



Zookeeper分布式系统开发实战 第1课

【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

- Dataguru (炼数成金) 是专业数据分析网站 , 提供教育 , 媒体 , 内容 , 社区 , 出版 , 数据分析业务等服务。我们的课程采用新兴的互联网教育形式 , 独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围 , 重竞争压力的特点 , 同时又发挥互联网的威力打破时空限制 , 把天南地北志同道合的朋友组织在一起交流学习 , 使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本 , 直线下降至百元范围 , 造福大众。我们的目标是 : 低成本传播高价值知识 , 构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情 , 请看我们的培训网站 <http://edu.dataguru.cn>

第一讲 zookeeper介绍

- Zookeeper是一个高性能，分布式的，开源分布式应用协调服务。它提供了简单原始的功能，分布式应用可以基于它实现更高级 的服务，比如同步，配置管理，集群管理，名空间。它被设计为易于编程，使用文件系统目录树作为数据模型。服务端跑在java上，提供java和C的客户端 API

什么是分布式？

什么是协调服务？

第一讲 zookeeper介绍---分布式系统

- A distributed system consists of multiple computers that communicate through a computer network and interact with each other to achieve a common goal

- 多台计算机构成
- 计算机之间通过网络进行通信
- 彼此进行交互
- 共同目标

来自wikipedia

分布式系统存在什么问题？



第一讲 zookeeper介绍---分布式系统的问题



第一讲 zookeeper介绍---协调者



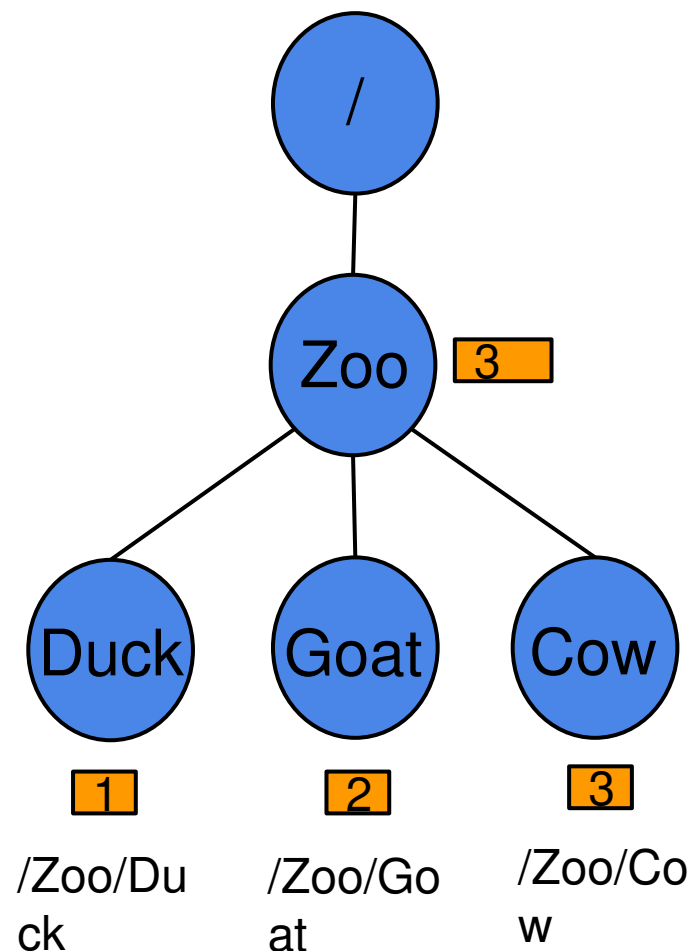
第一讲 zookeeper介绍—分布式系统协调

- 协调: 多个节点一起完成的一个动作 (An act that multiple nodes must perform together)
- 举例:
 - 集群成员管理 (Group membership)
 - 锁 (Locking)
 - 选主 (Leader Election)
 - 同步 (Synchronization)
 - 发布/订阅 (Publisher/Subscriber)

多个节点协调工作是非常困难的!

第一讲 zookeeper介绍---数据模型

- 分层结构
- 树形结构中的每个节点叫做Znode
- 每个Znode都有数据（byte[]类型），也可以有子节点
- 节点路径
 - 斜线分隔：/Zoo/Duck
 - 没有相对路径
- 通过数据结构stat来存储数据的变化 ACL的变化和时间戳
- 数据发生变化时，版本号会递增
- 可以对Znode中的数据进行读写操作

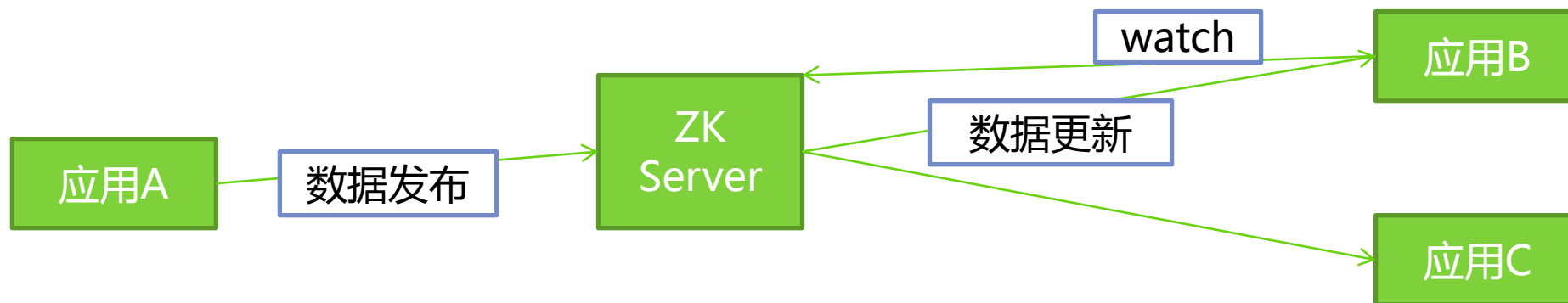


第一讲 zookeeper介绍---典型应用场景

■ 数据发布/订阅

数据发布/订阅即所谓的配置中心：发布者将数据发布到zk

的一个或者一系列节点上，订阅者进行数据订阅，当数据有变化时，可以及时得到数据的变化通知



第一讲 zookeeper介绍---典型应用场景

■ 负载均衡

本质是利用zookeeper的配置管理功能，涉及的步骤为：

- 1.服务提供者把自己的域名及IP 端口的映射注册到zk中
- 2.服务消费者通过域名从zk中获取到对应的IP及端口，这个IP及端口有多个，只是获取其中一个
- 3.当服务提供者宕机时，对应的域名与IP的对应就会减少一个映射
- 4.阿里的dubbo服务框架就是基于zk来实现服务路由和负载

■ 命名服务

在分布式系统中，命名服务（Name Service）也是很重要的应用场景，通过zk也可以实现类似于J2EE中的JNDI的效果；分布式环境下，命名服务更多是资源定位，并不是真正的实体资源，其本质也是用到zk的集中配置管理和查找

■ 分布式协调/通知

- 通过watcher和通知机制实现
- 分布式锁
- 分布式事务

第一讲 zookeeper介绍---典型应用场景

■ 集群管理

- 当前集群中的机器数量
- 集群中机器的运行时状态
- 集群中节点的上下线操作
- 集群节点的统一配置

■ Master选举

- 临时节点
- 顺序节点

第一讲 zookeeper介绍---典型应用场景

- 分布式锁
 - 排他锁
 - 共享锁
- 分布式队列
 - FIFO

第一讲 zookeeper介绍---基本概念

■ 集群角色

- Leader：为客户端提供读和写服务
- Follower：提供读服务，所有写服务都需要转交给Leader角色，参与选举
- Observer：提供读服务，不参与选举过程，一般是为了增强zk集群的读请求并发能力

■ 会话 (session)

- Zk的客户端与zk的服务端之间的连接
- 通过心跳检测保持客户端连接的存活
- 接收来自服务端的watch事件通知
- 可以设置超时时间

■ 数据节点 (Znode)

- 不是机器的意思
- Zk树形结构中的数据节点，用于存储数据
- 持久节点：一旦创建，除非主动调用删除操作，否则一直存储在zk上
- 临时节点：与客户端的会话绑定，一旦客户端会话失效，这个客户端创建的所有临时节点都会被移除
- SEQUENTIAL Znode：创建节点时，如果设置属性SEQUENTIAL，则会自动在节点名后面追加一个整型数字

第一讲 zookeeper介绍---基本概念

■ 版本

- Version : 当前Znode的版本
- Cversion : 当前Znode的子节点版本
- Aversion : 当前Znode的ACL (访问控制) 版本

■ Watcher

- 作用于Znode节点上
- 多种事件通知：数据更新，子节点状态等

第一讲 zookeeper介绍---基本概念

■ ACL

- Access Control Lists
- 类似于linux/unix的权限控制
- **CREATE** : 创建子节点的权限
- READ:获取节点数据和子节点列表的权限
- WRITE : 更新节点数据的权限
- **DELETE** : 删除子节点的权限
- ADMIN : 设置节点ACL的权限

CREATE和DELETE是针对子节点的权限控制

第一讲 zookeeper单机模式

- 在开发测试环境下，没有较多的物理资源，因此采用单机模式；当然，在单台物理机下面也可以部署集群模式，但是这会增大单台物理机的资源消耗；所以，在开发环境我们一般使用单机模式
- 单机模式不可用在生产环境上，因为无论从系统可靠性还是读写性能上都不能满足生产的需求
- Zk是基于java语言开发，因此单机模式的资源需求跟普通的java程序差不多，1个核的CPU和1G内存就足够

第一讲 zookeeper单机模式—安装部署

■ 环境准备

- Jdk1.6及以上
- 机器配置：2核 /2G

■ 安装

- Zk下载地址：<http://www.apache.org/dyn/closer.cgi/zookeeper/>
- 解压文件
 - Windows下采用winzip或者winwar都行
 - Linux下采用tar -zxvf zookeeper-3.4.6.tar.gz
- 配置环境变量
 - ZOOKEEPER_HOME=/home/zookeeper-3.4.6
 - PATH=\$JAVA_HOME/bin:\$ZOOKEEPER_HOME/bin

第一讲 zookeeper单机模式—目录结构

■ bin

- .sh为linux环境
- .cmd为windows环境

■ conf

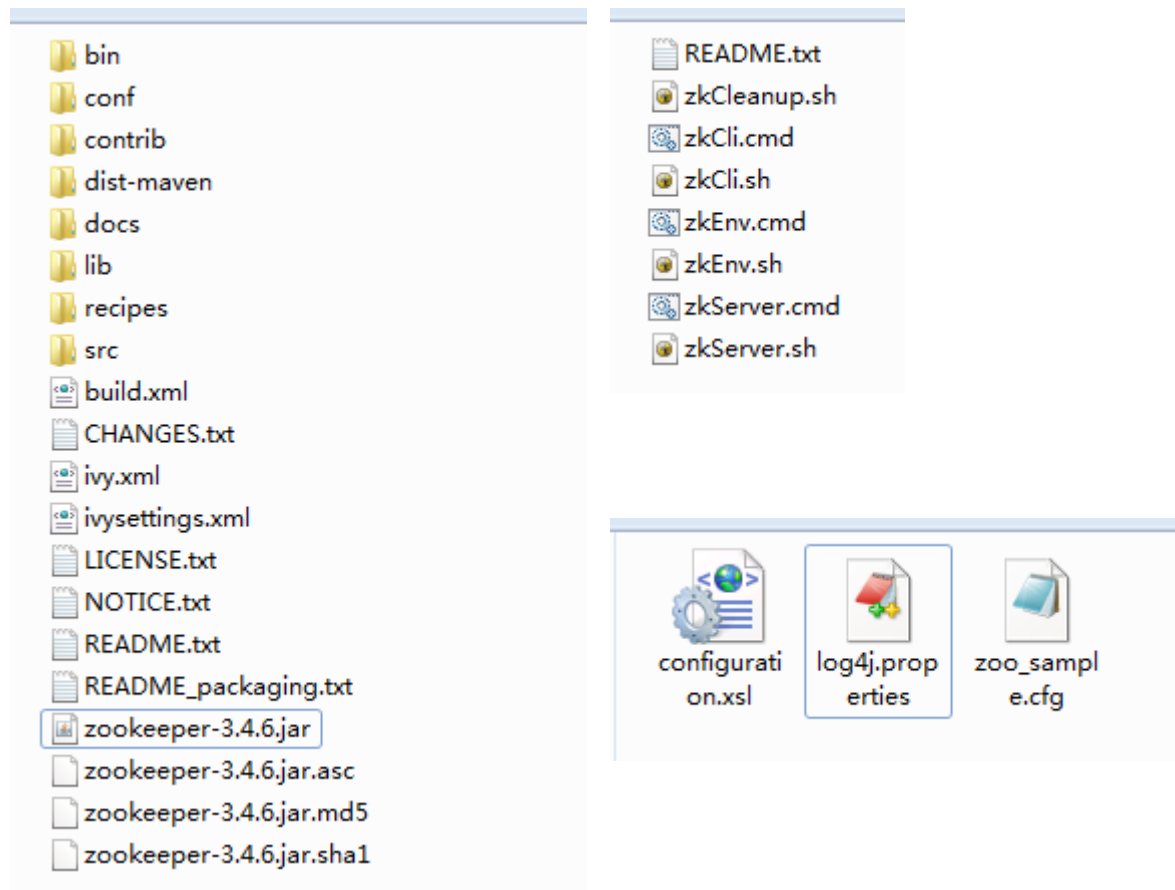
- zoo_sample.cfg为样例配置文件，需要修改为自己的名称，一般为zoo.cfg
- Log4j.properties为日志配置文件

■ Contrib:一些用于操作zk的工具包

■ lib : zk依赖的某些包

■ Recipes : zk某些用法的代码示例

■ Dist-maven : maven编译后的发布目录



第一讲 zookeeper单机模式—运行配置

- 把conf下的zoo_sample.cfg修改为zoo.cfg
- Zoo.cfg
 - tickTime:默认3000ms，作为一种基本单元，可用它的倍数来表示系统内部的时间间隔配置，比如：
 - 2*tickTime是客户端会话的超时时间
 - 1*tickTime是客户端与zk服务器端的心跳时间
 - dataDir:无默认配置，必须配置；用于配置存储快照文件的目录，如果没有配置dataLogDir，那么事务日志也会存储在此目录
 - clientPort：zk的运行端口，默认是2181
- 启动
 - Windows:直接双击zkServer.cmd
 - Linux:zkServer.sh start

第一讲 zookeeper单机模式—运行配置



```
G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin>java "-Dzookeeper.log.dir=G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\.." "-Dzookeeper.root.logger=INFO,CONSOLE" -cp "G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\build\lib\*;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\*;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\conf\zoo.cfg"
2015-11-01 22:38:45,956 [myid:1] - INFO [main:QuorumPeerConfig@103] - Reading configuration from: G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\conf\zoo.cfg
2015-11-01 22:38:45,971 [myid:1] - INFO [main:DatadirCleanupManager@78] - autopurge.snapRetainCount set to 3
2015-11-01 22:38:45,972 [myid:1] - INFO [main:DatadirCleanupManager@79] - autopurge.purgeInterval set to 0
2015-11-01 22:38:45,972 [myid:1] - INFO [main:DatadirCleanupManager@101] - Purge task is not scheduled.
2015-11-01 22:38:45,978 [myid:1] - WARN [main:QuorumPeerMain@113] - Either no config or no quorum defined in config, running in standalone mode
2015-11-01 22:38:46,131 [myid:1] - INFO [main:QuorumPeerConfig@103] - Reading configuration from: G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\conf\zoo.cfg
2015-11-01 22:38:46,132 [myid:1] - INFO [main:ZooKeeperServerMain@95] - Starting server
2015-11-01 22:38:46,182 [myid:1] - INFO [main:Environment@100] - Server environment:zookeeper.version=3.4.6-1569965, built on 02/20/2014 09:09 GMT
2015-11-01 22:38:46,183 [myid:1] - INFO [main:Environment@100] - Server environment:host.name=2012-20140406IS
2015-11-01 22:38:46,183 [myid:1] - INFO [main:Environment@100] - Server environment:java.version=1.7.0_67
2015-11-01 22:38:46,184 [myid:1] - INFO [main:Environment@100] - Server environment:java.vendor=Oracle Corporation
2015-11-01 22:38:46,184 [myid:1] - INFO [main:Environment@100] - Server environment:java.home=D:\Program Files\Java\jdk1.7.0_67\jre
2015-11-01 22:38:46,185 [myid:1] - INFO [main:Environment@100] - Server environment:java.class.path=G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\build\classes;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\zookeeper-3.4.6.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\jline-0.9.94.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\netty-3.7.0.Final.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\slf4j-api-1.6.1.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\slf4j-log4j12.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\conf
2015-11-01 22:38:46,190 [myid:1] - INFO [main:Environment@100] - Server environment:java.library.path=D:\Program Files\Java\jdk1.7.0_67\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Program Files (x86)\Windows7Master;D:\mysql-5.6.16-winx64\bin;D:\Program Files (x86)\MySQL\MySQL Utilities\bin;C:\Program Files\TortoiseSVN\bin;.;
2015-11-01 22:38:46,191 [myid:1] - INFO [main:Environment@100] - Server environment:java.io.tmpdir=C:\Users\ADMINI~1\AppData\Local\Temp\
2015-11-01 22:38:46,192 [myid:1] - INFO [main:Environment@100] - Server environment:java.compiler=<NA>
2015-11-01 22:38:46,196 [myid:1] - INFO [main:Environment@100] - Server environment:os.name=Windows 7
2015-11-01 22:38:46,197 [myid:1] - INFO [main:Environment@100] - Server environment:os.arch=amd64
2015-11-01 22:38:46,197 [myid:1] - INFO [main:Environment@100] - Server environment:os.version=6.1
2015-11-01 22:38:46,198 [myid:1] - INFO [main:Environment@100] - Server environment:user.name=Administrator
2015-11-01 22:38:46,198 [myid:1] - INFO [main:Environment@100] - Server environment:user.home=C:\Users\Administrator
2015-11-01 22:38:46,199 [myid:1] - INFO [main:Environment@100] - Server environment:user.dir=G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin
2015-11-01 22:38:46,219 [myid:1] - INFO [main:ZooKeeperServer@755] - tickTime set to 2000
2015-11-01 22:38:46,220 [myid:1] - INFO [main:ZooKeeperServer@764] - minSessionTimeout set to -1
2015-11-01 22:38:46,221 [myid:1] - INFO [main:ZooKeeperServer@773] - maxSessionTimeout set to -1
2015-11-01 22:38:46,285 [myid:1] - INFO [main:NIOServerCnxnFactory@94] - binding to port 0.0.0.0/0.0.0.0:2181
```

第一讲 zookeeper单机模式—客户端连接



客户端命令行

- Linux : bin/zkCli.sh
- Windows : bin/zkCli.cmd
- 不带任何参数默认连接到localhost : 2181
- zkCli.cmd -server ip:port连接到指定的服务器地址

```
zookeeper-3.4.6\zookeeper-3.4.6\bin\..\zookeeper-3.4.6.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\jline-0.9.94.jar;G:\开发软件\zookeeper-3.4.6\zooke
-3.4.6\bin\..\lib\netty-3.7.0.Final.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\..\lib\slf4j-api-1.6.1.jar;G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin\
n\..\conf
2015-11-01 23:18:30.204 [myid:1 - INFO [main:Environment@100] - Client environment:java.library.path=D:\Program Files\Java\jdk1.7.0_67\bin;C:\Windows\Sun\Java\bin;
va\jdk1.7.0_67\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Windows7Master;D:\mysq
les (x86)\MySQL\MySQL Utilities\;D:\Program Files (x86)\MySQL\MySQL Utilities\Doctrine extensions for PHP\;C:\Program Files\TortoiseSUN\bin;.;.
2015-11-01 23:18:30.205 [myid:1 - INFO [main:Environment@100] - Client environment:java.io.tmpdir=C:\Users\ADMINI~1\AppData\Local\Temp\
2015-11-01 23:18:30.205 [myid:1 - INFO [main:Environment@100] - Client environment:java.compiler=<NA>
2015-11-01 23:18:30.206 [myid:1 - INFO [main:Environment@100] - Client environment:os.name=Windows 7
2015-11-01 23:18:30.207 [myid:1 - INFO [main:Environment@100] - Client environment:os.arch=amd64
2015-11-01 23:18:30.207 [myid:1 - INFO [main:Environment@100] - Client environment:os.version=6.1
2015-11-01 23:18:30.208 [myid:1 - INFO [main:Environment@100] - Client environment:user.name=Administrator
2015-11-01 23:18:30.208 [myid:1 - INFO [main:Environment@100] - Client environment:user.home=C:\Users\Administrator
2015-11-01 23:18:30.209 [myid:1 - INFO [main:Environment@100] - Client environment:user.dir=G:\开发软件\zookeeper-3.4.6\zookeeper-3.4.6\bin
2015-11-01 23:18:30.214 [myid:1 - INFO [main:ZooKeeper@438] - Initiating client connection, connectString=localhost:2181 sessionTimeout=30000 watcher=org.apache.zo
Welcome to ZooKeeper!
2015-11-01 23:18:30.314 [myid:1 - INFO [main-SendThread<127.0.0.1:2181>:ClientCnxn$SendThread@975] - Opening socket connection to server 127.0.0.1/127.0.0.1:2181.
2015-11-01 23:18:30.319 [myid:1 - INFO [main-SendThread<127.0.0.1:2181>:ClientCnxn$SendThread@852] - Socket connection established to 127.0.0.1/127.0.0.1:2181, ini
JLine support is enabled
[zk: localhost:2181(CONNECTING) 0]
[zk: localhost:2181(CONNECTING) 0] 2015-11-01 23:18:33.112 [myid:1 - INFO [main-SendThread<127.0.0.1:2181>:ClientCnxn$SendThread@1235] - Session establishment comp
tiated timeout = 30000
WATCHER::
WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] _
```

连接成功

第一讲 zookeeper单机模式—客户端命令



■ 查看命令行帮助

- 命令行输入help可以查看所有的命令列表及语法
- 实际上，输入任何字符回车都会提示帮助信息

```
[zk: localhost:2181<CONNECTED> 231] d
ZooKeeper -server host:port cmd args
connect host:port
get path [watch]
ls path [watch]
set path data [version]
rmr path
delquota [-n|-b] path
quit
printwatches on|off
create [-s] [-e] path data acl
stat path [watch]
close
ls2 path [watch]
history
listquota path
setAcl path acl
getAcl path
sync path
redo cmdno
addauth scheme auth
delete path [version]
setquota -n|-b val path
```

第一讲 zookeeper单机模式—客户端命令

■ 读取

— ls path [watch]

- Path表示指定数据节点的节点路径
- 列出指定节点下的所有子节点
- 只能查看第一级的所有子节点
- 刚安装时 ls / 下只有默认的zookeeper保留节点
- Watch表示监听path的子节点的变化

```
[zk: localhost:2181(CONNECTED) 1]
[zk: localhost:2181(CONNECTED) 1] ls /
[zk: localhost:2181(CONNECTED) 2] ls / watch
[zk-book1, zk-book, zookeeper]
[zk: localhost:2181(CONNECTED) 3] 带watch

[zk: localhost:2181(CONNECTED) 2] create /zk-book2 123
Created /zk-book2 创建了一个新的节点
[zk: localhost:2181(CONNECTED) 3]
```

```
[zk: localhost:2181(CONNECTED) 3]
WATCHER::
WatchedEvent state:SyncConnected type:NodeChildrenChanged path: /
```

本客户端收到子节点变更通知

命令行输入**help**可以查看所有的命令列表及语法格式

第一讲 zookeeper单机模式—客户端命令

■ 创建

- create [-s] [-e] path data acl
 - 创建zookeeper节点
 - -s或者-e表示创建的是顺序或临时节点，不加默认创建的是持久节点
 - Path为节点的全路径，没有相对节点的表示方式
 - Data为当前节点内存储的数据
 - Acl 用来进行权限控制，缺省情况下不做任何权限控制

```
[zk: localhost:2181(CONNECTED) 3] ls /  
[zk-book2, zk-book, zookeeper]  
[zk: localhost:2181(CONNECTED) 4] create /zk-createtest 456  
Created /zk-createtest  
[zk: localhost:2181(CONNECTED) 5] ls /  
[zk-book2, zk-book, zookeeper, zk-createtest]  
[zk: localhost:2181(CONNECTED) 6]
```

第一讲 zookeeper单机模式—客户端命令

■ 读取

— get path [watch]

- 获取指定节点的数据内容和属性信息
- Path表示指定数据节点的节点路径
- 右图中
 - 123为节点内的数据
 - cZxid为创建该节点的事务id
 - Mzxid最后一次更新该节点的事务id
 - Mime为最后一次更新时间
 - Cversion子节点版本
 - dataVersion数据版本
 - aclVersion访问控制版本

```
MatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] ls /
[zk-book1, zk-book2, zk-book, zookeeper]
[zk: localhost:2181(CONNECTED) 1] get /zk-book1
123
cZxid = 0x6
ctime = Sun Nov 01 23:31:51 CST 2015
mZxid = 0x6
mtime = Sun Nov 01 23:31:51 CST 2015
pZxid = 0x6
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3
numChildren = 0
```

第一讲 zookeeper单机模式—客户端命令

■ 更新

— set path data [version]

- 更新指定节点的数据内容
- Path表示被更新的节点路径
- data为更新的数据
- Version为指定被更新的数据版本，一般不指定，如果数据版本已经更新，则指定旧版本时会报错

```
numChildren = 0
[zk: localhost:2181(CONNECTED) 2] set /zk-book1 456
cZxid = 0x6
ctime = Sun Nov 01 23:31:51 CST 2015
mZxid = 0xb → 发生变化
mtime = Wed Nov 04 23:41:32 CST 2015
pZxid = 0x6
cversion = 0
dataVersion = 1 → 由0变为1
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3
numChildren = 0
[zk: localhost:2181(CONNECTED) 3]
```

```
[zk: localhost:2181(CONNECTED) 17] get /zk-book1
456xiugai2
cZxid = 0x6
ctime = Sun Nov 01 23:31:51 CST 2015
mZxid = 0x10
mtime = Wed Nov 04 23:49:07 CST 2015
pZxid = 0x6
cversion = 0
dataVersion = 3 → 当前版本为3
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 10
numChildren = 0
[zk: localhost:2181(CONNECTED) 18] set /zk-book1 456xiugai2 2
version No is not valid : /zk-book1
[zk: localhost:2181(CONNECTED) 19]
```

更新2版本数据

第一讲 zookeeper单机模式—客户端命令



■ 删除

— delete path [version]

- 删除指定节点
- Path表示被删除的节点
- Version为指定被删除的数据版本，一般不指定，如果数据版本已经更新，则指定旧版本时会报错

```
[zk: localhost:2181(CONNECTED) 40] delete /zk-book1 2
version No is not valid : /zk-book1
[zk: localhost:2181(CONNECTED) 41]
```

删除旧版本

```
[zk: localhost:2181(CONNECTED) 39] get /zk-book1
456xiugai2
cZxid = 0x6
ctime = Sun Nov 01 23:31:51 CST 2015
mZxid = 0x10
mtime = Wed Nov 04 23:49:07 CST 2015
pZxid = 0x6
cversion = 0
dataVersion = 3
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 10
numChildren = 0
[zk: localhost:2181(CONNECTED) 40] delete /zk-book1 2
version No is not valid : /zk-book1
[zk: localhost:2181(CONNECTED) 41] delete /zk-book1 3
[zk: localhost:2181(CONNECTED) 42] get /zk-book1
Node does not exist: /zk-book1
[zk: localhost:2181(CONNECTED) 43]
```

删除之前

删除以后

Thanks

FAQ时间