ZooKeeper命令行界面（CLI）用于与ZooKeeper集合进行交互以进行开发。它有助于调试和解决不同的选项。

要执行ZooKeeper CLI操作，首先打开ZooKeeper服务器（“bin/zkServer.sh start”），然后打开ZooKeeper客户端（“bin/zkCli.sh”）。一旦客户端启动，你可以执行以下操作：  
1、创建znode  
2、获取数据  
3、监视znode的变化  
4、设置数据  
5、创建znode的子节点  
6、列出znode的子节点  
7、检查状态  
8、移除/删除znode  
现在让我们用一个例子逐个了解上面的命令。

**创建Znodes**

用给定的路径创建一个znode。flag参数指定创建的znode是临时的，持久的还是顺序的。默认情况下，所有znode都是持久的。  
当会话过期或客户端断开连接时，**临时节点**（flag：-e）将被自动删除。  
**顺序节点**保证znode路径将是唯一的。  
ZooKeeper集合将向znode路径填充10位序列号。例如，znode路径 /myapp 将转换为/myapp0000000001，下一个序列号将为/myapp0000000002。如果没有指定flag，则znode被认为是持久的。

**语法**

create /path /data

**示例**

create /FirstZnode “Myfirstzookeeper-app"

**输出**

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

Created /FirstZnode

要创建**顺序节点**，请添加flag：**-s**，如下所示。

**语法**

create -s /path /data

**示例**

create -s /FirstZnode second-data

**输出**

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

Created /FirstZnode0000000023

要创建**临时节点**，请添加flag：**-e**，如下所示。

**语法**

create -e /path /data

**示例**

create -e /SecondZnode “Ephemeral-data"

**输出**

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

Created /SecondZnode

记住当客户端断开连接时，临时节点将被删除。你可以通过退出ZooKeeper CLI，然后重新打开CLI来尝试。

**获取数据**

它返回znode的关联数据和指定znode的元数据。你将获得信息，例如上次修改数据的时间，修改的位置以及数据的相关信息。此CLI还用于分配监视器以显示数据相关的通知。

**语法**

get /path

**示例**

get /FirstZnode

**输出**

[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

要访问顺序节点，必须输入znode的完整路径。

**示例**

get /FirstZnode0000000023

**输出**

[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（监视）**

当指定的znode或znode的子数据更改时，监视器会显示通知。你只能在**get**命令中设置**watch**。

**语法**

get /path [watch] 1

**示例**

get /FirstZnode 1

**输出**

[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

输出类似于普通的**get**命令，但它会等待后台等待znode更改。<从这里开始>

**设置数据**

设置指定znode的数据。完成此设置操作后，你可以使用**get**CLI命令检查数据。

**语法**

set /path /data

**示例**

set /SecondZnode Data-updated

**输出**

[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

如果你在**get**命令中分配了**watch**选项（如上一个命令），则输出将类似如下所示。

**输出**

[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

**创建子项/子节点**

创建子节点类似于创建新的znode。唯一的区别是，子znode的路径也将具有父路径。

**语法**

create /parent/path/subnode/path /data

**示例**

create /FirstZnode/Child1 firstchildren

**输出**

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

created /FirstZnode/Child1

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

created /FirstZnode/Child2

**列出子项**

此命令用于列出和显示znode的子项。

**语法**

ls /path

**示例**

ls /MyFirstZnode

**输出**

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

[mysecondsubnode, myfirstsubnode]

**检查状态**

状态描述指定的znode的元数据。它包含时间戳，版本号，ACL，数据长度和子znode等细项。

**语法**

stat /path

**示例**

stat /FirstZnode

**输出**

[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

**移除Znode**

移除指定的znode并递归其所有子节点。只有在这样的znode可用的情况下才会发生。

**语法**

rmr /path

**示例**

rmr /FirstZnode

**输出**

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

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

Node does not exist: /FirstZnode

删除**(delete/path)**命令类似于**remove**命令，除了它只适用于没有子节点的znode。