassic.\\_getPaymentDomain\(\)](#)

Reimplemented from [SofortLib\\_SofortueberweisungClassic](#).

**6.15.3.2 `getError ( )`**

Getter for occurred errors (non-PHPdoc)

**See also**

[SofortLib\\_SofortueberweisungClassic.getError\(\)](#)

Reimplemented from [SofortLib\\_SofortueberweisungClassic](#).

**6.15.3.3 `getRelatedBanks ( )`**

Get related banks of iDeal

**Returns**

array

**6.15.3.4 `setSenderBankCode ( $ senderBankCode )`**

Set sender's bank code

**Parameters**

*string* \$senderBankCode

**Returns**

instance

### 6.15.3.5 setSenderCountryId ( \$ senderCountryId = 'NL' )

Set sender's country id

#### Parameters

*string* \$senderCountryId

#### Returns

instance

The documentation for this class was generated from the following file:

- sofortLib\_ideal\_classic.php

## 6.16 SofortLib\_Multipay Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey= '')
- [setLanguageCode](#) (\$arg)
- [setTimeout](#) (\$arg)
- [setEmailCustomer](#) (\$arg)
- [setPhoneNumberCustomer](#) (\$arg)
- [addUserVariable](#) (\$arg)
- [setSenderAccount](#) (\$bankCode, \$accountNumber, \$holder)
- [setAmount](#) (\$arg, \$currency= 'EUR')
- [setReason](#) (\$arg, \$arg2= '')
- [setSuccessLinkRedirect](#) (\$arg)
- [setSuccessUrl](#) (\$successUrl, \$redirect=true)
- [setAbortUrl](#) (\$arg)
- [setTimeoutUrl](#) (\$arg)
- [setNotificationUrl](#) (\$arg)
- [addNotificationUrl](#) (\$arg)
- [setNotificationEmail](#) (\$arg)
- [addNotificationEmail](#) (\$arg)
- [setVersion](#) (\$arg)
- [setSofortueberweisung](#) (\$amount= '')
- [setSofortueberweisungCustomerprotection](#) (\$customerProtection=true)
- [setSofortlastschrift](#) (\$amount= '')
- [setSofortlastschriftAddress](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE')
- [setLastschrift](#) (\$amount= '')
- [setLastschriftBaseCheckDisabled](#) ()
- [setLastschriftExtendedCheckDisabled](#) ()
- [setLastschriftMobileCheckDisabled](#) ()

- [setLastschriftAddress](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE')
- [setSofortrechnung](#) ()
- [setSofortvorkasse](#) (\$amount= '')
- [setSofortvorkasseCustomerprotection](#) (\$customerProtection=true)
- [setSofortrechnungCustomerId](#) (\$arg)
- [setSofortrechnungOrderId](#) (\$arg)
- [setDebitorVatNumber](#) (\$vatNumber)
- [setSofortrechnungInvoiceAddress](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE', \$nameAdditive= '', \$streetAdditive= '', \$companyName= '')
- [setSofortrechnungShippingAddress](#) (\$firstname, \$lastname, \$street, \$streetNumber, \$zipcode, \$city, \$salutation, \$country= 'DE', \$nameAdditive= '', \$streetAdditive= '', \$companyName= '')
- [addSofortrechnungItem](#) (\$itemId, \$productNumber, \$title, \$unitPrice, \$productType=0, \$description= '', \$quantity=1, \$tax=19)
- [setSofortrechnungComment](#) (\$comment)
- [removeSofortrechnungItem](#) (\$itemId)
- [updateSofortrechnungItem](#) (\$itemId, \$quantity, \$unitPrice)
- [getSofortrechnungItemAmount](#) (\$itemId)
- [setSofortrechnungTimeForPayment](#) (\$arg)
- [sendValidationRequest](#) ()
- [getSofortrechnungItem](#) (\$itemId)
- [getSofortrechnungItems](#) ()
- [getPaymentUrl](#) ()
- [getPaymentMethod](#) (\$i=0)
- [isSofortueberweisung](#) ()
- [isSofortvorkasse](#) ()
- [isSofortlastschrift](#) ()
- [isLastschrift](#) ()
- [isSofortrechnung](#) ()
- [isConsumerProtection](#) (\$product)
- [isDebitpayCheckDisabled](#) (\$product, \$check)
- [getTransactionId](#) ()

## Protected Member Functions

- [\\_parseXml](#) ()

### 6.16.1 Detailed Description

Setup a multipay payment session after the configuration of multiple payment methods you will receive an url and a transaction id, your customer should be redirected to this url you can use the transaction id for future reference of this payment

example by usage: `$objMultipay = new SofortLib_Multipays('my-API-KEY');`  
`$objMultipay->setSofortueberweisung();` //OR `setSofortrechnung(), setSofortvorkasse()` etc. `$objMultipay->set...($param);` //set params for PNAG-API (watch API-documentation for needed params) `$objMultipay->add...($param);` //add params for PNAG-API (watch API-documentation for needed params) `$errorsAndWarnings = $objMultipay->validateRequest();` //send param against the PNAG-API without setting an order ... make own validation of `$errorsAndWarnings` and if ok ... `$objMultipay->sendRequest();` //set the order at PNAG `$errorsAndWarnings = $objMultipay->getErrors();` //should not occur, if validation was ok ... make own validation of `$errorsAndWarnings` and if ok ... finish order in the shopsystem

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[\[http://www.gnu.org/licenses/gpl-2.0.html\]](http://www.gnu.org/licenses/gpl-2.0.html)

#### Date:

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

#### Version

[SofortLib](#) 1.5.4

#### Id:

sofortLib\_multipay.inc.php 5773 2012-11-23 16:15:47Z dehn

#### Author

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.16.2 Constructor & Destructor Documentation

### 6.16.2.1 `__construct ( $ configKey = " )`

create a new payment object

#### Parameters

*string* \$apikey your API key

*int* \$projectId your project id

## 6.16.3 Member Function Documentation

### 6.16.3.1 `_parseXml ( )` **[protected]**

Parse the XML (override) (non-PHPdoc)

#### See also

`SofortLib_Abstract._parseXml()`

### 6.16.3.2 addNotificationEmail ( \$ arg )

you can set set multiple emails for receiving notifications

#### Parameters

*string* \$arg email

#### Returns

[SofortLib\\_Multipay](#)

### 6.16.3.3 addNotificationUrl ( \$ arg )

you can set set multiple urls for receiving notifications this might be helpfull if you have several systems for processing an order (e.g. an ERP system)

#### Parameters

*string* \$arg url

#### Returns

[SofortLib\\_Multipay](#)

### 6.16.3.4 addSofortrechnungItem ( \$ itemId, \$ productNumber, \$ title, \$ unitPrice, \$ productType = 0, \$ description = "", \$ quantity = 1, \$ tax = 19 )

add one item to the cart

#### Parameters

*int* \$itemId unique item id

*string* \$productNumber product number, EAN code, ISBN number or similar

*string* \$title description of this title

*double* \$unit\_price gross price of one item

*int* \$productType product type number see manual (0=other, 1=shipping, ...)

*string* \$description additional description of this item

*int* \$quantity default 1

*int* \$tax tax in percent, default 19



### 6.16.3.5 addUserVariable ( \$ arg )

add another variable this can be your internal order id or similar

#### Parameters

*string* \$arg the contents of the variable

#### Returns

[SofortLib\\_Multipay](#)

### 6.16.3.6 getPaymentMethod ( \$ i = 0 )

Getter for payment method

#### Parameters

*int* \$i

### 6.16.3.7 getPaymentUrl ( )

after configuration and sending this request you can use this function to redirect the customer to the payment form

#### Returns

string url of payment form

### 6.16.3.8 getSofortrechnungItem ( \$ itemId )

Getter for invoice's item

#### Parameters

*int* \$itemId

### 6.16.3.9 getSofortrechnungItemAmount ( \$ itemId )

Getter for invoice's item

#### Parameters

*int* \$itemId

**6.16.3.10** `getSofortrechnungItems ( )`

Getter for all invoice's items

**6.16.3.11** `getTransactionId ( )`

use this id to track the transaction

**Returns**

string transaction id

**6.16.3.12** `isConsumerProtection ( $ product )`

Check if consumer protection / customer protection enabled

**Parameters**

*string* \$product

**6.16.3.13** `isDebitpayCheckDisabled ( $ product, $ check )`

Check if debit pay check disabled

**Parameters**

*string* \$product

*boolean* \$check

**6.16.3.14** `isLastschrift ( )`

Is lastschrift by sofort

**6.16.3.15** `isSofortlastschrift ( )`

Check if it is a sofortlastschrift

**6.16.3.16** `isSofortrechnung ( )`

Is rechnung by sofort

**6.16.3.17** `isSofortueberweisung ( )`

Is sofortüberweisung

**6.16.3.18 isSofortvorkasse ( )**

Is vorkasse by sofort

**6.16.3.19 removeSofortrechnungItem ( \$ *itemId* )**

Remove one item from cart

**Parameters**

*\$itemId*

**Returns**

boolean

**6.16.3.20 sendValidationRequest ( )**

makes a request against the pnag-API and returns all API-Fault/Warnings it doesnt result in an order at pnag!

**Returns**

bool

**6.16.3.21 setAbortUrl ( \$ *arg* )**

the customer will be redirected to this url if he uses the abort link on the payment form, should redirect him back to his cart or to the payment selection page

**Parameters**

*string* \$arg url for aborting the transaction

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.22 setAmount ( \$ *arg*, \$ *currency* = 'EUR' )**

amount of this payment

**Parameters**

*double* \$arg

*string* \$currency currency of this transaction, default EUR

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.23 setDebitorVatNumber ( \$ *vatNumber* )**

set debitor vat number for invoice

**Parameters**

*string* \$vatNumber

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.24 setEmailCustomer ( \$ *arg* )**

set the email address of the customer this will be used for sofortvorkasse and so-fortrechnung

**Parameters**

*string* \$arg email address

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.25 setLanguageCode ( \$ *arg* )**

the language code will help in determing what language to use when displaying the payment form, other data like browser settings and ip will be used as well

**Parameters**

*string* \$arg de|en|nl|fr ...

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.26 setLastschrift ( \$ *amount* = " )**

add lastschrift as payment method

**Parameters**

*double* \$amount this amount only applies to this payment method

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.27 setLastschriftAddress ( \$ *firstname*, \$ *lastname*, \$ *street*, \$ *streetNumber*, \$ *zipcode*, \$ *city*, \$ *salutation*, \$ *country* = 'DE' )**

set the address of the customer for address validation, this should be the invoice address of the customer

**Parameters**

*string* \$firstname

*string* \$lastname

*string* \$street

*string* \$streetNumber

*string* \$zipcode

*string* \$city

*int* \$salutation [2|3] 2=Mr. 3=Mrs.

*string* \$country country code, only DE allowed at the moment

**Returns**

[SofortLib\\_Multipay](#) object

**6.16.3.28 setLastschriftBaseCheckDisabled ( )**

Setter for base checks disabled of Lastschrift

**6.16.3.29 setLastschriftExtendedCheckDisabled ( )**

Setter for extended checks disabled of Lastschrift

**6.16.3.30 setLastschriftMobileCheckDisabled ( )**

Setter for mobile checks disabled of Lastschrift

**6.16.3.31 setNotificationEmail ( \$ *arg* )**

set the email address where you want notification about status changes being sent to.

**Parameters**

*string* \$arg email address

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.32 setNotificationUrl ( \$ arg )**

set the url where you want notification about status changes being sent to. Use [SofortLib\\_Notification](#) and [SofortLib\\_TransactionData](#) to further process that notification

**Parameters**

*string* \$arg url

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.33 setPhoneNumberCustomer ( \$ arg )**

set the phone number of the customer

**Parameters**

*string* \$arg phone number

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.34 setReason ( \$ arg, \$ arg2 = " )**

set the reason values of this transfer

**Parameters**

*string* \$arg max 27 characters

*string* \$arg2 max 27 characters

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.35 setSenderAccount ( \$ bankCode, \$ accountNumber, \$ holder )**

set data of account

**Parameters**

*string* \$bank\_code bank code of bank

*string* \$account\_number account number

*string* \$holder Name/Holder of this account

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.36** `setSofortlastschrift ( $ amount = " )`

add sofortlastschrift as payment method

**Parameters**

*double* \$amount this amount only applies to this payment method

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.37** `setSofortlastschriftAddress ( $ firstname, $ lastname, $ street, $ streetNumber, $ zipcode, $ city, $ salutation, $ country = 'DE' )`

set the address of the customer for address validation, this should be the invoice address of the customer

**Parameters**

*string* \$firstname

*string* \$lastname

*string* \$street

*string* \$streetNumber

*string* \$zipcode

*string* \$city

*int* \$salutation [2|3] 2=Mr. 3=Mrs.

*string* \$country country code, only DE allowed at the moment

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.38** `setSofortrechnung ( )`

add sofortrechnung as payment method if you use this payment method you have to provide the customer address and cart as well the total amount of this payment method will be determined by the total of the cart

**Returns**

[SofortLib\\_Multipay](#) object

**6.16.3.39** `setSofortrechnungCustomerId ( $ arg )`

set the customer id which will appear on top of the invoice

**Parameters**

*int* \$arg

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.40** `setSofortrechnungInvoiceAddress ( $ firstname, $ lastname, $ street, $ streetNumber, $ zipcode, $ city, $ salutation, $ country = 'DE', $ nameAdditive = "", $ streetAdditive = "", $ companyName = "" )`

set the invoice address of the customer

**Parameters**

*string* \$firstname

*string* \$lastname

*string* \$street

*string* \$streetNumber

*string* \$zipcode

*string* \$city

*int* \$salutation [2|3] 2 = Mr. 3 = Mrs.

*string* \$country country code, only DE allowed at the moment

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.41** `setSofortrechnungOrderId ( $ arg )`

set the order id which will appear on top of the invoice

**Parameters**

*int* \$arg

**Returns**

[SofortLib\\_Multipay](#) \$this



**6.16.3.42** `setSofortrechnungShippingAddress ( $ firstname, $ lastname, $ street, $ streetNumber, $ zipcode, $ city, $ salutation, $ country = 'DE', $ nameAdditive = "", $ streetAdditive = "", $ companyName = "" )`

set the shipping address of the customer

#### Parameters

*string* \$firstname  
*string* \$lastname  
*string* \$street  
*string* \$streetNumber  
*string* \$zipcode  
*string* \$city  
*int* \$salutation [2|3] 1 = Mr. 2 = Mrs.  
*string* \$country country code, only DE allowed at the moment

#### Returns

[SofortLib\\_Multipay](#) \$this

**6.16.3.43** `setSofortrechnungTimeForPayment ( $ arg )`

Setter for invoice's time for payment

#### Parameters

*string* \$arg

**6.16.3.44** `setSofortrechnungComment ( $ comment )`

Setter for commenting Rechnung by sofort

#### Parameters

*string* \$comment

**6.16.3.45** `setSofortueberweisung ( $ amount = "" )`

add sofortueberweisung as payment method

#### Parameters

*double* \$amount this amount only applies to this payment method

#### Returns

[SofortLib\\_Multipay](#) \$this

**6.16.3.46   `setSofortueberweisungCustomerprotection ( $ customerProtection = true )`**

add sofortueberweisung as payment method adds customer protection

**Parameters**

*double* \$amount this amount only applies to this payment method

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.47   `setSofortvorkasse ( $ amount = " )`**

add sofortvorkasse as payment method

**Parameters**

*double* \$amount this amount only applies to this payment method

**Returns**

[SofortLib\\_Multipay](#) objet

**6.16.3.48   `setSofortvorkasseCustomerprotection ( $ customerProtection = true )`**

add sofortvorkasse as payment method adds customer protection

**Parameters**

*double* \$amount this amount only applies to this payment method

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.16.3.49   `setSuccessLinkRedirect ( $ arg )`**

Setter for redirecting the success link automatically

**Parameters**

*boolean* \$arg

**6.16.3.50 setSuccessUrl ( \$ successUrl, \$ redirect = true )**

the customer will be redirected to this url after a successful transaction, this should be a page where a short confirmation is displayed

**Parameters**

*string* \$arg the url after a successful transaction

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.51 setTimeout ( \$ arg )**

timeout how long this transaction configuration will be valid for this is the time between the generation of the payment url and the user completing the form, should be at least two to three minutes defaults to unlimited if not set

**Parameters**

*int* \$arg timeout in seconds

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.52 setTimeoutUrl ( \$ arg )**

if the customer takes too much time or if your timeout is set too short he will be redirected to this page

**Parameters**

*string* \$arg url

**Returns**

[SofortLib\\_Multipay](#)

**6.16.3.53 setVersion ( \$ arg )**

set the version of this payment module this is helpfull so the support staff can easily find out if someone uses an outdated module

**Parameters**

*string* \$arg version string of your module

**Returns**

[SofortLib\\_Multipay](#)

#### 6.16.3.54 updateSofortrechnungItem ( \$ itemId, \$ quantity, \$ unitPrice )

Update one item in cart

##### Parameters

*\$itemId*

*\$quantity*

*\$unit\_price*

##### Returns

boolean

The documentation for this class was generated from the following file:

- sofortLib\_multipay.inc.php

## 6.17 SofortLib\_Notification Class Reference

### Public Member Functions

- [\\_\\_construct](#) ()
- [getNotification](#) (\$source= 'php://input')
- [sendRequest](#) ()
- [getTime](#) ()
- [getTransactionId](#) ()

#### 6.17.1 Detailed Description

Instance of this class handles the callback of Payment Network to notify about a status change

In rare cases notifications might be doubled or even wrong altogether (if send by a malicious user). So don't use this to change your status but instead use the transaction id to query the webservice for detailed data ([SofortLib\\_TransactionData](#))

eg: \$notificationObj = new SofortLib\_Notification();

\$transactionId = \$notificationObj->[getNotification](#)();

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

##### Date:

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

**Version**

[SofortLib](#) 1.5.4

**Id:**

sofortLib\_notification.inc.php 5773 2012-11-23 16:15:47Z dehn

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

**6.17.2 Constructor & Destructor Documentation****6.17.2.1 \_\_construct ( )**

Constructor for [SofortLib\\_Notification](#)

**6.17.3 Member Function Documentation****6.17.3.1 getNotification ( \$ source = 'php://input' )**

reads the input and tries to read the transaction id

**Returns**

array transactionid=>status

**6.17.3.2 getTime ( )**

Getter for variable time

**6.17.3.3 getTransactionId ( )**

Getter for transaction

**6.17.3.4 sendRequest ( )**

Sending a request is not possible (non-PHPdoc)

**See also**

[SofortLib\\_Abstract.sendRequest\(\)](#)

The documentation for this class was generated from the following file:

- [sofortLib\\_notification.inc.php](#)

## 6.18 SofortLib\_Refund Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$configKey= '')
- [sendRequest](#) ()
- [addRefund](#) (\$transaction, \$amount, \$comment= '')
- [setSenderAccount](#) (\$bankCode, \$accountNumber, \$holder= '')
- [setTitle](#) (\$arg)
- [getTransactionId](#) (\$i=0)
- [getAmount](#) (\$i=0)
- [getStatus](#) (\$i=0)
- [getComment](#) (\$i=0)
- [getTitle](#) ()
- [getRefundError](#) (\$i=0)
- [isRefundError](#) (\$i=0)
- [getDta](#) ()
- [getAsArray](#) ()
- [getStatusArray](#) ()

### Protected Member Functions

- [\\_parseXml](#) ()
- [\\_handleErrors](#) ()

#### 6.18.1 Detailed Description

class for refund/rueckbuchung

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

#### Date:

2012-11-23 17:26:25 +0100 (Fri, 23 Nov 2012)

#### Version

[SofortLib](#) 1.5.4

#### Id:

sofortLib\_refund.inc.php 5776 2012-11-23 16:26:25Z dehn

#### Author

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

## 6.18.2 Constructor & Destructor Documentation

### 6.18.2.1 `__construct ( $ configKey = " )`

Constructor for [SofortLib\\_Refund](#)

#### Parameters

*string* \$configKey

## 6.18.3 Member Function Documentation

### 6.18.3.1 `_handleErrors ( ) [protected]`

Handle errors occurred

#### See also

[SofortLib.\\_handleErrors\(\)](#)

### 6.18.3.2 `_parseXml ( ) [protected]`

Parse the XML (override)

#### See also

[SofortLib\\_Abstract.\\_parseXml\(\)](#)

### 6.18.3.3 `addRefund ( $ transaction, $ amount, $ comment = " )`

add a new refund to this message

#### Parameters

*string* \$transaction transaction id of transfer you want to refund

*float* \$amount amount of money to refund, less or equal to amount of original transfer

*string* \$comment comment that will be displayed in admin-menu later

#### Returns

[SofortLib\\_Refund](#) \$this

### 6.18.3.4 `getAmount ( $ i = 0 )`

Getter for amounts

#### Parameters

*int* \$i

**6.18.3.5   getFromArray (   )**

Getter for response, as an array

**6.18.3.6   getComment ( \$ i = 0 )**

Getter for comments

**Parameters**

*int* \$i

**6.18.3.7   getDta (   )**

Getter for DTA (MT940)

**6.18.3.8   getRefundError ( \$ i = 0 )**

Getter for refund's errors

**Parameters**

*int* \$i

**6.18.3.9   getStatus ( \$ i = 0 )**

Getter for statuses

**Parameters**

*int* \$i

**6.18.3.10   getStatusArray (   )**

Getter for status array

**6.18.3.11   getTitle (   )**

Getter for refund's title

**6.18.3.12   getTransactionId ( \$ i = 0 )**

Getter for transactions

**Parameters**

*int* \$i



**6.18.3.13 isRefundError ( \$ i = 0 )**

Has an error occurred for refund

**Parameters**

*int* \$i

**6.18.3.14 sendRequest ( )**

send this message and get response

**Returns**

array transactionid=>status

**6.18.3.15 setSenderAccount ( \$ bankCode, \$ accountNumber, \$ holder = " )**

set data of account

**Parameters**

*string* \$bank\_code bank code of bank

*string* \$account\_number account number

*string* \$holder Name/Holder of this account

**Returns**

[SofortLib\\_Multipay](#) \$this

**6.18.3.16 setTitle ( \$ arg )**

Setter for title

**Parameters**

*string* \$arg

The documentation for this class was generated from the following file:

- `sofortLib_refund.inc.php`

## 6.19 SofortLib\_SofortueberweisungClassic Class Reference

Inherited by [SofortLib\\_iDealClassic](#).

## Public Member Functions

- [\\_\\_construct](#) (\$userId, \$projectId, \$password, \$hashFunction= 'sha1', \$paymentUrl=null)
- [setAmount](#) (\$arg, \$currency= 'EUR')
- [setSenderHolder](#) (\$senderHolder)
- [setSenderAccountNumber](#) (\$senderAccountNumber)
- [setReason](#) (\$arg, \$arg2= '')
- [addUserVariable](#) (\$arg)
- [setSuccessUrl](#) (\$arg)
- [setAbortUrl](#) (\$arg)
- [setNotificationUrl](#) (\$arg)
- [setVersion](#) (\$arg)
- [getPaymentUrl](#) ()
- [isError](#) ()
- [getError](#) ()
- [getHashHexValue](#) (\$data, \$hashFunction= 'sha1')
- [getSupportedHashAlgorithm](#) ()
- [getSupportedHashAlgorithms](#) ()

## Static Public Member Functions

- static [generatePassword](#) (\$length=24)

## Protected Member Functions

- [\\_getPaymentDomain](#) ()

### 6.19.1 Detailed Description

Setup a sofortueberweisung.de session using the classic api after the configuration of the configuration you will receive an url and a transaction id, your customer should be redirected to this url

Called by the `sofortLib.php/sofortLib_ideal_classic.php` etc. `$sofort->new SofortLib_SofortueberweisungClassic( $userid, $projectId, $password [, $hashfunction='sha1'] );` `$sofort->set...();` //set params for Hashcalculation `$sofort->set...();` //set more params for Hashcalculation `$sofort->getPaymentUrl();` Notice: sometimes `getPaymentUrl()` must be overwritten by calling class because of changed hash-params

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[<http://www.gnu.org/licenses/gpl-2.0.html>]

Date:

2012-11-23 17:15:47 +0100 (Fri, 23 Nov 2012)

**Version**

[SofortLib](#) 1.5.4

**Id:**

sofortLib\_sofortueberweisung\_classic.php 5773 2012-11-23 16:15:47Z dehn

**Author**

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))

**6.19.2 Constructor & Destructor Documentation****6.19.2.1** `__construct ( $ userId, $ projectId, $ password, $ hashFunction = 'sha1', $ paymentUrl = null )`

Constructor for [SofortLib\\_SofortueberweisungClassic](#)

**Parameters**

*int* \$userId  
*int* \$projectId  
*string* \$password  
*string* \$hashFunction  
*string* \$paymentUrl

**6.19.3 Member Function Documentation****6.19.3.1** `_getPaymentDomain ( )` [**protected**]

Getter for payment domain

Reimplemented in [SofortLib\\_iDealClassic](#).

**6.19.3.2** `addUserVariable ( $ arg )`

User variables can be added here

**Parameters**

*string* \$arg

**6.19.3.3** `static generatePassword ( $ length = 24 )` [**static**]

Generate a password

**Parameters**

*int* [optional] \$length length of return value, default 24

**Returns**

string

**6.19.3.4   getError (   )**

Getter for error occurred

Reimplemented in [SofortLib\\_iDealClassic](#).

**6.19.3.5   getHashHexValue ( \$ data, \$ hashFunction = 'sha1' )**

Get the hash value

**Parameters**

*string* \$data string to be hashed

**Returns**

string the hash

**6.19.3.6   getPaymentUrl (   )**

Getter for payment URL

**6.19.3.7   getSupportedHashAlgorithm (   )**

checks wich hash algorithms are supported by the server and returns the best one

**Returns**

sha512|sha256|sha1|md5|empty string

**6.19.3.8   getSupportedHashAlgorithms (   )**

checks wich hash algorithms are supported by the server

**Returns**

array with all supported algorithms, preferred as first one (index 0)

**6.19.3.9   isError (   )**

Has an error occurred

### 6.19.3.10 setAbortUrl ( \$ arg )

the customer will be redirected to this url if he uses the abort link on the payment form, should redirect him back to his cart or to the payment selection page

#### Parameters

*string* \$arg url for aborting the transaction

#### Returns

[SofortLib\\_Multipay](#)

### 6.19.3.11 setAmount ( \$ arg, \$ currency = 'EUR' )

Setter for amount

#### Parameters

*float* \$arg

*string* \$currency

### 6.19.3.12 setNotificationUrl ( \$ arg )

set the url where you want notification about status changes being sent to. Use [SofortLib\\_Notification](#) and [SofortLib\\_TransactionData](#) to further process that notification

#### Parameters

*string* \$arg url

#### Returns

[SofortLib\\_Multipay](#)

### 6.19.3.13 setReason ( \$ arg, \$ arg2 = '' )

Set the reason (Verwendungszweck) for sending money

#### Parameters

*string* \$arg

*string* \$arg2

**6.19.3.14** `setSenderAccountNumber ( $ senderAccountNumber )`

Setter for sender's account number

**Parameters**

*string* \$senderAccountNumber

**6.19.3.15** `setSenderHolder ( $ senderHolder )`

Setter for sender and holder

**Parameters**

*unknown\_type* \$senderHolder

**6.19.3.16** `setSuccessUrl ( $ arg )`

the customer will be redirected to this url after a successful transaction, this should be a page where a short confirmation is displayed

**Parameters**

*string* \$arg the url after a successful transaction

**Returns**

[SofortLib\\_Multipay](#)

**6.19.3.17** `setVersion ( $ arg )`

Setter for interface version

**Parameters**

*string* \$arg

The documentation for this class was generated from the following file:

- `sofortLib_sofortueberweisung_classic.php`

## 6.20 SofortLib\_TransactionData Class Reference

**Public Member Functions**

- [\\_\\_construct](#) (\$configKey= '')

- [setTransaction](#) (\$arg)
- [addTransaction](#) (\$arg)
- [setTime](#) (\$from, \$to)
- [setNumber](#) (\$number, \$page= '1')
- [getConsumerProtection](#) (\$i=0)
- [getInvoiceAddress](#) (\$i=0)
- [getShippingAddress](#) (\$i=0)
- [getStatus](#) (\$i=0)
- [getStatusReason](#) (\$i=0)
- [getStatusModifiedTime](#) (\$i=0)
- [getLanguageCode](#) (\$i=0)
- [getAmount](#) (\$i=0)
- [getOrderNumber](#) (\$i=0)
- [getAmountRefunded](#) (\$i=0)
- [getAmountReceived](#) (\$i=0)
- [getCurrency](#) (\$i=0)
- [getPaymentMethod](#) (\$i=0)
- [getTransaction](#) (\$i=0)
- [getItems](#) (\$i=0)
- [getReason](#) (\$i=0)
- [getUserVariable](#) (\$n, \$i=0)
- [getTime](#) (\$i=0)
- [getProjectId](#) (\$i=0)
- [getInvoiceUrl](#) (\$i=0)
- [getInvoiceStatus](#) (\$i=0)
- [getInvoiceObjection](#) (\$i=0)
- [isTest](#) (\$i=0)
- [isSofortueberweisung](#) (\$i=0)
- [isSofortvorkasse](#) (\$i=0)
- [isSofortlastschrift](#) (\$i=0)
- [isLastschrift](#) (\$i=0)
- [isSofortrechnung](#) (\$i=0)
- [isReceived](#) (\$i=0)
- [isLoss](#) (\$i=0)
- [isPending](#) (\$i=0)
- [isRefunded](#) (\$i=0)
- [getRecipientHolder](#) (\$i=0)
- [getRecipientAccountNumber](#) (\$i=0)
- [getRecipientBankCode](#) (\$i=0)
- [getRecipientCountryCode](#) (\$i=0)
- [getRecipientBankName](#) (\$i=0)
- [getRecipientBic](#) (\$i=0)
- [getRecipientIban](#) (\$i=0)
- [getSenderHolder](#) (\$i=0)
- [getSenderAccountNumber](#) (\$i=0)
- [getSenderBankCode](#) (\$i=0)

- [getSenderCountryCode](#) (\$i=0)
- [getSenderBankName](#) (\$i=0)
- [getSenderBic](#) (\$i=0)
- [getSenderIban](#) (\$i=0)
- [getInvoiceReason](#) (\$n=0, \$i=0)
- [getInvoiceDebitorText](#) (\$i=0)
- [getInvoiceDate](#) (\$i=0)
- [getInvoiceDueDate](#) (\$i=0)
- [getInvoiceNumber](#) (\$i=0)
- [getInvoiceBankHolder](#) (\$i=0)
- [getInvoiceBankAccountNumber](#) (\$i=0)
- [getInvoiceBankCode](#) (\$i=0)
- [getInvoiceBankName](#) (\$i=0)
- [getInvoiceType](#) (\$i=0)
- [getCount](#) ()

## Protected Member Functions

- [\\_parseXml](#) ()

### 6.20.1 Detailed Description

This class is for retrieving information about transactions, you can search by transaction-id or by date

```
eg: $transactionDataObj = new SofortLib_TransactionData('yourapikey');
$transactionDataObj->setTransaction('1234-456-789654-31321')->sendRequest();
echo $transactionDataObj->getStatus();
```

Copyright (c) 2012 SOFORT AG

Released under the GNU General Public License (Version 2)  
[\[http://www.gnu.org/licenses/gpl-2.0.html\]](http://www.gnu.org/licenses/gpl-2.0.html)

#### Date:

2013-04-29 13:22:26 +0200 (Mon, 29 Apr 2013)

#### Version

[SofortLib](#) 1.5.4

#### Id:

sofortLib\_transaction\_data.inc.php 6103 2013-04-29 11:22:26Z dehn

#### Author

SOFORT AG <http://www.sofort.com> ([integration@sofort.com](mailto:integration@sofort.com))



## 6.20.2 Constructor & Destructor Documentation

### 6.20.2.1 \_\_construct ( \$ configKey = " )

Constructor for [SofortLib\\_TransactionData](#)

#### Parameters

*string* \$configKey

## 6.20.3 Member Function Documentation

### 6.20.3.1 \_parseXml ( ) [protected]

Parse the XML (override) (non-PHPdoc)

#### See also

[SofortLib\\_Abstract.\\_parseXml\(\)](#)

### 6.20.3.2 addTransaction ( \$ arg )

use this function if you want to request detailed information about several transactions at once

#### Parameters

*String* \$arg

#### Returns

[SofortLib\\_TransactionData](#) \$this

### 6.20.3.3 getAmount ( \$ i = 0 )

returns the total amount of a transaction

#### Parameters

*int* \$i if you request multiple transactions at once you can set the number here

#### Returns

double amount

**6.20.3.4** `getAmountReceived ( $ i = 0 )`

Getter for the amounts received

**Parameters**

*int* \$i

**6.20.3.5** `getAmountRefunded ( $ i = 0 )`

refund, if a transaction was refundend. amount = amountRefunded if everything was refunded

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

double amount

**6.20.3.6** `getConsumerProtection ( $ i = 0 )`

returns the state of consumer\_protection if set

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean

**6.20.3.7** `getCount ( )`

Getter for count

**6.20.3.8** `getCurrency ( $ i = 0 )`

returns the currency of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string EUR|USD|GBP...

**6.20.3.9   getInvoiceAddress ( \$ i = 0 )**

returns the InvoiceAddress

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

array

**6.20.3.10   getInvoiceBankAccountNumber ( \$ i = 0 )**

invoice receiving bank account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.11   getInvoiceBankCode ( \$ i = 0 )**

invoice receiving bank account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.12   getInvoiceBankHolder ( \$ i = 0 )**

invoice receiving bank account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.13** `getInvoiceBankName ( $ i = 0 )`

invoice receiving bank account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.14** `getInvoiceDate ( $ i = 0 )`

date of the invoice

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.15** `getInvoiceDebitorText ( $ i = 0 )`

get debtor text (Forderungsabtretung)

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.16** `getInvoiceDueDate ( $ i = 0 )`

due date of the invoice, only available for confirmed invoices

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.17 getInvoiceNumber ( \$ i = 0 )**

Getter for invoice number

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string

**6.20.3.18 getInvoiceObjection ( \$ i = 0 )**

returns the status of an invoice

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string the status can be pending|received|reminder\_1|reminder\_2|reminder\_-3|encashment

**6.20.3.19 getInvoiceReason ( \$ n = 0, \$ i = 0 )**

returns the reason the customer needs to use when paying for "Rechnung by sofort"

**Parameters**

*int* \$n specify reason linenummer, can be 1 or 2; us 0 for an array with all reasons

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string|array reason

**6.20.3.20 getInvoiceStatus ( \$ i = 0 )**

returns the status of an invoice

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string the status can be pending|received|reminder\_1|reminder\_2|reminder\_-3|encashment

**6.20.3.21   `getInvoiceType ( $ i = 0 )`**

get the invoice type

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string (OR or LS)

**6.20.3.22   `getInvoiceUrl ( $ i = 0 )`**

you can request the url to the pdf of a sr-invoice with this function

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string url to the pdf

**6.20.3.23   `getItems ( $ i = 0 )`**

Returns an array containing all items of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here @  
return array transactions items

**6.20.3.24   `getLanguageCode ( $ i = 0 )`**

returns the language code of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string language\_code

**6.20.3.25 getOrderNumber ( \$ i = 0 )**

Getter for order number

**Parameters**

*int* \$i

**6.20.3.26 getPaymentMethod ( \$ i = 0 )**

returns the payment method of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string su|sr|sl|sv|ls

**6.20.3.27 getProjectId ( \$ i = 0 )**

returns the project id of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

int project id

**6.20.3.28 getReason ( \$ i = 0 )**

Returns an array containing reason of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here @  
return array transaction reason

**6.20.3.29 getRecipientAccountNumber ( \$ i = 0 )**

returns the account number of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.30 getRecipientBankCode ( \$ i = 0 )**

returns the bank code of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.31 getRecipientBankName ( \$ i = 0 )**

returns the bank name of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.32 getRecipientBic ( \$ i = 0 )**

returns the BIC of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.33 getRecipientCountryCode ( \$ i = 0 )**

returns the country code of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value



**6.20.3.34 getRecipientHolder ( \$ i = 0 )**

returns the holder of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.35 getRecipientIban ( \$ i = 0 )**

returns the IBAN of the receiving account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.36 getSenderAccountNumber ( \$ i = 0 )**

returns the account number of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.37 getSenderBankCode ( \$ i = 0 )**

returns the bank code of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.38** `getSenderBankName ( $ i = 0 )`

returns the bank name of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.39** `getSenderBic ( $ i = 0 )`

returns the BIC of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.40** `getSenderCountryCode ( $ i = 0 )`

returns the country code of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.41** `getSenderHolder ( $ i = 0 )`

returns the holder of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.42 getSenderIban ( \$ i = 0 )**

returns the IBAN of the sending account

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string value

**6.20.3.43 getShippingAddress ( \$ i = 0 )**

returns the ShippingAddress

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

array

**6.20.3.44 getStatus ( \$ i = 0 )**

returns the status of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string pending|received|loss|refunded

**6.20.3.45 getStatusModifiedTime ( \$ i = 0 )**

returns the time of the last status-change so you can check if sth. changed

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string time e.g. 2011-01-01T12:35:09+01:00 use strtotime() to convert it to unix-time

**6.20.3.46 getStatusReason ( \$ i = 0 )**

returns the detailed status description of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string message

**6.20.3.47 getTime ( \$ i = 0 )**

returns the time of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string time e.g. 2011-01-01T12:35:09+01:00 use strtotime() to convert it to unix-time

**6.20.3.48 getTransaction ( \$ i = 0 )**

returns the transaction id of a transaction

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string transaction id

**6.20.3.49 getUserVariable ( \$ n, \$ i = 0 )**

returns the user variable of a transaction

**Parameters**

*int* \$n number of the variable

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

string the content of this variable

**6.20.3.50 isLastschrift ( \$ i = 0 )**

check if the transaction was a lastschrift by sofort

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.51 isLoss ( \$ i = 0 )**

check if status of transaction is loss

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.52 isPending ( \$ i = 0 )**

check if status of transaction is pending

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.53 isReceived ( \$ i = 0 )**

check if status of transaction is received

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.54 isRefunded ( \$ i = 0 )**

check if status of transaction is refunded

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.55 isSofortlastschrift ( \$ i = 0 )**

check if the transaction was a sofortlastschrift

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.56 isSofortrechnung ( \$ i = 0 )**

check if the transaction was a sofortrechnung

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.57 isSofortueberweisung ( \$ i = 0 )**

check if the transaction was a sofortueberweisung

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.58 isSofortvorkasse ( \$ i = 0 )**

check if the transaction was a sofortvorkasse

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

boolean true|false

**6.20.3.59 isTest ( \$ i = 0 )**

checks if the transaction was a test

**Parameters**

*int* \$i if you request multiple transactions at once you can set the number here

**Returns**

bool true|false

**6.20.3.60 setNumber ( \$ number, \$ page = ' 1 ' )**

you can limit the number of results

**Parameters**

*int* \$number number of results [0-100]

*int* \$page result page

**Returns**

[SofortLib\\_TransactionData](#) \$this

**See also**

[setTime\(\)](#)

**6.20.3.61 setTime ( \$ from, \$ to )**

you can request all transactions of a certain time period

use [setNumber\(\)](#) to limit the results

**Parameters**

*string* \$from date possible formats: 2011-01-25 or 2011-01-25T19:01:02+02:00

*string* \$to date possible formats: 2011-01-25 or 2011-01-25T19:01:02+02:00

#### Returns

[SofortLib\\_TransactionData](#) \$this

#### See also

[setNumber\(\)](#)

### 6.20.3.62 setTransaction ( \$ arg )

use this function if you want to request detailed information about a single transaction

#### Parameters

*String* \$arg

#### Returns

[SofortLib\\_TransactionData](#) \$this

The documentation for this class was generated from the following file:

- [sofortLib\\_transaction\\_data.inc.php](#)

## 6.21 SofortTag Class Reference

Inherits [SofortElement](#).

Inherited by [SofortHtmlTag](#).

### Public Member Functions

- [\\_\\_construct](#) (\$tagname, array \$attributes=array(), \$children=array())
- [render](#) ()

### Protected Member Functions

- [\\_render](#) (\$output, \$attributes)

### 6.21.1 Detailed Description

#### Author

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SO-FORT AG Released under the GNU General Public License (Version 2) [<http://www.gnu.org/licenses/gpl-2.0.html>] Implementation of a simple tag



## 6.21.2 Constructor & Destructor Documentation

### 6.21.2.1 `__construct ( $ tagname, array $ attributes = array (), $ children = array () )`

Constructor for [SofortTag](#)

#### Parameters

*string* \$tagname

*array* \$attributes

*array* \$children

Reimplemented in [SofortHtmlTag](#).

## 6.21.3 Member Function Documentation

### 6.21.3.1 `_render ( $ output, $ attributes )` [protected]

Render the output

#### Parameters

*string* \$output

*string* \$attributes

Reimplemented in [SofortHtmlTag](#).

### 6.21.3.2 `render ( )`

Renders the element (override) (non-PHPdoc)

#### See also

[SofortElement.render\(\)](#)

Reimplemented from [SofortElement](#).

The documentation for this class was generated from the following file:

- `sofort_tag.php`

## 6.22 SofortText Class Reference

Inherits [SofortElement](#).

## Public Member Functions

- [\\_\\_construct](#) (\$text, \$escape=false, \$trim=true)
- [render](#) ()

### 6.22.1 Detailed Description

#### Author

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SOFORT AG Released under the GNU General Public License (Version 2) [<http://www.gnu.org/licenses/gpl-2.0.html>] Implementation of simple text

### 6.22.2 Constructor & Destructor Documentation

#### 6.22.2.1 `__construct ( $ text, $ escape = false, $ trim = true )`

Constructor for [SofortText](#)

#### Parameters

*string* \$text  
*boolean* \$escape  
*boolean* \$trim

### 6.22.3 Member Function Documentation

#### 6.22.3.1 `render ( )`

Renders the element (override) (non-PHPdoc)

#### See also

[SofortElement.render\(\)](#)

Reimplemented from [SofortElement](#).

The documentation for this class was generated from the following file:

- `sofort_text.php`

## 6.23 XmlToArray Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$input, \$maxDepth=20)
- [log](#) (\$msg, \$type=2)
- [toArray](#) (\$simpleStructure=false)

## Static Public Member Functions

- static [render](#) (\$input, \$simpleStructure=false, \$maxDepth=20)

### 6.23.1 Detailed Description

#### Author

SOFORT AG ([integration@sofort.com](mailto:integration@sofort.com)) Copyright (c) 2012 SOFORT AG Released under the GNU General Public License (Version 2) [<http://www.gnu.org/licenses/gpl-2.0.html>] XML To Array conversion

### 6.23.2 Constructor & Destructor Documentation

#### 6.23.2.1 `__construct ( $ input, $ maxDepth = 20 )`

Loads XML into array representation.

#### Parameters

*string* \$input  
*int* \$maxDepth

#### Exceptions

[\*XmlToArrayException\*](#)

### 6.23.3 Member Function Documentation

#### 6.23.3.1 `log ( $ msg, $ type = 2 )`

Log messages (debugging purpose)

#### Parameters

*string* \$msg  
*int* \$type

#### 6.23.3.2 `static render ( $ input, $ simpleStructure = false, $ maxDepth = 20 ) [static]`

Static entry point

#### Parameters

*string* \$input  
*bool* \$simpleStructure

*int* \$maxDepth only parse XML to the provided depth

**Returns**

array

**Exceptions**

[\*XmlToArrayException\*](#)

**6.23.3.3 toArray ( \$ simpleStructure = false )**

Returns parsed XML as array structure

**Returns**

array

The documentation for this class was generated from the following file:

- xml\_to\_array.php

## 6.24 XmlToArrayException Class Reference

### 6.24.1 Detailed Description

XML to Array Exception

The documentation for this class was generated from the following file:

- xml\_to\_array\_exception.php

## 6.25 XmlToArrayNode Class Reference

### Public Member Functions

- [\\_\\_construct](#) (\$name, \$attributes)
- [addChild](#) (XmlToArrayNode \$XmlToArrayNode)
- [getData](#) ()
- [getName](#) ()
- [getParentXmlToArrayNode](#) ()
- [hasChildren](#) ()
- [hasParentXmlToArrayNode](#) ()
- [isOpen](#) ()
- [render](#) (\$simpleStructure)
- [setClosed](#) ()
- [setData](#) (\$data)
- [setParentXmlToArrayNode](#) (XmlToArrayNode \$XmlToArrayNode)

### 6.25.1 Detailed Description

XML To Array Node

### 6.25.2 Constructor & Destructor Documentation

#### 6.25.2.1 `__construct ( $ name, $ attributes )`

Constructor for [XmlToArrayNode](#)

##### Parameters

*string* \$name

*array* \$attributes

### 6.25.3 Member Function Documentation

#### 6.25.3.1 `addChild ( XmlToArrayNode $ XmlToArrayNode )`

Add a child to collection

##### Parameters

[XmlToArrayNode](#) \$XmlToArrayNode

#### 6.25.3.2 `getData ( )`

Getter for data, returns an array

#### 6.25.3.3 `getName ( )`

Getter for name, returns the name

#### 6.25.3.4 `getParentXmlToArrayNode ( )`

Getter for parent node

#### 6.25.3.5 `hasChildren ( )`

Does it have any children

#### 6.25.3.6 `hasParentXmlToArrayNode ( )`

Does it have a node

**6.25.3.7 isOpen ( )**

Is it open, returns `_open`

**6.25.3.8 render ( \$ *simpleStructure* )**

Renders nodes as array

**Parameters**

*bool* \$simpleStructure pass true to get an array without and fields

**Exceptions**

[\*XmlToArrayException\*](#)

**6.25.3.9 setClosed ( )**

Set it to closed

**6.25.3.10 setData ( \$ *data* )**

Setter for variable data

**Parameters**

*string* \$data

**6.25.3.11 setParentXmlNode ( XmlNode \$ *XmlNode* )**

Setter for parent node

**Parameters**

[\*XmlNode\*](#) \$XmlNode

The documentation for this class was generated from the following file:

- `xml_to_array_node.php`

# Index

- \_\_construct
  - [ArrayToXml](#), [13](#)
  - [PnagArticle](#), [15](#)
  - [PnagInvoice](#), [21](#)
  - [SofortHtmlTag](#), [40](#)
  - [SofortLib](#), [42](#)
  - [SofortLib\\_CancelSr](#), [48](#)
  - [SofortLib\\_ClassicNotification](#), [50](#)
  - [SofortLib\\_ConfirmSr](#), [54](#)
  - [SofortLib\\_Debit](#), [58](#)
  - [SofortLib\\_EditSr](#), [63](#)
  - [SofortLib\\_iDeal\\_Banks](#), [66](#)
  - [SofortLib\\_iDealClassic](#), [67](#)
  - [SofortLib\\_Multipay](#), [71](#)
  - [SofortLib\\_Notification](#), [85](#)
  - [SofortLib\\_Refund](#), [87](#)
  - [SofortLib\\_-  
SofortueberweisungClassic](#), [91](#)
  - [SofortLib\\_TransactionData](#), [97](#)
  - [SofortTag](#), [113](#)
  - [SofortText](#), [114](#)
  - [XmlToArray](#), [115](#)
  - [XmlToArrayNode](#), [117](#)
- \_\_toString
  - [PnagInvoice](#), [21](#)
- \_getErrorBlock
  - [SofortLib](#), [42](#)
- \_getHashHexValue
  - [SofortLib\\_ClassicNotification](#), [51](#)
- \_getPaymentDomain
  - [SofortLib\\_iDealClassic](#), [68](#)
  - [SofortLib\\_-  
SofortueberweisungClassic](#), [91](#)
- \_handleErrors
  - [SofortLib](#), [42](#)
  - [SofortLib\\_CancelSr](#), [48](#)
  - [SofortLib\\_ConfirmSr](#), [54](#)
  - [SofortLib\\_EditSr](#), [63](#)
  - [SofortLib\\_Refund](#), [87](#)
- \_parseXml
  - [SofortLib](#), [42](#)
  - [SofortLib\\_CancelSr](#), [48](#)
  - [SofortLib\\_ConfirmSr](#), [54](#)
  - [SofortLib\\_EditSr](#), [63](#)
  - [SofortLib\\_iDeal\\_Banks](#), [66](#)
  - [SofortLib\\_Multipay](#), [71](#)
  - [SofortLib\\_Refund](#), [87](#)
  - [SofortLib\\_TransactionData](#), [97](#)
- \_render
  - [SofortHtmlTag](#), [40](#)
  - [SofortTag](#), [113](#)
- \_sendMessage
  - [SofortLib](#), [42](#)
- addChild
  - [XmlToArrayNode](#), [117](#)
- addInvoiceAddress
  - [PnagInvoice](#), [21](#)
- addItem
  - [SofortLib\\_ConfirmSr](#), [54](#)
  - [SofortLib\\_EditSr](#), [63](#)
- addItemToInvoice
  - [PnagInvoice](#), [22](#)
- addNotificationEmail
  - [SofortLib\\_Multipay](#), [71](#)
- addNotificationUrl
  - [SofortLib\\_Multipay](#), [72](#)
- addReason
  - [SofortLib\\_Debit](#), [58](#)
- addRefund
  - [SofortLib\\_Refund](#), [87](#)
- addShippingAddress
  - [PnagInvoice](#), [22](#)
- addShippingAddresss
  - [PnagInvoice](#), [23](#)
- addSofortrechnungItem
  - [SofortLib\\_Multipay](#), [72](#)
- addTransaction
  - [SofortLib\\_TransactionData](#), [97](#)
- addUserVariable

- PnagInvoice, [23](#)
  - SofortLib\_Debit, [58](#)
  - SofortLib\_Multipay, [72](#)
  - SofortLib\_-
    - SofortueberweisungClassic, [91](#)
- ArrayToXml, [13](#)
  - \_\_construct, [13](#)
  - render, [14](#)
  - toXml, [14](#)
- ArrayToXmlException, [14](#)
- cancelInvoice
  - PnagInvoice, [24](#)
  - SofortLib\_CancelSr, [48](#)
  - SofortLib\_ConfirmSr, [54](#)
- checkout
  - PnagInvoice, [24](#)
- confirmInvoice
  - PnagInvoice, [24](#)
  - SofortLib\_ConfirmSr, [55](#)
- debug
  - SofortLib, [42](#)
- deleteAllErrors
  - SofortLib, [42](#)
- deleteAllWarnings
  - SofortLib, [43](#)
- disableLog
  - PnagInvoice, [24](#)
  - SofortLib, [43](#)
- enableLog
  - PnagInvoice, [24](#)
  - SofortLib, [43](#)
- error
  - SofortLib, [43](#)
- fatalError
  - SofortLib, [43](#)
- generatePassword
  - SofortLib\_-
    - SofortueberweisungClassic, [91](#)
- getAmount
  - PnagInvoice, [25](#)
  - SofortLib\_ClassicNotification, [51](#)
  - SofortLib\_Debit, [58](#)
  - SofortLib\_Refund, [87](#)
  - SofortLib\_TransactionData, [97](#)
- getAmountReceived
  - SofortLib\_TransactionData, [97](#)
- getAmountRefunded
  - PnagInvoice, [25](#)
  - SofortLib\_TransactionData, [98](#)
- getAsArray
  - SofortLib\_Refund, [87](#)
- getBanks
  - SofortLib\_iDeal\_Banks, [66](#)
- getComment
  - SofortLib\_Refund, [88](#)
- getConstantById
  - PnagInvoice, [25](#)
- getConstantByName
  - PnagInvoice, [25](#)
- getConsumerProtection
  - SofortLib\_TransactionData, [98](#)
- getCount
  - SofortLib\_TransactionData, [98](#)
- getCurrency
  - SofortLib\_ClassicNotification, [51](#)
  - SofortLib\_TransactionData, [98](#)
- getCustomerNumber
  - PnagInvoice, [25](#)
- getData
  - XmlToArrayNode, [117](#)
- getDate
  - SofortLib\_Debit, [59](#)
- getDescription
  - PnagArticle, [16](#)
- getDta
  - SofortLib\_Refund, [88](#)
- getError
  - PnagInvoice, [25](#)
  - SofortLib, [43](#)
  - SofortLib\_ClassicNotification, [51](#)
  - SofortLib\_Debit, [59](#)
  - SofortLib\_iDealClassic, [68](#)
  - SofortLib\_-
    - SofortueberweisungClassic, [92](#)
- getErrorCodes
  - PnagInvoice, [26](#)
- getErrors
  - PnagInvoice, [26](#)
  - SofortLib, [44](#)
- getHashHexValue
  - SofortLib\_-
    - SofortueberweisungClassic, [92](#)



- getHttpResponseCode
  - PnagInvoice, 26
- getInvoice
  - PnagInvoice, 26
- getInvoiceAddress
  - SofortLib\_TransactionData, 98
- getInvoiceBankAccountNumber
  - SofortLib\_TransactionData, 99
- getInvoiceBankCode
  - SofortLib\_TransactionData, 99
- getInvoiceBankHolder
  - SofortLib\_TransactionData, 99
- getInvoiceBankName
  - SofortLib\_TransactionData, 99
- getInvoiceDate
  - SofortLib\_TransactionData, 100
- getInvoiceDebitorText
  - SofortLib\_TransactionData, 100
- getInvoiceDownloadMethod
  - PnagInvoice, 26
- getInvoiceDueDate
  - SofortLib\_TransactionData, 100
- getInvoiceNumber
  - PnagInvoice, 26
  - SofortLib\_TransactionData, 100
- getInvoiceObjection
  - PnagInvoice, 26
  - SofortLib\_TransactionData, 101
- getInvoiceReason
  - SofortLib\_TransactionData, 101
- getInvoiceStatus
  - PnagInvoice, 27
  - SofortLib\_TransactionData, 101
- getInvoiceTye
  - PnagInvoice, 27
- getInvoiceType
  - SofortLib\_TransactionData, 101
- getInvoiceUrl
  - PnagInvoice, 27
  - SofortLib\_CancelSr, 48
  - SofortLib\_ConfirmSr, 55
  - SofortLib\_EditSr, 64
  - SofortLib\_TransactionData, 102
- getItemAmount
  - PnagInvoice, 27
- getItemId
  - PnagArticle, 16
- getItems
  - PnagInvoice, 27
  - SofortLib\_TransactionData, 102
- getLanguageCode
  - PnagInvoice, 28
  - SofortLib\_TransactionData, 102
- getName
  - XmlToArrayNode, 117
- getNotification
  - SofortLib\_ClassicNotification, 51
  - SofortLib\_Notification, 85
- getOrderNumber
  - PnagInvoice, 28
  - SofortLib\_TransactionData, 102
- getParentXmlToArrayNode
  - XmlToArrayNode, 117
- getPaymentMethod
  - PnagInvoice, 28
  - SofortLib\_Multipay, 73
  - SofortLib\_TransactionData, 103
- getPaymentUrl
  - PnagInvoice, 28
  - SofortLib\_Multipay, 73
  - SofortLib\_-
    - SofortueberweisungClassic, 92
- getProductNumber
  - PnagArticle, 16
- getProjectId
  - SofortLib\_TransactionData, 103
- getQuantity
  - PnagArticle, 16
- getReason
  - SofortLib\_Debit, 59
  - SofortLib\_TransactionData, 103
- getRecipientAccountNumber
  - SofortLib\_TransactionData, 103
- getRecipientBankCode
  - SofortLib\_TransactionData, 104
- getRecipientBankName
  - SofortLib\_TransactionData, 104
- getRecipientBic
  - SofortLib\_TransactionData, 104
- getRecipientCountryCode
  - SofortLib\_TransactionData, 104
- getRecipientHolder
  - SofortLib\_TransactionData, 104
- getRecipientIban
  - SofortLib\_TransactionData, 105
- getRefundError
  - SofortLib\_Refund, 88
- getRelatedBanks
  - SofortLib\_iDealClassic, 68

- getResponse
  - SofortLib\_Debit, 59
- getSenderAccountNumber
  - SofortLib\_TransactionData, 105
- getSenderBankCode
  - SofortLib\_TransactionData, 105
- getSenderBankName
  - SofortLib\_TransactionData, 105
- getSenderBic
  - SofortLib\_TransactionData, 106
- getSenderCountryCode
  - SofortLib\_TransactionData, 106
- getSenderHolder
  - SofortLib\_TransactionData, 106
- getSenderIban
  - SofortLib\_TransactionData, 106
- getShippingAddress
  - SofortLib\_TransactionData, 107
- getSofortrechnungItem
  - SofortLib\_Multipay, 73
- getSofortrechnungItemAmount
  - SofortLib\_Multipay, 73
- getSofortrechnungItems
  - SofortLib\_Multipay, 73
- getState
  - PnagInvoice, 28
- getStatus
  - PnagInvoice, 28
  - SofortLib\_ClassicNotification, 51
  - SofortLib\_Refund, 88
  - SofortLib\_TransactionData, 107
- getStatusArray
  - SofortLib\_Refund, 88
- getStatusModifiedTime
  - SofortLib\_TransactionData, 107
- getStatusOfInvoice
  - PnagInvoice, 29
- getStatusReason
  - PnagInvoice, 29
  - SofortLib\_ClassicNotification, 52
  - SofortLib\_TransactionData, 107
- getSupportedHashAlgorithm
  - SofortLib\_-
    - SofortueberweisungClassic, 92
- getSupportedHashAlgorithms
  - SofortLib\_-
    - SofortueberweisungClassic, 92
- getTax
  - PnagArticle, 16
- getTime
  - SofortLib\_ClassicNotification, 52
  - SofortLib\_Notification, 85
  - SofortLib\_TransactionData, 108
- getTitle
  - PnagArticle, 16
  - SofortLib\_Refund, 88
- getTransaction
  - SofortLib\_ClassicNotification, 52
  - SofortLib\_TransactionData, 108
- getTransactionData
  - PnagInvoice, 29
- getTransactionId
  - PnagInvoice, 29
  - SofortLib\_Debit, 59
  - SofortLib\_Multipay, 74
  - SofortLib\_Notification, 85
  - SofortLib\_Refund, 88
- getTransactionInfo
  - PnagInvoice, 29
- getUnitPrice
  - PnagArticle, 16
- getUserVariable
  - SofortLib\_ClassicNotification, 52
  - SofortLib\_Debit, 59
  - SofortLib\_TransactionData, 108
- getWarnings
  - PnagInvoice, 30
  - SofortLib, 44
- handleDownload
  - PnagInvoice, 30
- handleErrors
  - SofortLib, 44
- hasChildren
  - XmlToArrayNode, 117
- hasParentXmlToArrayNode
  - XmlToArrayNode, 117
- isConsumerProtection
  - SofortLib\_Multipay, 74
- isDebitpayCheckDisabled
  - SofortLib\_Multipay, 74
- isError
  - PnagInvoice, 30
  - SofortLib, 44
  - SofortLib\_ClassicNotification, 52
  - SofortLib\_Debit, 59

- SofortLib\_
  - SofortueberweisungClassic,  
92
- isLastschrift
  - SofortLib\_Multipay, 74
  - SofortLib\_TransactionData, 108
- isLoss
  - SofortLib\_TransactionData, 109
- isOpen
  - XmlToArrayNode, 117
- isPending
  - SofortLib\_TransactionData, 109
- isReceived
  - SofortLib\_TransactionData, 109
- isRefunded
  - SofortLib\_TransactionData, 109
- isRefundError
  - SofortLib\_Refund, 88
- isSofortlastschrift
  - SofortLib\_Multipay, 74
  - SofortLib\_TransactionData, 110
- isSofortrechnung
  - SofortLib\_Multipay, 74
  - SofortLib\_TransactionData, 110
- isSofortueberweisung
  - SofortLib\_Multipay, 74
  - SofortLib\_TransactionData, 110
- isSofortvorkasse
  - SofortLib\_Multipay, 74
  - SofortLib\_TransactionData, 110
- isTest
  - SofortLib\_TransactionData, 111
- isWarning
  - PnagInvoice, 30
  - SofortLib, 45
- log
  - PnagInvoice, 30
  - SofortLib, 45
  - XmlToArray, 115
- logError
  - PnagInvoice, 30
  - SofortLib, 45
- logWarning
  - PnagInvoice, 31
  - SofortLib, 45
- PnagArticle, 15
  - \_\_construct, 15
  - getDescription, 16
  - getItemId, 16
  - getProductNumber, 16
  - getQuantity, 16
  - getTax, 16
  - getTitle, 16
  - getUnitPrice, 16
  - setDescription, 16
  - setProductNumber, 17
  - setQuantity, 17
  - setTax, 17
  - setUnitPrice, 17
- PnagCustomer, 18
  - PnagCustomer, 18
- PnagInvoice, 19
  - \_\_construct, 21
  - \_\_toString, 21
  - addInvoiceAddress, 21
  - addItemToInvoice, 22
  - addShippingAddress, 22
  - addShippingAddresss, 23
  - addUserVariable, 23
  - cancelInvoice, 24
  - checkout, 24
  - confirmInvoice, 24
  - disableLog, 24
  - enableLog, 24
  - getAmount, 25
  - getAmountRefunded, 25
  - getConstantById, 25
  - getConstantByName, 25
  - getCustomerNumber, 25
  - getError, 25
  - getErrorCodes, 26
  - getErrors, 26
  - getHttpStatusCode, 26
  - getInvoice, 26
  - getInvoiceDownloadMethod, 26
  - getInvoiceNumber, 26
  - getInvoiceObjection, 26
  - getInvoiceStatus, 27
  - getInvoiceTye, 27
  - getInvoiceUrl, 27
  - getItemAmount, 27
  - getItems, 27
  - getLanguageCode, 28
  - getOrderNumber, 28
  - getPaymentMethod, 28
  - getPaymentUrl, 28
  - getState, 28
  - getStatus, 28

- [getStatusOfInvoice](#), 29
- [getStatusReason](#), 29
- [getTransactionData](#), 29
- [getTransactionId](#), 29
- [getTransactionInfo](#), 29
- [getWarnings](#), 30
- [handleDownload](#), 30
- [isError](#), 30
- [isWarning](#), 30
- [log](#), 30
- [logError](#), 30
- [logWarning](#), 31
- [refreshTransactionData](#), 31
- [removeItemfromInvoice](#), 31
- [setAbortUrl](#), 31
- [setAmount](#), 32
- [setAmountRefunded](#), 32
- [setBitmask](#), 32
- [setCustomerId](#), 32
- [setDebitorVatNumber](#), 32
- [setEmailCustomer](#), 33
- [setInvoiceObjection](#), 33
- [setInvoiceStatus](#), 33
- [setInvoiceUrl](#), 33
- [setItems](#), 33
- [setLanguageCode](#), 34
- [setNotificationUrl](#), 34
- [setOrderId](#), 34
- [setPaymentMethod](#), 34
- [setPhoneNumberCustomer](#), 34
- [setReason](#), 35
- [setSofortLibCancelSr](#), 35
- [setSofortLibEditSr](#), 35
- [setSofortLibMultipay](#), 35
- [setSofortLibTransactionData](#), 35
- [setSofortrechnung](#), 35
- [setState](#), 36
- [setStatus](#), 36
- [setStatusOfInvoice](#), 36
- [setStatusReason](#), 36
- [setSuccessUrl](#), 36
- [setTime](#), 37
- [setTimeout](#), 37
- [setTimeoutUrl](#), 37
- [setTransaction](#), 37
- [setTransactionId](#), 37
- [setVersion](#), 38
- [updateInvoice](#), 38
- [updateInvoiceItem](#), 38
- [updateOrderNumber](#), 38
- [refreshTransactionData](#)
  - [PnagInvoice](#), 31
- [removeItem](#)
  - [SofortLib\\_ConfirmSr](#), 55
- [removeItemfromInvoice](#)
  - [PnagInvoice](#), 31
- [removeSofortrechnungItem](#)
  - [SofortLib\\_Multipay](#), 75
- [render](#)
  - [ArrayToXml](#), 14
  - [SofortElement](#), 39
  - [SofortTag](#), 113
  - [SofortText](#), 114
  - [XmlToArray](#), 115
  - [XmlToArrayNode](#), 118
- [sendRequest](#)
  - [SofortLib\\_Debit](#), 60
  - [SofortLib\\_Notification](#), 85
  - [SofortLib\\_Refund](#), 89
- [sendValidationRequest](#)
  - [SofortLib\\_Multipay](#), 75
- [setAbortUrl](#)
  - [PnagInvoice](#), 31
  - [SofortLib\\_Multipay](#), 75
  - [SofortLib\\_-](#)
    - [SofortueberweisungClassic](#), 92
- [setAmount](#)
  - [PnagInvoice](#), 32
  - [SofortLib\\_Debit](#), 60
  - [SofortLib\\_Multipay](#), 75
  - [SofortLib\\_-](#)
    - [SofortueberweisungClassic](#), 93
- [setAmountRefunded](#)
  - [PnagInvoice](#), 32
- [setApiVersion](#)
  - [SofortLib](#), 46
- [setBitmask](#)
  - [PnagInvoice](#), 32
- [setClosed](#)
  - [XmlToArrayNode](#), 118
- [setComment](#)
  - [SofortLib\\_CancelSr](#), 49
  - [SofortLib\\_ConfirmSr](#), 55
  - [SofortLib\\_EditSr](#), 64
- [setCreditNoteNumber](#)
  - [SofortLib\\_CancelSr](#), 49
- [setCustomerId](#)

- PnagInvoice, 32
- setCustomerNumber
  - SofortLib\_ConfirmSr, 56
  - SofortLib\_EditSr, 64
- setData
  - XmlToArrayNode, 118
- setDate
  - SofortLib\_Debit, 60
- setDebitorVatNumber
  - PnagInvoice, 32
  - SofortLib\_Multipay, 75
- setDescription
  - PnagArticle, 16
- setEmailCustomer
  - PnagInvoice, 33
  - SofortLib\_Multipay, 76
- setError
  - SofortLib, 46
- setInvoiceNumber
  - SofortLib\_ConfirmSr, 56
  - SofortLib\_EditSr, 64
- setInvoiceObjection
  - PnagInvoice, 33
- setInvoiceStatus
  - PnagInvoice, 33
- setInvoiceUrl
  - PnagInvoice, 33
- setItems
  - PnagInvoice, 33
- setLanguageCode
  - PnagInvoice, 34
  - SofortLib\_Multipay, 76
- setLastschrift
  - SofortLib\_Multipay, 76
- setLastschriftAddress
  - SofortLib\_Multipay, 76
- setLastschriftBaseCheckDisabled
  - SofortLib\_Multipay, 77
- setLastschriftExtendedCheckDisabled
  - SofortLib\_Multipay, 77
- setLastschriftMobileCheckDisabled
  - SofortLib\_Multipay, 77
- setLogDisabled
  - SofortLib, 46
- setLogEnabled
  - SofortLib, 46
- setLogfilePath
  - SofortLib, 46
- setLogger
  - SofortLib, 46
- setNotificationEmail
  - SofortLib\_Multipay, 77
- setNotificationUrl
  - PnagInvoice, 34
  - SofortLib\_Multipay, 77
  - SofortLib\_-
    - SofortueberweisungClassic, 93
- setNumber
  - SofortLib\_TransactionData, 111
- setOrderId
  - PnagInvoice, 34
- setOrderNumber
  - SofortLib\_ConfirmSr, 56
  - SofortLib\_EditSr, 64
- setParentXmlToArrayNode
  - XmlToArrayNode, 118
- setPaymentMethod
  - PnagInvoice, 34
- setPhoneNumberCustomer
  - PnagInvoice, 34
  - SofortLib\_Multipay, 78
- setProductNumber
  - PnagArticle, 17
- setProjectId
  - SofortLib\_Debit, 60
- setQuantity
  - PnagArticle, 17
- setReason
  - PnagInvoice, 35
  - SofortLib\_Debit, 60
  - SofortLib\_Multipay, 78
  - SofortLib\_-
    - SofortueberweisungClassic, 93
- setSenderAccount
  - SofortLib\_Debit, 61
  - SofortLib\_Multipay, 78
  - SofortLib\_Refund, 89
- setSenderAccountNumber
  - SofortLib\_Debit, 61
  - SofortLib\_-
    - SofortueberweisungClassic, 93
- setSenderBankCode
  - SofortLib\_Debit, 61
  - SofortLib\_iDealClassic, 68
- setSenderCountryId
  - SofortLib\_iDealClassic, 68
- setSenderHolder

- SofortLib\_Debit, [61](#)
- SofortLib\_-
  - SofortueberweisungClassic, [94](#)
- setSofortlastschrift
  - SofortLib\_Multipay, [78](#)
- setSofortlastschriftAddress
  - SofortLib\_Multipay, [79](#)
- setSofortLibCancelSr
  - PnagInvoice, [35](#)
- setSofortLibEditSr
  - PnagInvoice, [35](#)
- setSofortLibMultipay
  - PnagInvoice, [35](#)
- setSofortLibTransactionData
  - PnagInvoice, [35](#)
- setSofortrechnung
  - PnagInvoice, [35](#)
  - SofortLib\_Multipay, [79](#)
- setSofortrechnungCustomerId
  - SofortLib\_Multipay, [79](#)
- setSofortrechnungInvoiceAddress
  - SofortLib\_Multipay, [80](#)
- setSofortrechnungOrderId
  - SofortLib\_Multipay, [80](#)
- setSofortrechnungShippingAddress
  - SofortLib\_Multipay, [80](#)
- setSofortrechnungTimeForPayment
  - SofortLib\_Multipay, [81](#)
- setSofortrechnungComment
  - SofortLib\_Multipay, [81](#)
- setSofortueberweisung
  - SofortLib\_Multipay, [81](#)
- setSofortueberweisungCustomerprotection
  - SofortLib\_Multipay, [81](#)
- setSofortvorkasse
  - SofortLib\_Multipay, [82](#)
- setSofortvorkasseCustomerprotection
  - SofortLib\_Multipay, [82](#)
- setState
  - PnagInvoice, [36](#)
- setStatus
  - PnagInvoice, [36](#)
- setStatusOfInvoice
  - PnagInvoice, [36](#)
- setStatusReason
  - PnagInvoice, [36](#)
- setSuccessLinkRedirect
  - SofortLib\_Multipay, [82](#)
- setSuccessUrl
  - PnagInvoice, [36](#)
  - SofortLib\_Multipay, [82](#)
  - SofortLib\_-
    - SofortueberweisungClassic, [94](#)
- setTax
  - PnagArticle, [17](#)
- setTime
  - PnagInvoice, [37](#)
  - SofortLib\_TransactionData, [111](#)
- setTimeout
  - PnagInvoice, [37](#)
  - SofortLib\_Multipay, [83](#)
- setTimeoutUrl
  - PnagInvoice, [37](#)
  - SofortLib\_Multipay, [83](#)
- setTitle
  - SofortLib\_Refund, [89](#)
- setTransaction
  - PnagInvoice, [37](#)
  - SofortLib\_CancelSr, [49](#)
  - SofortLib\_ConfirmSr, [56](#)
  - SofortLib\_EditSr, [64](#)
  - SofortLib\_TransactionData, [112](#)
- setTransactionId
  - PnagInvoice, [37](#)
- setUnitPrice
  - PnagArticle, [17](#)
- setVersion
  - PnagInvoice, [38](#)
  - SofortLib\_Multipay, [83](#)
  - SofortLib\_-
    - SofortueberweisungClassic, [94](#)
- SofortElement, [39](#)
  - render, [39](#)
- SofortHtmlTag, [39](#)
  - \_\_construct, [40](#)
  - \_render, [40](#)
- SofortLib, [40](#)
  - \_\_construct, [42](#)
  - \_getErrorBlock, [42](#)
  - \_handleErrors, [42](#)
  - \_parseXml, [42](#)
  - \_sendMessage, [42](#)
  - debug, [42](#)
  - deleteAllErrors, [42](#)
  - deleteAllWarnings, [43](#)
  - disableLog, [43](#)
  - enableLog, [43](#)

- error, [43](#)
- fatalError, [43](#)
- getError, [43](#)
- getErrors, [44](#)
- getWarnings, [44](#)
- handleErrors, [44](#)
- isError, [44](#)
- isWarning, [45](#)
- log, [45](#)
- logError, [45](#)
- logWarning, [45](#)
- setApiVersion, [46](#)
- setError, [46](#)
- setLogDisabled, [46](#)
- setLogEnabled, [46](#)
- setLogfilePath, [46](#)
- setLogger, [46](#)
- sofortLib, [11](#)
- SofortLib\_CancelSr, [47](#)
  - \_\_construct, [48](#)
  - \_handleErrors, [48](#)
  - \_parseXml, [48](#)
  - cancelInvoice, [48](#)
  - getInvoiceUrl, [48](#)
  - setComment, [49](#)
  - setCreditNoteNumber, [49](#)
  - setTransaction, [49](#)
- SofortLib\_ClassicNotification, [49](#)
  - \_\_construct, [50](#)
  - \_getHashHexValue, [51](#)
  - getAmount, [51](#)
  - getCurrency, [51](#)
  - getError, [51](#)
  - getNotification, [51](#)
  - getStatus, [51](#)
  - getStatusReason, [52](#)
  - getTime, [52](#)
  - getTransaction, [52](#)
  - getUserVariable, [52](#)
  - isError, [52](#)
- SofortLib\_ConfirmSr, [53](#)
  - \_\_construct, [54](#)
  - \_handleErrors, [54](#)
  - \_parseXml, [54](#)
  - addItem, [54](#)
  - cancelInvoice, [54](#)
  - confirmInvoice, [55](#)
  - getInvoiceUrl, [55](#)
  - removeItem, [55](#)
  - setComment, [55](#)
  - setCustomerNumber, [56](#)
  - setInvoiceNumber, [56](#)
  - setOrderNumber, [56](#)
  - setTransaction, [56](#)
  - updateCart, [56](#)
- SofortLib\_Debit, [57](#)
  - \_\_construct, [58](#)
  - addReason, [58](#)
  - addUserVariable, [58](#)
  - getAmount, [58](#)
  - getDate, [59](#)
  - getError, [59](#)
  - getReason, [59](#)
  - getResponse, [59](#)
  - getTransactionId, [59](#)
  - getUserVariable, [59](#)
  - isError, [59](#)
  - sendRequest, [60](#)
  - setAmount, [60](#)
  - setDate, [60](#)
  - setProjectId, [60](#)
  - setReason, [60](#)
  - setSenderAccount, [61](#)
  - setSenderAccountNumber, [61](#)
  - setSenderBankCode, [61](#)
  - setSenderHolder, [61](#)
- SofortLib\_EditSr, [62](#)
  - \_\_construct, [63](#)
  - \_handleErrors, [63](#)
  - \_parseXml, [63](#)
  - addItem, [63](#)
  - getInvoiceUrl, [64](#)
  - setComment, [64](#)
  - setCustomerNumber, [64](#)
  - setInvoiceNumber, [64](#)
  - setOrderNumber, [64](#)
  - setTransaction, [64](#)
  - updateCart, [65](#)
- SofortLib\_iDeal\_Banks, [65](#)
  - \_\_construct, [66](#)
  - \_parseXml, [66](#)
  - getBanks, [66](#)
- SofortLib\_iDealClassic, [66](#)
  - \_\_construct, [67](#)
  - \_getPaymentDomain, [68](#)
  - getError, [68](#)
  - getRelatedBanks, [68](#)
  - setSenderBankCode, [68](#)
  - setSenderCountryId, [68](#)
- SofortLib\_Multipay, [69](#)

- [\\_\\_construct, 71](#)
- [\\_parseXml, 71](#)
- [addNotificationEmail, 71](#)
- [addNotificationUrl, 72](#)
- [addSofortrechnungItem, 72](#)
- [addUserVariable, 72](#)
- [getPaymentMethod, 73](#)
- [getPaymentUrl, 73](#)
- [getSofortrechnungItem, 73](#)
- [getSofortrechnungItemAmount, 73](#)
- [getSofortrechnungItems, 73](#)
- [getTransactionId, 74](#)
- [isConsumerProtection, 74](#)
- [isDebitpayCheckDisabled, 74](#)
- [isLastschrift, 74](#)
- [isSofortlastschrift, 74](#)
- [isSofortrechnung, 74](#)
- [isSofortueberweisung, 74](#)
- [isSofortvorkasse, 74](#)
- [removeSofortrechnungItem, 75](#)
- [sendValidationRequest, 75](#)
- [setAbortUrl, 75](#)
- [setAmount, 75](#)
- [setDebitorVatNumber, 75](#)
- [setEmailCustomer, 76](#)
- [setLanguageCode, 76](#)
- [setLastschrift, 76](#)
- [setLastschriftAddress, 76](#)
- [setLastschriftBaseCheckDisabled, 77](#)
- [setLastschriftExtendedCheckDisabled, 77](#)
- [setLastschriftMobileCheckDisabled, 77](#)
- [setNotificationEmail, 77](#)
- [setNotificationUrl, 77](#)
- [setPhoneNumberCustomer, 78](#)
- [setReason, 78](#)
- [setSenderAccount, 78](#)
- [setSofortlastschrift, 78](#)
- [setSofortlastschriftAddress, 79](#)
- [setSofortrechnung, 79](#)
- [setSofortrechnungCustomerId, 79](#)
- [setSofortrechnungInvoiceAddress, 80](#)
- [setSofortrechnungOrderId, 80](#)
- [setSofortrechnungShippingAddress, 80](#)
- [setSofortrechnungTimeForPayment, 81](#)
- [setSofortrechnungComment, 81](#)
- [setSofortueberweisung, 81](#)
- [setSofortueberweisungCustomerprotection, 81](#)
- [setSofortvorkasse, 82](#)
- [setSofortvorkasseCustomerprotection, 82](#)
- [setSuccessLinkRedirect, 82](#)
- [setSuccessUrl, 82](#)
- [setTimeout, 83](#)
- [setTimeoutUrl, 83](#)
- [setVersion, 83](#)
- [updateSofortrechnungItem, 83](#)
- [SofortLib\\_Notification, 84](#)
- [\\_\\_construct, 85](#)
- [getNotification, 85](#)
- [getTime, 85](#)
- [getTransactionId, 85](#)
- [sendRequest, 85](#)
- [SofortLib\\_Refund, 86](#)
- [\\_\\_construct, 87](#)
- [\\_handleErrors, 87](#)
- [\\_parseXml, 87](#)
- [addRefund, 87](#)
- [getAmount, 87](#)
- [getAsArray, 87](#)
- [getComment, 88](#)
- [getDta, 88](#)
- [getRefundError, 88](#)
- [getStatus, 88](#)
- [getStatusArray, 88](#)
- [getTitle, 88](#)
- [getTransactionId, 88](#)
- [isRefundError, 88](#)
- [sendRequest, 89](#)
- [setSenderAccount, 89](#)
- [setTitle, 89](#)
- [SofortLib\\_SofortueberweisungClassic, 89](#)
- [\\_\\_construct, 91](#)
- [\\_getPaymentDomain, 91](#)
- [addUserVariable, 91](#)
- [generatePassword, 91](#)
- [getError, 92](#)
- [getHashHexValue, 92](#)
- [getPaymentUrl, 92](#)
- [getSupportedHashAlgorithm, 92](#)
- [getSupportedHashAlgorithms, 92](#)
- [isError, 92](#)
- [setAbortUrl, 92](#)



- setAmount, 93
- setNotificationUrl, 93
- setReason, 93
- setSenderAccountNumber, 93
- setSenderHolder, 94
- setSuccessUrl, 94
- setVersion, 94
- SofortLib\_TransactionData, 94
  - \_\_construct, 97
  - \_parseXml, 97
  - addTransaction, 97
  - getAmount, 97
  - getAmountReceived, 97
  - getAmountRefunded, 98
  - getConsumerProtection, 98
  - getCount, 98
  - getCurrency, 98
  - getInvoiceAddress, 98
  - getInvoiceBankAccountNumber, 99
  - getInvoiceBankCode, 99
  - getInvoiceBankHolder, 99
  - getInvoiceBankName, 99
  - getInvoiceDate, 100
  - getInvoiceDebitorText, 100
  - getInvoiceDueDate, 100
  - getInvoiceNumber, 100
  - getInvoiceObjection, 101
  - getInvoiceReason, 101
  - getInvoiceStatus, 101
  - getInvoiceType, 101
  - getInvoiceUrl, 102
  - getItems, 102
  - getLanguageCode, 102
  - getOrderNumber, 102
  - getPaymentMethod, 103
  - getProjectId, 103
  - getReason, 103
  - getRecipientAccountNumber, 103
  - getRecipientBankCode, 104
  - getRecipientBankName, 104
  - getRecipientBic, 104
  - getRecipientCountryCode, 104
  - getRecipientHolder, 104
  - getRecipientIban, 105
  - getSenderAccountNumber, 105
  - getSenderBankCode, 105
  - getSenderBankName, 105
  - getSenderBic, 106
  - getSenderCountryCode, 106
  - getSenderHolder, 106
  - getSenderIban, 106
  - getShippingAddress, 107
  - getStatus, 107
  - getStatusModifiedTime, 107
  - getStatusReason, 107
  - getTime, 108
  - getTransaction, 108
  - getUserVariable, 108
  - isLastschrift, 108
  - isLoss, 109
  - isPending, 109
  - isReceived, 109
  - isRefunded, 109
  - isSofortlastschrift, 110
  - isSofortrechnung, 110
  - isSofortueberweisung, 110
  - isSofortvorkasse, 110
  - isTest, 111
  - setNumber, 111
  - setTime, 111
  - setTransaction, 112
- SofortTag, 112
  - \_\_construct, 113
  - \_render, 113
  - render, 113
- SofortText, 113
  - \_\_construct, 114
  - render, 114
- toArray
  - XmlToArray, 116
- toXml
  - ArrayToXml, 14
- updateCart
  - SofortLib\_ConfirmSr, 56
  - SofortLib\_EditSr, 65
- updateInvoice
  - PnagInvoice, 38
- updateInvoiceItem
  - PnagInvoice, 38
- updateOrderNumber
  - PnagInvoice, 38
- updateSofortrechnungItem
  - SofortLib\_Multipay, 83
- XmlToArray, 114
  - \_\_construct, 115
  - log, 115
  - render, 115

- toArray, [116](#)
- XmlToArrayException, [116](#)
- XmlToArrayNode, [116](#)
  - \_\_construct, [117](#)
  - addChild, [117](#)
  - getData, [117](#)
  - getName, [117](#)
  - getParentXmlToArrayNode, [117](#)
  - hasChildren, [117](#)
  - hasParentXmlToArrayNode, [117](#)
  - isOpen, [117](#)
  - render, [118](#)
  - setClosed, [118](#)
  - setData, [118](#)
  - setParentXmlToArrayNode, [118](#)