

# Hyperwave API

# Introduction

Hyperwave has been developed at » [IICM](#) in Graz. It started with the name Hyper-G and changed to Hyperwave when it was commercialised (in 1996).

Hyperwave is not free software. The current version, 5.5, is available at » <http://www.hyperwave.com/>. A time limited version can be ordered for free (30 days).

See also the [Hyperwave](#) module.

Hyperwave is an information system similar to a database ( HIS, Hyperwave Information Server). Its focus is the storage and management of documents. A document can be any possible piece of data that may as well be stored in file. Each document is accompanied by its object record. The object record contains meta data for the document. The meta data is a list of attributes which can be extended by the user. Certain attributes are always set by the Hyperwave server, other may be modified by the user.

<b>Note</b>
This extension has been moved to the » <a href="#">PECL</a> repository and is no longer bundled with PHP as of PHP 5.2.0.

# Installing/Configuring

## Requirements

Since 2001 there is a Hyperwave SDK available. It supports Java, JavaScript and C++. This PHP Extension is based on the C++ interface. In order to activate the hwapi support in PHP you will have to install the Hyperwave SDK first.

## Installation

After installing the Hyperwave SDK, configure PHP with `--with-hwapi[=DIR]`.

## Runtime Configuration

The behaviour of these functions is affected by settings in *php.ini*.

### Hyperwave API configuration options

Name	Default	Changeable	Changelog
hwapi.allow_persistent	"0"	PHP_INI_SYSTEM	

For further details and definitions of the `PHP_INI_*` constants, see the [php.ini directives](#).

## Resource Types

This extension has no resource types defined.

# Predefined Constants

This extension has no constants defined.

# Hyperwave API Functions

## Integration with Apache

The integration with Apache and possible other servers is already described in the [Hyperwave module](#) which has been the first extension to connect a Hyperwave Server.

## Classes

The API provided by the HW\_API extension is fully object oriented. It is very similar to the C++ interface of the Hyperwave SDK. It consist of the following classes.

- HW\_API
- HW\_API\_Object
- HW\_API\_Attribute
- HW\_API\_Error
- HW\_API\_Content
- HW\_API\_Reason

Some basic classes like HW\_API\_String, HW\_API\_String\_Array, etc., which exist in the Hyperwave SDK have not been implemented since PHP has powerful replacements for them.

Each class has certain method, whose names are identical to its counterparts in the Hyperwave SDK. Passing arguments to this function differs from all the other PHP extensions but is close to the C++ API of the HW SDK. Instead of passing several parameters they are all put into an associated array and passed as one parameter. The names of the keys are identical to those documented in the HW SDK. The common parameters are listed below. If other parameters are required they will be documented if needed.

- objectIdentifier The name or id of an object, e.g. "rootcollection", "0x873A87680x00000002".
- parentIdentifier The name or id of an object which is considered to be a parent.
- object An instance of class HW\_API\_Object.
- parameters An instance of class HW\_API\_Object.
- version The version of an object.
- mode An integer value determine the way an operation is executed.
- attributeSelector Any array of strings, each containing a name of an attribute. This is used if you retrieve the object record and want to include certain attributes.

- `objectQuery` A query to select certain object out of a list of objects. This is used to reduce the number of objects which was delivered by a function like [hw\\_api->children\(\)](#) or [hw\\_api->find\(\)](#).

<b>Note</b>
Methods returning <code>boolean</code> can return <b>TRUE</b> , <b>FALSE</b> or <code>HW_API_Error</code> object.

# hw\_api\_attribute->key

hw\_api\_attribute->key -- Returns key of the attribute

## Description

hw_api_attribute
------------------

string **key** ( void )

Returns the name of the attribute.

## Return Values

Returns the name of the attribute as a string.

## See Also

- [hw\\_api\\_attribute->value](#)

# hw\_api\_attribute->langdepvalue

hw\_api\_attribute->langdepvalue -- Returns value for a given language

## Description

<b>hw_api_attribute</b>
-------------------------

string **langdepvalue** ( string \$language )

Returns the value in the given language of the attribute.

## Parameters

*language*

## Return Values

Returns the value of the attribute as a string.

## See Also

- [hw\\_api\\_attribute->value](#)



# hw\_api\_attribute->value

hw\_api\_attribute->value -- Returns value of the attribute

## Description

hw_api_attribute
------------------

string **value** ( void )

Gets the value of the attribute.

## Return Values

Returns the value, as a string.

## See Also

- [hw\\_api\\_attribute->key](#)
- [hw\\_api\\_attribute->values](#)

# hw\_api\_attribute->values

hw\_api\_attribute->values -- Returns all values of the attribute

## Description

hw_api_attribute
------------------

array **values** ( void )

Gets all values of the attribute.

## Return Values

Returns an array of attribute values.

## See Also

- [hw\\_api\\_attribute->value](#)

# hw\_api\_attribute

hw\_api\_attribute -- Creates instance of class hw\_api\_attribute

## Description

[HW\\_API\\_Attribute](#) **hw\_api\_attribute** ( [ string \$name [, string \$value ] ] )

Creates a new instance of hw\_api\_attribute with the given name and value.

## Parameters

*name*

The attribute name.

*value*

The attribute value.

## Return Values

Returns an instance of hw\_api\_attribute.

# hw\_api->checkin

hw\_api->checkin -- Checks in an object

## Description

<b>hw_api</b>
---------------

bool **checkin** ( array \$parameter )

This function checks in an object or a whole hierarchy of objects. The parameters array contains the required element 'objectIdentifier' and the optional element 'version', 'comment', 'mode' and 'objectQuery'. 'version' sets the version of the object. It consists of the major and minor version separated by a period. If the version is not set, the minor version is incremented. 'mode' can be one of the following values:

HW\_API\_CHECKIN\_NORMAL

Checks in and commits the object. The object must be a document.

HW\_API\_CHECKIN\_RECURSIVE

If the object to check in is a collection, all children will be checked in recursively if they are documents. Trying to check in a collection would result in an error.

HW\_API\_CHECKIN\_FORCE\_VERSION\_CONTROL

Checks in an object even if it is not under version control.

HW\_API\_CHECKIN\_REVERT\_IF\_NOT\_CHANGED

Check if the new version is different from the last version. Unless this is the case the object will be checked in.

HW\_API\_CHECKIN\_KEEP\_TIME\_MODIFIED

Keeps the time modified from the most recent object.

HW\_API\_CHECKIN\_NO\_AUTO\_COMMIT

The object is not automatically committed on check-in.

## Parameters

*parameter*

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_api->checkout](#)

# hw\_api->checkout

hw\_api->checkout -- Checks out an object

## Description

hw_api
--------

bool **checkout** ( array \$parameter )

This function checks out an object or a whole hierarchy of objects.

## Parameters

*parameter*

The parameters array contains the required element 'objectIdentifier' and the optional element 'version', 'mode' and 'objectQuery'. 'mode' can be one of the following values:

### **HW\_API\_CHECKIN\_NORMAL**

Checks out an object. The object must be a document.

### **HW\_API\_CHECKIN\_RECURSIVE**

If the object to check out is a collection, all children will be checked out recursively if they are documents. Trying to check out a collection would result in an error.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_api->checkin](#)

# hw\_api->children

hw\_api->children -- Returns children of an object

## Description

hw_api
--------

array **children** ( array *\$parameter* )

Retrieves the children of a collection or the attributes of a document. The children can be further filtered by specifying an object query.

## Parameters

*parameter*

The parameter array contains the required elements 'objectIdentifier' and the optional elements 'attributeSelector' and 'objectQuery'.

## Return Values

The return value is an array of objects of type HW\_API\_Object or HW\_API\_Error.

## See Also

- [hw\\_api->parents](#)

# hw\_api\_content->mimetype

hw\_api\_content->mimetype -- Returns mimetype

## Description

<b>hw_api_content</b>
-----------------------

string **mimetype** ( void )

Returns the mimetype of the content.

## Return Values

Returns the mimetype as a string.



# hw\_api\_content->read

hw\_api\_content->read -- Read content

## Description

hw_api_content
----------------

string **read** ( string \$buffer, int \$len )

Reads *len* bytes from the content into the given buffer.

## Parameters

*buffer*

*len*

Number of bytes to read.

## Return Values

# hw\_api->content

hw\_api->content -- Returns content of an object

## Description

hw_api
--------

HW\_API\_Content **content** ( array *\$parameter* )

This function returns the content of a document as an object of type hw\_api\_content.

## Parameters

*parameter*

The parameter array contains the required elements 'objectidentifier' and the optional element 'mode'. The mode can be one of the constants

**HW\_API\_CONTENT\_ALLLINKS**, **HW\_API\_CONTENT\_REACHABLELINKS** or **HW\_API\_CONTENT\_PLAIN**. **HW\_API\_CONTENT\_ALLLINKS** means to insert all anchors even if the destination is not reachable.

**HW\_API\_CONTENT\_REACHABLELINKS** tells this method to insert only reachable links and **HW\_API\_CONTENT\_PLAIN** will lead to document without any links.

## Return Values

Returns an instance of hw\_api\_content.

# hw\_api->copy

hw\_api->copy -- Copies physically

## Description

hw_api
--------

hw\_api\_content **copy** ( array \$parameter )

This function will make a physical copy including the content if it exists and returns the new object or an error object.

## Parameters

*parameter*

The parameter array contains the required elements 'objectIdentifier' and 'destinationParentIdentifier'. The optional parameter is 'attributeSelector'

## Return Values

Returns the copied object.

## See Also

- [hw\\_api->move](#)
- [hw\\_api->link](#)

# hw\_api->dbstat

hw\_api->dbstat -- Returns statistics about database server

## Description

hw_api
--------

hw\_api\_object **dbstat** ( array \$parameter )

Returns statistics about database server.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->dcstat](#)
- [hw\\_api->hwstat](#)
- [hw\\_api->ftstat](#)

# hw\_api->dcstat

hw\_api->dcstat -- Returns statistics about document cache server

## Description

hw_api
--------

hw\_api\_object **dcstat** ( array \$parameter )

Returns statistics about document cache server.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->dbstat](#)
- [hw\\_api->hwstat](#)
- [hw\\_api->ftstat](#)

# hw\_api->dstanchors

hw\_api->dstanchors -- Returns a list of all destination anchors

## Description

hw_api
--------

array **dstanchors** ( array *\$parameter* )

Retrieves all destination anchors of an object.

## Parameters

*parameter*

The parameter array contains the required element 'objectIdentifier' and the optional elements 'attributeSelector' and 'objectQuery'.

## Return Values

## See Also

- [hw\\_api->srcanchors](#)

# hw\_api->dstofsrcanchor

hw\_api->dstofsrcanchor -- Returns destination of a source anchor

## Description

hw_api
--------

hw\_api\_object **dstofsrcanchor** ( array \$parameter )

Retrieves the destination object pointed by the specified source anchors. The destination object can either be a destination anchor or a whole document.

## Parameters

*parameter*

The parameters array contains the required element 'objectIdentifier' and the optional element 'attributeSelector'.

## Return Values

## See Also

- [hw\\_api->srcanchors](#)
- [hw\\_api->dstanchors](#)
- [hw\\_api->objectbyanchor](#)

# hw\_api\_error->count

hw\_api\_error->count -- Returns number of reasons

## Description

hw_api_error
--------------

int **count** ( void )

Returns the number of error reasons.

## Return Values

Returns the number of errors, as an integer.

## See Also

- [hw\\_api\\_error->reason](#)



# hw\_api\_error->reason

hw\_api\_error->reason -- Returns reason of error

## Description

hw_api_error
--------------

HW\_API\_Reason **reason** ( void )

Returns the first error reason.

## Return Values

## See Also

- [hw\\_api\\_error->count](#)

# hw\_api->find

hw\_api->find -- Search for objects

## Description

hw_api
--------

array **find** ( array \$parameter )

This functions searches for objects either by executing a key or/and full text query. The found objects can further be filtered by an optional object query. They are sorted by their importance. The second search operation is relatively slow and its result can be limited to a certain number of hits. This allows to perform an incremental search, each returning just a subset of all found documents, starting at a given index.

## Parameters

*parameter*

The parameter array contains the 'keyquery' or/and 'fulltextquery' depending on who you would like to search. Optional parameters are 'objectquery', 'scope', 'languages' and 'attributeselector'. In case of an incremental search the optional parameters 'startIndex', 'numberOfObjectsToGet' and 'exactMatchUnit' can be passed.

## Return Values

# hw\_api->ftstat

hw\_api->ftstat -- Returns statistics about fulltext server

## Description

hw_api
--------

hw\_api\_object **ftstat** ( array \$parameter )

Returns statistics about fulltext server.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->dcstat](#)
- [hw\\_api->dbstat](#)
- [hw\\_api->hwstat](#)

# hwapi\_hgcsp

hwapi\_hgcsp -- Returns object of class hw\_api

## Description

**HW\_API hwapi\_hgcsp** ( string \$hostname [, int \$port ] )

Opens a connection to the Hyperwave server on host *hostname*. The protocol used is HGCSP.

## Parameters

*hostname*

The host name.

*port*

If you do not pass a port number, 418 is used.

## Return Values

Returns an instance of HW\_API.

# hw\_api->hwstat

hw\_api->hwstat -- Returns statistics about Hyperwave server

## Description

hw_api
--------

hw\_api\_object **hwstat** ( array \$parameter )

Returns statistics about Hyperwave server.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->dcstat](#)
- [hw\\_api->dbstat](#)
- [hw\\_api->ftstat](#)

# hw\_api->identify

hw\_api->identify -- Log into Hyperwave Server

## Description

hw_api
--------

bool **identify** ( array \$parameter )

Logs into the Hyperwave Server.

## Parameters

*parameter*

The parameter array must contain the elements 'username' and 'password'.

## Return Values

Returns an object of type HW\_API\_Error if identification failed or **TRUE** if it was successful.

# hw\_api->info

hw\_api->info -- Returns information about server configuration

## Description

hw_api
--------

array **info** ( array \$parameter )

Returns information about server configuration.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->dcstat](#)
- [hw\\_api->dbstat](#)
- [hw\\_api->ftstat](#)
- [hw\\_api->hwstat](#)

# hw\_api->insert

hw\_api->insert -- Inserts a new object

## Description

hw_api
--------

hw\_api\_object **insert** ( array \$parameter )

Insert a new object. The object type can be user, group, document or anchor. Depending on the type other object attributes has to be set.

## Parameters

*parameter*

The parameter array contains the required elements 'object' and 'content' (if the object is a document) and the optional parameters 'parameters', 'mode' and 'attributeSelector'. The 'object' must contain all attributes of the object. 'parameters' is an object as well holding further attributes like the destination (attribute key is 'Parent'). 'content' is the content of the document. 'mode' can be a combination of the following flags:

**HW\_API\_INSERT\_NORMAL**

The object is inserted into the server.

**HW\_API\_INSERT\_FORCE\_VERSION\_CONTROL**

**HW\_API\_INSERT\_AUTOMATIC\_CHECKOUT**

**HW\_API\_INSERT\_PLAIN**

**HW\_API\_INSERT\_KEEP\_TIME\_MODIFIED**

**HW\_API\_INSERT\_DELAY\_INDEXING**

## Return Values



## See Also

- [hw\\_api->replace](#)

# hw\_api->insertanchor

hw\_api->insertanchor -- Inserts a new object of type anchor

## Description

hw_api
--------

hw\_api\_object **insertanchor** ( array \$parameter )

This function is a shortcut for **hwapi\_insert()**. It inserts an object of type anchor and sets some of the attributes required for an anchor.

## Parameters

*parameter*

The parameter array contains the required elements 'object' and 'documentIdentifier' and the optional elements 'destinationIdentifier', 'parameter', 'hint' and 'attributeSelector'. The 'documentIdentifier' specifies the document where the anchor shall be inserted. The target of the anchor is set in 'destinationIdentifier' if it already exists. If the target does not exist the element 'hint' has to be set to the name of object which is supposed to be inserted later. Once it is inserted the anchor target is resolved automatically.

## Return Values

## See Also

- [hw\\_api->insert](#)
- [hw\\_api->insertdocument](#)
- [hw\\_api->insertcollection](#)

# hw\_api->insertcollection

hw\_api->insertcollection -- Inserts a new object of type collection

## Description

hw_api
--------

hw\_api\_object **insertcollection** ( array \$parameter )

This function is a shortcut for **hwapi\_insert()**. It inserts an object of type collection and sets some of the attributes required for a collection.

## Parameters

*parameter*

The parameter array contains the required elements 'object' and 'parentIdentifier' and the optional elements 'parameter' and 'attributeSelector'. See **hwapi\_insert()** for the meaning of each element.

## Return Values

## See Also

- [hw\\_api->insert](#)
- [hw\\_api->insertdocument](#)
- [hw\\_api->insertanchor](#)

# hw\_api->insertdocument

hw\_api->insertdocument -- Inserts a new object of type document

## Description

hw_api
--------

hw\_api\_object **insertdocument** ( array \$parameter )

This function is a shortcut for **hwapi\_insert()**. It inserts an object with content and sets some of the attributes required for a document.

## Parameters

*parameter*

The parameter array contains the required elements 'object', 'parentIdentifier' and 'content' and the optional elements 'mode', 'parameter' and 'attributeSelector'. See **hwapi\_insert()** for the meaning of each element.

## Return Values

## See Also

- [hw\\_api->insert](#)
- [hw\\_api->insertcollection](#)
- [hw\\_api->insertanchor](#)

# hw\_api->link

hw\_api->link -- Creates a link to an object

## Description

hw_api
--------

bool **link** ( array \$parameter )

Creates a link to an object. Accessing this link is like accessing the object to links points to.

## Parameters

*parameter*

The parameter array contains the required elements 'objectIdentifier' and 'destinationParentIdentifier'. 'destinationParentIdentifier' is the target collection.

## Return Values

The function returns **TRUE** on success or an error object.

## See Also

- [hw\\_api->copy](#)

# hw\_api->lock

hw\_api->lock -- Locks an object

## Description

hw_api
--------

bool **lock** ( array \$parameter )

Locks an object for exclusive editing by the user calling this function. The object can be only unlocked by this user or the system user.

## Parameters

*parameter*

The parameter array contains the required element 'objectIdentifier' and the optional parameters 'mode' and 'objectquery'. 'mode' determines how an object is locked.

**HW\_API\_LOCK\_NORMAL** means, an object is locked until it is unlocked.

**HW\_API\_LOCK\_RECURSIVE** is only valid for collection and locks all objects within the collection and possible subcollections. **HW\_API\_LOCK\_SESSION** means, an object is locked only as long as the session is valid.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_api->unlock](#)

# hw\_api->move

hw\_api->move -- Moves object between collections

## Description

hw_api
--------

bool **move** ( array \$parameter )

Moves object between collections.

## Parameters

*parameter*

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_objrec2array\(\)](#)

# hw\_api\_content

hw\_api\_content -- Create new instance of class hw\_api\_content

## Description

[HW\\_API\\_Content](#) **hw\_api\_content** ( string \$content, string \$mimetype )

Creates a new content object from the string *content*.

## Parameters

*content*

*mimetype*

The mimetype for the contents.

## Return Values



# hw\_api\_object->assign

hw\_api\_object->assign -- Clones object

## Description

hw_api_object
---------------

bool **assign** ( array \$parameter )

Clones the attributes of an object.

## Parameters

*parameter*

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

# hw\_api\_object->attreditable

hw\_api\_object->attreditable -- Checks whether an attribute is editable

## Description

hw_api_object
---------------

bool **attreditable** ( array \$parameter )

Checks whether an attribute is editable.

## Parameters

*parameter*

## Return Values

Returns **TRUE** if the attribute is editable, **FALSE** otherwise.

# hw\_api\_object->count

hw\_api\_object->count -- Returns number of attributes

## Description

hw_api_object
---------------

int **count** ( array \$parameter )

Returns the number of attributes.

## Parameters

*parameter*

## Return Values

Returns the number as an integer.

# hw\_api\_object->insert

hw\_api\_object->insert -- Inserts new attribute

## Description

hw_api_object
---------------

bool **insert** ( [HW\\_API\\_Attribute](#) \$attribute )

Adds an attribute to the object.

## Parameters

*attribute*

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_api\\_object->remove](#)

# hw\_api\_object

hw\_api\_object -- Creates a new instance of class hw\_api\_object

## Description

[hw\\_api\\_object](#) **hw\_api\_object** ( array \$parameter )

Creates a new instance of the class hw\_api\_object.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->lock](#)

# hw\_api\_object->remove

hw\_api\_object->remove -- Removes attribute

## Description

hw_api_object
---------------

bool **remove** ( string \$name )

Removes the attribute with the given name.

## Parameters

*name*

The attribute name.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_api\\_object->insert](#)

# hw\_api\_object->title

hw\_api\_object->title -- Returns the title attribute

## Description

<b>hw_api_object</b>
----------------------

string **title** ( array \$parameter )

Returns the title attribute.

## Parameters

*parameter*

## Return Values

Returns the title as a string.

# hw\_api\_object->value

hw\_api\_object->value -- Returns value of attribute

## Description

hw_api_object
---------------

string **value** ( string \$name )

Returns value of an attribute.

## Parameters

*name*

The attribute name.

## Return Values

Returns the value of the attribute with the given name or **FALSE** if an error occurred.



# hw\_api->object

hw\_api->object -- Retrieve attribute information

## Description

<b>hw_api</b>
---------------

hw\_api\_object **object** ( array \$parameter )

This function retrieves the attribute information of an object of any version. It will not return the document content.

## Parameters

*parameter*

The parameter array contains the required elements 'objectIdentifier' and the optional elements 'attributeSelector' and 'version'.

## Return Values

The returned object is an instance of class HW\_API\_Object on success or HW\_API\_Error if an error occurred.

## Examples

This simple example retrieves an object and checks for errors.

<b>Example #1 - Retrieve an object</b>
--

<pre>&lt;?php function handle_error(\$error) {     \$reason = \$error-&gt;reason(0);     echo "Type: &lt;b&gt;";     switch (\$reason-&gt;type()) {         case 0:             echo "Error";             break;         case 1:             echo "Warning";             break;         case 2:             echo "Message";             break;     } }</pre>
--

```

    }
    echo "</b><br />\n";
    echo "Description: " . $reason->description("en") . "<br />\n";
}

function list_attr($obj)
{
    echo "<table>\n";
    $count = $obj->count();
    for ($i=0; $i<$count; $i++) {
        $attr = $obj->attribute($i);
        printf("<tr><td align=\"right\" bgcolor=\"\#c0c0c0\"><b>%s</b></td><td  

        bgcolor=\"\#F0F0F0\">%s</td></tr>\n",
            $attr->key(), $attr->value());
    }
    echo "</table>\n";
}

$hwwapi = hwwapi_hgcsp($g_config[HOSTNAME]);
$params = array("objectIdentifier"=>"rootcollection",
    "attributeSelector"=>array("Title", "Name", "DocumentType"));
$root = $hwwapi->object($params);
if (get_class($root) == "HW_API_Error") {
    handle_error($root);
    exit;
}
list_attr($root);
?>

```

## See Also

- [hw\\_api->content](#)

# hw\_api->objectbyanchor

hw\_api->objectbyanchor -- Returns the object an anchor belongs to

## Description

hw_api
--------

hw\_api\_object **objectbyanchor** ( array \$parameter )

This function retrieves an object the specified anchor belongs to.

## Parameters

*parameter*

The parameter array contains the required element 'objectIdentifier' and the optional element 'attributeSelector'.

## Return Values

## See Also

- [hw\\_api->dstofsrcanchor](#)
- [hw\\_api->srcanchors](#)
- [hw\\_api->dstanchors](#)

# hw\_api->parents

hw\_api->parents -- Returns parents of an object

## Description

hw_api
--------

array **parents** ( array *\$parameter* )

Retrieves the parents of an object. The parents can be further filtered by specifying an object query.

## Parameters

*parameter*

The parameter array contains the required elements 'objectidentifier' and the optional elements 'attributeselector' and 'objectquery'.

## Return Values

The return value is an array of objects of type HW\_API\_Object or HW\_API\_Error.

## See Also

- [hw\\_api->children](#)

# hw\_api\_reason->description

hw\_api\_reason->description -- Returns description of reason

## Description

hw_api_reason
---------------

string **description** ( void )

Returns the description of a reason

## Return Values

Returns the description, as a string.

# hw\_api\_reason->type

hw\_api\_reason->type -- Returns type of reason

## Description

hw_api_reason
---------------

HW\_API\_Reason **type** ( void )

Returns the type of a reason.

## Return Values

Returns an instance of HW\_API\_Reason.

# hw\_api->remove

hw\_api->remove -- Delete an object

## Description

hw_api
--------

bool **remove** ( array \$parameter )

Removes an object from the specified parent. Collections will be removed recursively.

## Parameters

*parameter*

You can pass an optional object query to remove only those objects which match the query. An object will be deleted physically if it is the last instance. The parameter array contains the required elements 'objectidentifier' and 'parentidentifier'. If you want to remove a user or group 'parentidentifier' can be skipped. The optional parameter 'mode' determines how the deletion is performed. In normal mode the object will not be removed physically until all instances are removed. In physical mode all instances of the object will be deleted immediately. In removelinks mode all references to and from the objects will be deleted as well. In nonrecursive the deletion is not performed recursive. Removing a collection which is not empty will cause an error.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [hw\\_api->move](#)

# hw\_api->replace

hw\_api->replace -- Replaces an object

## Description

hw_api
--------

hw\_api\_object **replace** ( array \$parameter )

Replaces the attributes and the content of an object.

## Parameters

*parameter*

The parameter array contains the required elements 'objectIdentifier' and 'object' and the optional parameters 'content', 'parameters', 'mode' and 'attributeSelector'.

'objectIdentifier' contains the object to be replaced. 'object' contains the new object. 'content' contains the new content. 'parameters' contain extra information for HTML documents. HTML\_Language is the letter abbreviation of the language of the title. HTML\_Base sets the base attribute of the HTML document. 'mode' can be a combination of the following flags:

**HW\_API\_REPLACE\_NORMAL**

The object on the server is replaced with the object passed.

**HW\_API\_REPLACE\_FORCE\_VERSION\_CONTROL**

**HW\_API\_REPLACE\_AUTOMATIC\_CHECKOUT**

**HW\_API\_REPLACE\_AUTOMATIC\_CHECKIN**

**HW\_API\_REPLACE\_PLAIN**

**HW\_API\_REPLACE\_REVERT\_IF\_NOT\_CHANGED**

**HW\_API\_REPLACE\_KEEP\_TIME\_MODIFIED**



## Return Values

## See Also

- [hw\\_api->insert](#)

# hw\_api->setcommittedversion

hw\_api->setcommittedversion -- Commits version other than last version

## Description

hw_api
--------

hw\_api\_object **setcommittedversion** ( array \$parameter )

Commits a version of a document. The committed version is the one which is visible to users with read access. By default the last version is the committed version.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->checkin](#)
- [hw\\_api->checkout](#)

# hw\_api->srcanchors

hw\_api->srcanchors -- Returns a list of all source anchors

## Description

hw_api
--------

array **srcanchors** ( array *\$parameter* )

Retrieves all source anchors of an object.

## Parameters

*parameter*

The parameter array contains the required element 'objectIdentifier' and the optional elements 'attributeSelector' and 'objectQuery'.

## Return Values

## See Also

- [hw\\_api->dstanchors](#)

# hw\_api->srcsofdst

hw\_api->srcsofdst -- Returns source of a destination object

## Description

hw_api
--------

array **srcsofdst** ( array *\$parameter* )

Retrieves all the source anchors pointing to the specified destination. The destination object can either be a destination anchor or a whole document.

## Parameters

*parameter*

The parameters array contains the required element 'objectIdentifier' and the optional element 'attributeSelector' and 'objectQuery'. The function returns an array of objects or an error.

## Return Values

## See Also

- [hw\\_api->dstofsrcanchor](#)

# hw\_api->unlock

hw\_api->unlock -- Unlocks a locked object

## Description

hw_api
--------

bool **unlock** ( array \$parameter )

Unlocks a locked object. Only the user who has locked the object and the system user may unlock an object.

## Parameters

*parameter*

The parameter array contains the required element 'objectIdentifier' and the optional parameters 'mode' and 'objectquery'. The meaning of 'mode' is the same as in function **hwapi\_lock()**.

## Return Values

Returns **TRUE** on success or an object of class HW\_API\_Error.

## See Also

- [hw\\_api->lock](#)

# hw\_api->user

hw\_api->user -- Returns the own user object

## Description

hw_api
--------

hw\_api\_object **user** ( array \$parameter )

Returns the own user object.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->userlist](#)

# hw\_api->userlist

hw\_api->userlist -- Returns a list of all logged in users

## Description

hw_api
--------

array **userlist** ( array \$parameter )

Returns a list of all logged in users.

## Parameters

*parameter*

## Return Values

## See Also

- [hw\\_api->user](#)