
libzookeeper Documentation

Release latest

December 19, 2016

1	Introduction	3
2	Installing/Configuring	5
2.1	Requirements	5
2.2	Installation	5
3	Predefined Constants	7
4	Examples	9
4.1	No.1	9
4.2	Create Node	9
4.3	Add Auth	9
4.4	Get Acls	9
4.5	Set Acls	10
4.6	After a fork	10
5	ZookeeperClient	11
5.1	ZookeeperClient::connect	11
5.2	ZookeeperClient::get	11
5.3	ZookeeperClient::getChildren	12
5.4	ZookeeperClient::create	12
5.5	ZookeeperClient::delete	12
5.6	ZookeeperClient::exists	13
5.7	ZookeeperClient::set	13
5.8	ZookeeperClient::close	14
5.9	ZookeeperClient::addAuth	14
5.10	ZookeeperClient::setAcls	14
5.11	ZookeeperClient::getAcls	15
5.12	ZookeeperClient::setLogLevel	15
6	Exceptions	17
6.1	ZookeeperClientException	17
6.2	ZookeeperClientCoreException	17

Contents:

Introduction

This is an extension that allows people communicate with Zookeeper cluster using PHP.

Installing/Configuring

Contents:

2.1 Requirements

This extension requires Zookeeper C Client (<https://zookeeper.apache.org/>).

2.2 Installation

```
$ phpize
$ ./configure --with-libzookeeper=/path/to/zookeeper-c-client
$ make
$ make install
```

Predefined Constants

ZookeeperClient::ERR_OK 0, /*!< Everything is OK */

ZookeeperClient::ERR_SYSTEMERROR -1,

ZookeeperClient::ERR_RUNTIMEINCONSISTENCY -2, /*!< A runtime inconsistency was found */

ZookeeperClient::ERR_DATAINCONSISTENCY -3, /*!< A data inconsistency was found */

ZookeeperClient::ERR_CONNECTIONLOSS -4, /*!< Connection to the server has been lost */

ZookeeperClient::ERR_MARSHALLINGERROR -5, /*!< Error while marshalling or unmarshalling data */

ZookeeperClient::ERR_UNIMPLEMENTED -6, /*!< Operation is unimplemented */

ZookeeperClient::ERR_OPERATIONTIMEOUT -7, /*!< Operation timeout */

ZookeeperClient::ERR_BADARGUMENTS -8, /*!< Invalid arguments */

ZookeeperClient::ERR_INVALIDSTATE -9, /*!< Invalid zhandle state */

ZookeeperClient::ERR_APIERROR -100,

ZookeeperClient::ERR_NONODE -101, /*!< Node does not exist */

ZookeeperClient::ERR_NOAUTH -102, /*!< Not authenticated */

ZookeeperClient::ERR_BADVERSION -103, /*!< Version conflict */

ZookeeperClient::ERR_NOCHILDRENFOREPHEMERALS -108, /*!< Ephemeral nodes may not have children */

ZookeeperClient::ERR_NODEEXISTS -110, /*!< The node already exists */

ZookeeperClient::ERR_NOTEMPTY -111, /*!< The node has children */

ZookeeperClient::ERR_SESSIONEXPIRED -112, /*!< The session has been expired by the server */

ZookeeperClient::ERR_INVALIDCALLBACK -113, /*!< Invalid callback specified */

ZookeeperClient::ERR_INVALIDACL -114, /*!< Invalid ACL specified */

ZookeeperClient::ERR_AUTHFAILED -115, /*!< Client authentication failed */

ZookeeperClient::ERR_CLOSING -116, /*!< ZooKeeper is closing */

ZookeeperClient::ERR_NOTHING -117, /*!< (not error) no server responses to process */

ZookeeperClient::ERR_SESSIONMOVED -118, /*!< session moved to another server, so operation is ignored */

(Zookeeper >= 3.5.0):

ZookeeperClient::ERR_NEWCONFIGNOQUORUM -13, /*!< No quorum of new config is connected and up-to-date with the leader of last committed config - try invoking reconfiguration after new servers are connected and synced */

ZookeeperClient::ERR_RECONFIGINPROGRESS -14, /*!< Reconfiguration requested while another reconfiguration is currently in progress. This is currently not supported. Please retry. */

ZookeeperClient::ERR_NOTREADONLY -119, /*!< state-changing request is passed to read-only server */

ZookeeperClient::ERR_EPHEMERALONLOCALSESSION -120, /*!< Attempt to create ephemeral node on a local session */

ZookeeperClient::ERR_NOWATCHER -121, /*!< The watcher couldn't be found */

ZookeeperClient::ERR_RWSERVERFOUND -122, /*!< r/w server found while in r/o mode */

ZookeeperClient::LOG_LEVEL_NONE 0,

ZookeeperClient::LOG_LEVEL_ERROR 1,

ZookeeperClient::LOG_LEVEL_WARN 2,

ZookeeperClient::LOG_LEVEL_INFO 3,

ZookeeperClient::LOG_LEVEL_DEBUG 4,

ZookeeperClient::PERM_READ 1, //can read node's value and list its children

ZookeeperClient::PERM_WRITE 2, // can set the node's value

ZookeeperClient::PERM_CREATE 4, //can create children

ZookeeperClient::PERM_DELETE 8, // can delete children

ZookeeperClient::PERM_ADMIN 16, //can execute set_acl()

ZookeeperClient::PERM_ALL 0x1f, // all of the above flags OR'd together

ZookeeperClientException::ERR_INIT_FAILURE 1,

ZookeeperClientException::ERR_CONNECT_FIRST 2,

ZookeeperClientException::ERR_CLOSE_FIRST 3,

Examples

4.1 No.1

```
<?php
$zc = new ZookeeperClient();
$zc->connect('localhost:2181');
var_dump($zc->get('/zookeeper'));
?>
```

4.2 Create Node

```
<?php
$zc = new \ZookeeperClient();
$zc->connect('localhost:2181');

$acls = array (
    'world:anyone' => \ZookeeperClient::PERM_READ,
    'digest:user0:4LVnEr8VX5VVtn/KiU6Aff8jqh8=' => \ZookeeperClient::PERM_ALL,
    'digest:user1:4LVnEr8VX5VVtn/KiU6Aff8jqh8=' => \ZookeeperClient::PERM_CREATE & \ZookeeperClient::
);
$zc->create('/zookeeper/node0', 'value0', $acls);
```

4.3 Add Auth

```
<?php
$zc = new \ZookeeperClient();
$zc->connect('localhost:2181');

$zc->addAuth('digest', 'user0:password0');
```

4.4 Get Acls

```
<?php
$zc = new \ZookeeperClient();
$zc->connect('localhost:2181');
```

```
$acls = $zc->getAcls('/zookeeper/node0');
var_dump($acls);
```

4.5 Set Acls

```
<?php
$zc = new \ZookeeperClient();
$zc->connect('localhost:2181');

$acls = array (
    'world:anyone' => \ZookeeperClient::PERM_READ,
    'digest:user0:4LVnEr8VX5VVtn/KiU6Aff8jqh8=' => \ZookeeperClient::PERM_ALL,
    'digest:user1:4LVnEr8VX5VVtn/KiU6Aff8jqh8=' => \ZookeeperClient::PERM_CREATE & \ZookeeperClient::
);
$zc->setAcls('/zookeeper/node0', $acls);
```

4.6 After a fork

```
<?php
$zc = new ZookeeperClient();
$zc->connect('localhost:2181');
var_dump($zc->get('/zookeeper'));

$pid = pcntl_fork();
if ($pid < 0)
    die("Error");
elseif ($pid == 0) { // child
    $zc->close();
    $zc->connect('localhost:2181');

    var_dump($zc->get('/zookeeper'));
    exit;
}

// parent
var_dump($zc->get('/zookeeper'));

?>
```

ZookeeperClient

Contents:

5.1 ZookeeperClient::connect

ZookeeperClient::connect — Connect to Zookeeper cluster

5.1.1 Description

```
public void ZookeeperClient::connect (string $hosts)
```

5.1.2 Parameters

hosts CSV list of host:port values (e.g. “host1:2181,host2:2181”)

5.1.3 Return Values

No value is returned.

5.2 ZookeeperClient::get

ZookeeperClient::get — Get node value from Zookeeper cluster

5.2.1 Description

```
public string ZookeeperClient::get (string $path)
```

5.2.2 Parameters

path Path of the node (e.g. “/zookeeper/quota”)

5.2.3 Return Values

Returns value as a string(NULL for empty value) on success, or false on error.

5.3 ZookeeperClient::getChildren

ZookeeperClient::getChildren — Get child nodes of given node

5.3.1 Description

```
public array ZookeeperClient::getChildren(string $path)
```

5.3.2 Parameters

path Path of the node (e.g. “/zookeeper/quota”)

5.3.3 Return Values

Returns array filled of names of child nodes on success, or false on error.

5.4 ZookeeperClient::create

ZookeeperClient::create — Create a new node

5.4.1 Description

```
public string ZookeeperClient::create(string $path [, string $value], array $acls)
```

5.4.2 Parameters

path Path of the node (e.g. “/zookeeper/quota”)

value Default value of the new node

acls The acls to be set on the path

5.4.3 Return Values

Returns real path of the new node on success, or false on error.

5.5 ZookeeperClient::delete

ZookeeperClient::delete — Delete a existing node

5.5.1 Description

```
public bool ZookeeperClient::delete(string $path)
```

5.5.2 Parameters

path Path of the node (e.g. “/zookeeper/quota”)

5.5.3 Return Values

Returns true on success, or false on error.

5.6 ZookeeperClient::exists

ZookeeperClient::exists — Check existence of some node

5.6.1 Description

```
public bool ZookeeperClient::exists(string $path)
```

5.6.2 Parameters

path Path of the node (e.g. “/zookeeper/quota”)

5.6.3 Return Values

Returns true when node existing, otherwise returns false.

5.7 ZookeeperClient::set

ZookeeperClient::set — Sets the data associated with a node

5.7.1 Description

```
public void ZookeeperClient::set(string $path, string $value)
```

5.7.2 Parameters

path path the name of the node. Expressed as a file name with slashes separating ancestors of the node.

value data to be written to the node.

5.7.3 Return Values

No value is returned.

5.8 ZookeeperClient::close

ZookeeperClient::close — Close the zookeeper handle and free up any resources

5.8.1 Description

```
public void ZookeeperClient::close()
```

5.8.2 Parameters

No arguments.

5.8.3 Return Values

No value is returned.

5.9 ZookeeperClient::addAuth

ZookeeperClient::addAuth — Specify application credentials

5.9.1 Description

```
public void ZookeeperClient::addAuth(string $scheme, string $cert)
```

5.9.2 Parameters

scheme the id of authentication scheme.

cert application credentials. The actual value depends on the scheme.

5.9.3 Return Values

No value is returned.

5.10 ZookeeperClient::setAcls

ZookeeperClient::setAcls — sets the acl associated with a node synchronously

5.10.1 Description

```
public void ZookeeperClient::setAcls(string $path, array $acls)
```

5.10.2 Parameters

path path the name of the node. Expressed as a file name with slashes separating ancestors of the node.

acls the acl to be set on the path.

5.10.3 Return Values

No value is returned.

5.11 ZookeeperClient::getAcls

ZookeeperClient::getAcls — Get node value from Zookeeper cluster

5.11.1 Description

```
public string ZookeeperClient::getAcls(string $path)
```

5.11.2 Parameters

path the name of the node. Expressed as a file name with slashes separating ancestors of the node.

5.11.3 Return Values

Returns acls as a assoc array on success, or false on error.

5.12 ZookeeperClient::setLogLevel

ZookeeperClient::setLogLevel — sets the debugging level for the library

5.12.1 Description

```
public static void ZookeeperClient::setLogLevel(int $logLevel)
```

5.12.2 Parameters

logLevel Zookeeper::LOG_LEVEL_* constants.

5.12.3 Return Values

No value is returned.

Exceptions

Contents:

6.1 ZookeeperClientException

6.1.1 Introduction

Exception thrown if anything wrong occurred.

6.1.2 Class synopsis

```
class ZookeeperClientException extends RuntimeException {  
}
```

6.2 ZookeeperClientCoreException

6.2.1 Introduction

Exception thrown if anything wrong about zookeeper client occurred.

Method *getCode()* returns the error code which is returned by Zookeeper C Client, and method *getMessage()* returns the error message which is returned by function 'zerror()'.

6.2.2 Class synopsis

```
class ZookeeperClientCoreException extends ZookeeperClientException {  
}
```