Zookeeper的基本命令：

1. 启动Zookeeper：**zkServer.sh start**(在多台集群里分别运行)
2. 在其中一台机器执行客户端脚本（连接客户端）：**zkCli.sh -server 192.168.201.128:2181** （默认是用2181做为端口）

**一、Zookeeper客户端（Client）源代码阅读**

**一共涉及以下类（按照访问流程）：**

1. **org.apache.zookeeper.ZookeeperMain:**

**zkCli.sh 的客户端入口函数。**

**执行流程：**

1. **public ZooKeeperMain(String[] args){}**

**解析参数 args,并且调用连接函数。**

1. **connectToZK(this.cl.getOption("server"));**

**首先关闭现有的ZK实例，并且创建zookeeper实例**

**Zookeeper.Class**

**public static enum States**

**{**

**CONNECTING, ASSOCIATING, CONNECTED, CONNECTEDREADONLY, CLOSED, AUTH\_FAILED, NOT\_CONNECTED;**

**private States() {}**

**public boolean isAlive()**

**{**

**return (this != CLOSED) && (this != AUTH\_FAILED);**

**}**

**public boolean isConnected()**

**{**

**return (this == CONNECTED) || (this == CONNECTEDREADONLY);**

**}**

**}**

**Watcher.Class**

**public abstract interface Watcher**

**{**

**public abstract void process(WatchedEvent paramWatchedEvent);**

**public static abstract interface Event**

**{**

**public static enum KeeperState**

**{**

**Unknown(-1), Disconnected(0), NoSyncConnected(1), SyncConnected(3), AuthFailed(4), ConnectedReadOnly(5), SaslAuthenticated(6), Expired(-112);**

**private final int intValue;**

**private KeeperState(int intValue)**

**{**

**this.intValue = intValue;**

**}**

**public int getIntValue()**

**{**

**return this.intValue;**

**}**

**public static KeeperState fromInt(int intValue)**

**{**

**switch (intValue)**

**{**

**case -1:**

**return Unknown;**

**case 0:**

**return Disconnected;**

**case 1:**

**return NoSyncConnected;**

**case 3:**

**return SyncConnected;**

**case 4:**

**return AuthFailed;**

**case 5:**

**return ConnectedReadOnly;**

**case 6:**

**return SaslAuthenticated;**

**case -112:**

**return Expired;**

**}**

**throw new RuntimeException("Invalid integer value for conversion to KeeperState");**

**}**

**}**

**public static enum EventType**

**{**

**None(-1), NodeCreated(1), NodeDeleted(2), NodeDataChanged(3), NodeChildrenChanged(4);**

**private final int intValue;**

**private EventType(int intValue)**

**{**

**this.intValue = intValue;**

**}**

**public int getIntValue()**

**{**

**return this.intValue;**

**}**

**public static EventType fromInt(int intValue)**

**{**

**switch (intValue)**

**{**

**case -1:**

**return None;**

**case 1:**

**return NodeCreated;**

**case 2:**

**return NodeDeleted;**

**case 3:**

**return NodeDataChanged;**

**case 4:**

**return NodeChildrenChanged;**

**}**

**throw new RuntimeException("Invalid integer value for conversion to EventType");**

**}**

**}**

**}**

**}**