ZooKeeper Command Line Interface (CLI) is used to interact with the ZooKeeper ensemble for development purpose. It is useful for debugging and working around with different options.

To perform ZooKeeper CLI operations, first turn on your ZooKeeper server (*“bin/zkServer.sh start”*) and then, ZooKeeper client (*“bin/zkCli.sh”*). Once the client starts, you can perform the following operation −

* Create znodes
* Get data
* Watch znode for changes
* Set data
* Create children of a znode
* List children of a znode
* Check Status
* Remove / Delete a znode

Now let us see above command one by one with an example.

Create Znodes

Create a znode with the given path. The **flag** argument specifies whether the created znode will be ephemeral, persistent, or sequential. By default, all znodes are persistent.

* **Ephemeral znodes** (flag: e) will be automatically deleted when a session expires or when the client disconnects.
* **Sequential znodes** guaranty that the znode path will be unique.
* ZooKeeper ensemble will add sequence number along with 10 digit padding to the znode path. For example, the znode path */myapp will be converted to /myapp0000000001* and the next sequence number will be */myapp0000000002*. If no flags are specified, then the znode is considered as **persistent**.

Syntax

create /path /data

Sample

create /FirstZnode “Myfirstzookeeper-app”

Output

[zk: localhost:2181(CONNECTED) 0] create /FirstZnode “Myfirstzookeeper-app”

Created /FirstZnode

To create a **Sequential znode**, add **-s flag** as shown below.

Syntax

create -s /path /data

Sample

create -s /FirstZnode second-data

Output

[zk: localhost:2181(CONNECTED) 2] create -s /FirstZnode “second-data”

Created /FirstZnode0000000023

To create an **Ephemeral Znode**, add **-e flag** as shown below.

Syntax

create -e /path /data

Sample

create -e /SecondZnode “Ephemeral-data”

Output

[zk: localhost:2181(CONNECTED) 2] create -e /SecondZnode “Ephemeral-data”

Created /SecondZnode

Remember when a client connection is lost, the ephemeral znode will be deleted. You can try it by quitting the ZooKeeper CLI and then re-opening the CLI.

Get Data

It returns the associated data of the znode and metadata of the specified znode. You will get information such as when the data was last modified, where it was modified, and information about the data. This CLI is also used to assign watches to show notification about the data.

Syntax

get /path

Sample

get /FirstZnode

Output

[zk: localhost:2181(CONNECTED) 1] get /FirstZnode

“Myfirstzookeeper-app”

cZxid = 0x7f

ctime = Tue Sep 29 16:15:47 IST 2015

mZxid = 0x7f

mtime = Tue Sep 29 16:15:47 IST 2015

pZxid = 0x7f

cversion = 0

dataVersion = 0

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 22

numChildren = 0

To access a sequential znode, you must enter the full path of the znode.

Sample

get /FirstZnode0000000023

Output

[zk: localhost:2181(CONNECTED) 1] get /FirstZnode0000000023

“Second-data”

cZxid = 0x80

ctime = Tue Sep 29 16:25:47 IST 2015

mZxid = 0x80

mtime = Tue Sep 29 16:25:47 IST 2015

pZxid = 0x80

cversion = 0

dataVersion = 0

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 13

numChildren = 0

Watch

Watches show a notification when the specified znode or znode’s children data changes. You can set a **watch** only in **get** command.

Syntax

get /path [watch] 1

Sample

get /FirstZnode 1

Output

[zk: localhost:2181(CONNECTED) 1] get /FirstZnode 1

“Myfirstzookeeper-app”

cZxid = 0x7f

ctime = Tue Sep 29 16:15:47 IST 2015

mZxid = 0x7f

mtime = Tue Sep 29 16:15:47 IST 2015

pZxid = 0x7f

cversion = 0

dataVersion = 0

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 22

numChildren = 0

The output is similar to normal **get** command, but it will wait for znode changes in the background. <Start here>

Set Data

Set the data of the specified znode. Once you finish this set operation, you can check the data using the **get** CLI command.

Syntax

set /path /data

Sample

set /SecondZnode Data-updated

Output

[zk: localhost:2181(CONNECTED) 1] get /SecondZnode “Data-updated”

cZxid = 0x82

ctime = Tue Sep 29 16:29:50 IST 2015

mZxid = 0x83

mtime = Tue Sep 29 16:29:50 IST 2015

pZxid = 0x82

cversion = 0

dataVersion = 1

aclVersion = 0

ephemeralOwner = 0x15018b47db00000

dataLength = 14

numChildren = 0

If you assigned **watch** option in **get** command (as in previous command), then the output will be similar as shown below −

Output

[zk: localhost:2181(CONNECTED) 1] get /FirstZnode “Mysecondzookeeper-app”

WATCHER: :

WatchedEvent state:SyncConnected type:NodeDataChanged path:/FirstZnode

cZxid = 0x7f

ctime = Tue Sep 29 16:15:47 IST 2015

mZxid = 0x84

mtime = Tue Sep 29 17:14:47 IST 2015

pZxid = 0x7f

cversion = 0

dataVersion = 1

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 23

numChildren = 0

Create Children / Sub-znode

Creating children is similar to creating new znodes. The only difference is that the path of the child znode will have the parent path as well.

Syntax

create /parent/path/subnode/path /data

Sample

create /FirstZnode/Child1 firstchildren

Output

[zk: localhost:2181(CONNECTED) 16] create /FirstZnode/Child1 “firstchildren”

created /FirstZnode/Child1

[zk: localhost:2181(CONNECTED) 17] create /FirstZnode/Child2 “secondchildren”

created /FirstZnode/Child2

List Children

This command is used to list and display the **children** of a znode.

Syntax

ls /path

Sample

ls /MyFirstZnode

Output

[zk: localhost:2181(CONNECTED) 2] ls /MyFirstZnode

[mysecondsubnode, myfirstsubnode]

Check Status

**Status** describes the metadata of a specified znode. It contains details such as Timestamp, Version number, ACL, Data length, and Children znode.

Syntax

stat /path

Sample

stat /FirstZnode

Output

[zk: localhost:2181(CONNECTED) 1] stat /FirstZnode

cZxid = 0x7f

ctime = Tue Sep 29 16:15:47 IST 2015

mZxid = 0x7f

mtime = Tue Sep 29 17:14:24 IST 2015

pZxid = 0x7f

cversion = 0

dataVersion = 1

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 23

numChildren = 0

Remove a Znode

Removes a specified znode and recursively all its children. This would happen only if such a znode is available.

Syntax

rmr /path

Sample

rmr /FirstZnode

Output

[zk: localhost:2181(CONNECTED) 10] rmr /FirstZnode

[zk: localhost:2181(CONNECTED) 11] get /FirstZnode

Node does not exist: /FirstZnode

Delete **(delete /path)** command is similar to **remove** command, except the fact that it works only on znodes with no children.