

阿里云 表格存储Tablestore

开发工具

文档版本：20200609

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务时间约十分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意： 权重设置为0，该服务器不会再接受新请求。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击 设置 > 网络 > 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
[]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ }或者[a b]	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

- 法律声明..... I
- 通用约定..... I
- 1 客户端工具..... 1
- 2 表格存储命令行工具Tablestore CLI..... 2

1 客户端工具

本文主要为您介绍表格存储客户端管理工具。

简介

表格存储客户端管理工具在本地安装之后，提供图形化的操作界面，支持以下操作：

- 创建、更新和删除数据表
- 写入、更新、读取和删除数据

安装客户端

- [控制台客户端 Windows 版](#)
- [控制台客户端 Mac 版](#)

更多云上软件尽在[云市场](#)。

登录客户端

1. 双击打开客户端。
2. 输入登录信息。



说明：

AccessKeyID 和 AccessKeySecret 在控制台右上角的[账号](#) > **accesskeys**中获取。

创建数据表

以Windows客户端为例。

1. 在左侧导航栏，单击**新建**。
2. 输入数据表的各项参数值。
3. 单击**确认创建**。

2 表格存储命令行工具Tablestore CLI

Tablestore CLI提供简洁、方便的管理命令，支持Windows、Linux、Mac平台。本文介绍如何使用Tablestore CLI以命令行的方式管理表格存储的数据。

下载

Tablestore CLI的运行平台以及对应的下载地址请参见下表。

平台	下载地址
Windows	Windows10
Linux	<ul style="list-style-type: none">Linux (AMD64)Linux (ARM64)
Mac	Mac

快速开始

使用Tablestore CLI操作表格存储中的数据。

```
#1 启动命令行工具。
$./ts

#2 配置对应的接入信息，请根据实际情况修改对应的字段。
ts> config --endpoint https://myinstance.cn-hangzhou.ots.aliyuncs.com --instance
myinstance --id test_accessid --key test_accesskey

#3 查看当前实例下的所有表。
ts> lt

#4 创建一张数据表。
ts> create -t mysampletable --pk '[{"c":"uid","t":"string"}, {"c":"pid","t":"integer"}]' --ttl
864000 --version 1

'''
表名: mysampletable
指定两个主键列: uid (string类型) 和pid (integer类型)
数据生命周期: 864000秒 (10天)
最大版本数: 1
如不指定数据生命周期和最大版本数，则数据生命周期默认为永不过期，最大版本数为1。
'''

#5 查看表信息。
ts> desc -t mysampletable

#6 使用表。
ts> use -t mysampletable

#7 插入数据。
ts> put --pk ['"86", 6771]' --attr '[{"c":"name", "v":"redchen1"}, {"c":"country", "v":"china"}]'
ts> put --pk ['"86", 6772]' --attr '[{"c":"name", "v":"redchen2"}, {"c":"country", "v":"china"}]'
ts> put --pk ['"86", 6773]' --attr '[{"c":"name", "v":"redchen3"}, {"c":"country", "v":"china"}]'
ts> put --pk ['"86", 6774]' --attr '[{"c":"name", "v":"redchen4"}, {"c":"country", "v":"china"}]'
```

```
#8 读取数据。
ts> get --pk ['"86",6771]'
ts> get --pk ['"86",6772]'
ts> get --pk ['"86",6773]'
ts> get --pk ['"86",6774]'
```

```
#9 删除表。
ts> drop -t mysampletable -y
```

查看选项

通过help选项来查看Tablestore CLI支持的所有option。

```
$. /ts help

Commands:
alter          Alter table 更新表信息
clear         Clear the screen
config        Config the TableStore access information 配置访问Tablestore的基本信息
create        Creat a new table 创建表
del           Delete the specify row from TableStore 删除一行数据
desc          Show table meta 查询表描述信息
drop          Drop the table 删除表
exit          Exit the program
export        Export the data of table to disk from TableStore, not support multi
version 表数据导出磁盘文件, 不支持数据多版本
get           Get specify row from TableStore 读取一行数据
help          Display help
import        Load the data to TableStore, not support multi version 磁盘文件数据导入
表
list          List all tables 列出表名称
points        Logically divide the data of the full table into several shards close to the
specified size
press_check   Check data for press 检查压测状态
press_input   Input data for press 开启压测
put           Insert a row to TableStore 插入一行数据
quit          Quit the program
update        Insert a row to TableStore 更新一行数据
use           Select table 选择进行数据操作的表
```

初始化

配置对应的接入信息，请根据实际情况修改对应的字段。

```
Sample:
config --endpoint https://myinstance.cn-hangzhou.ots.aliyuncs.com --instance
myinstance --id test_accessid --key test_accesskey
Flags:
--endpoint Endpoint    TableStore Endpoint
--id AccessKeyId       User AccessKeyId
--instance Instance    TableStore Instance
--key AccessKeySecret  User AccessKeySecret
```

表操作

通过Tablestore CLI可以对表进行的操作，包括创建表、选择进行操作的表，列出表名称，更新表，查询表描述信息和删除表。

- 创建表

```
Sample:
#创建表, 第一个主键string类型, 第二个主键integer类型。
#第二个主键integer类型, 并且设置为自增列。
#设置数据过期时间和最大版本数。
create -t mytable --pk '{"c":"uid", "t":"string"}, {"c":"pid", "t":"integer"}'
create -t mytable --pk '{"c":"uid", "t":"string"}, {"c":"pid", "t":"integer", "opt":"auto"}'
create -t mytable --pk '{"C":"uid","t":"string"}, {"c":"pid","t":"integer"}' --ttl 864000 --
version 1
Flags:
-i, --input string      Create table json 支持从配置文件建表
-p, --pk string         Primary Key
--read_cu string        read capacity unit (default "0") 读预留CU
-t, --table string      table name
--ttl string            Time to live, unit is second, default(-1), means never timeout. (
default "-1") 数据过期时间, 秒为单位, (默认'-1', 永不过期)
--version string        Max version, default(1) (default "1") 数据最大版本数
--write_cu string       write capacity unit (default "0") 写预留CU
```

- 选择进行数据操作的表

```
Sample:
use -t tablename
Flags:
-t, --table string      table name
```

- 列出表名称

```
./ts list
List all table name from instance 列出实例下所有表的名称
```

- 更新表

```
Sample:
alter -t mytable --ttl 86400 --version 1 --read_cu 0 --write_cu 0
Flags:
--read_cu string        read capacity unit
-t, --table string      table name
--ttl string            Time to live, unit is second
--version string        Max version
--write_cu string       write capacity unit
```

- 查询表描述信息

```
Sample:
desc -t tablename
Flags:
-o, --output string     Output file path
-t, --table string      Table name
```

- 删除表

```
Sample:
drop -t tablename
Flags:
-t, --table string      table name
```



```
-y, --yes      confirm yes
```

单行数据操作

通过Tablestore CLI可以进行的单行数据操作，包括插入一行数据，读取一行数据，更新一行数据和删除一行数据。

- 插入一行数据

```
Sample:
#插入一行数据，第一主键为“86”，第二主键为6771，包含两列，其中第一列列名“
name”列值“redchen”。
#选择插入数据时的检查逻辑，1.忽略检查 2.存在相同主键的行允许插入 3.不存在相同主键
的行允许插入。
#支持指定时间戳插入。
#主键中有自增列的插入。
#从文件读取数据并插入。
put --pk '["86", 6771]' --attr '{"c":"name", "v":"redchen"}, {"c":"country", "v":"china"}'
put --pk '["86", 6771]' --attr '{"c":"name", "v":"redchen"}, {"c":"country", "v":"china"}'
--condition ignore/exist/not_exist
put --pk '["86", 6771]' --attr '{"c":"name", "v":"redchen"}, {"c":"country", "v":"china",
"ts":15327798534}'
put --pk '["86", null]' --attr '{"c":"name", "v":"redchen"}, {"c":"country", "v":"china"}'
put -i /tmp/input
Flags:
--attr string      Attribute column value
--condition string input condition, ignore/exist/not_exist (default "ignore")
-i, --input string  input file
--pk string        PrimaryKey value
```

- 读取一行数据

```
Sample:
get --pk '["86",6771]'
Flags:
-o, --output string Output file path
--pk string          Input primary key column value
```

- 更新一行数据

```
Sample:
update --pk '["86", 6771]' --attr '{"c":"name", "v":"redchen"}, {"c":"country", "v":"
china"}'
update --pk '["86", 6771]' --attr '{"c":"name", "v":"redchen"}, {"c":"country", "v":"
china"}' --condition ignore/exist/not_exist
update -i /tmp/input
Flags:
--attr string      Attribute column value
--condition string input condition, ignore/exist/not_exist (default "ignore")
-i, --input string  input file
--pk string        PrimaryKey value
```

- 删除一行数据

```
Sample:
del --pk '["86", 6771]'
Flags:
```

```
--pk string PrimaryKey value
```

简单压测操作

通过Tablestore CLI可以进行的简单压测操作，包括开启压测和检查压测状态。

- 开启压测

```
Sample:
#创建一张表, 第一主键为“分区”, 第二主键为“行序号”。
#使用该表, 向分区“redchen”插入10条数据。
create -t mytable --pk '{"c":"uid", "t":"string"}, {"c":"pid", "t":"integer"}'
use -t mytable
press_input --part redchen --count 10
Flags:
--begin string Begin index, default 0 从指定行序号开始插入
--count string Input count, length of single row is 1KB 插入的行数, 每行数据长度为1KB
--part string Partition key, default 'redchen' (default "redchen")
```

- 检查压测状态

```
Sample:
create -t mytable --pk '{"c":"uid", "t":"string"}, {"c":"pid", "t":"integer"}'
use -t mytable
press_check --part redchen --begin 0 --count 1000
Flags:
--begin string Begin index, default 0
--count string Check count
--part string Partition key
-y, --yes show cost time
```

数据备份操作

通过Tablestore CLI可以进行的数据备份操作，包括表数据导出磁盘文件和磁盘文件数据导入表。

- 表数据导出磁盘文件

```
Sample:
#导出当前表的全部数据到磁盘文件。
#导出指定列数据。
export -o /tmp/mydata
export -o /tmp/mydata -c attr1,attr2,attr3
Flags:
-c, --columns string Column names
-o, --output string Output file path
```

- 磁盘文件数据导入表

```
Sample:
#读取磁盘文件数据导入表。
#忽略时间戳检查, 使用当前时间。
import -i /tmp/mydata
import -i /tmp/mydata --ignore_ts
'''
file format:
{"PK":{"Values":["redchen",0]}, "Attr":{"Values":{"C":"country", "V":"china0"}, {"C":"name", "V":"redchen0}}}
```

```
 {"PK":{"Values":["redchen",1]},"Attr":{"Values":{"C":"country","V":"china1"},"C":"name", "V":"redchen1"}}}
```

Flags:

--ignore_ts ignore timestamp
-i, --input string Input file name