

游戏数据库 TcaplusDB

快速入门

产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

快速入门

注册游戏

开通服务

批量加表

获取访问点信息

访问 TcaplusDB 表

通过 RESTful API 读写

通过 client 工具读写

快速入门 注册游戏

最近更新时间：2019-01-03 10:54:36

操作场景

本文指导您如何申请开通游戏数据库 TcaplusDB，并添加游戏获取游戏 ID 和游戏 KEY。

前提条件

您已注册腾讯云账号并认证成功。

如未注册腾讯云账号请参考 [注册腾讯云](#) 文档，并参考 [实名认证指引](#) 文档进行实名认证。

操作步骤

1. 申请游戏数据库 TcaplusDB（游戏存储）服务

在腾讯云 [游戏数据库 TcaplusDB 介绍页](#) 单击【立即申请】游戏数据库 TcaplusDB（游戏存储）的申请表单页，填写对应的申请信息，单击【提交申请】等待审核通过即可。

2. 添加游戏

2.1 进入 [游戏服务控制台](#)，单击【添加游戏】进入填写添加游戏信息界面。



2.2 根据需求填写游戏名称、联系人姓名、手机号码、开发平台和细分品类，单击【确定】，完成添加游戏。

←
添加游戏

所属项目 请选择 ▾

游戏名称 请输入游戏名称

联系人姓名 请输入联系人姓名

手机号码 请输入联系电话

开发平台

PC客户端
 PC网页端
 移动端
 电视端
 主机

细分品类

MOBA
 ROG
 ARPG
 ACG
 休闲

FPS
 TPS
 策略
 VR
 棋牌

体育
 其他

确定

2.3 完成添加游戏后，系统会自动分配游戏 ID 和游戏 KEY，如下图所示，注册游戏完成，下一步即可开通服务。

游戏名称	游戏ID	游戏KEY ⓘ
腾讯云官网	1226	5b23fdb21e
test	5774	b661192961
免费体验专区	9345	466228e169

ⓘ 说明：

如图所示系统自动分配了一个名为“免费体验专区”的游戏项目，开通此项目下的所有服务均是免费的。

开通服务

最近更新时间：2019-08-14 19:38:11

操作场景

本文指导您在添加游戏成功后，如何为您的游戏开通游戏存储服务。

前提条件

您已申请开通游戏数据库 TcaplusDB，并成功添加游戏。

如未添加游戏，请参考 [注册游戏](#) 操作文档完成游戏添加。

操作步骤

1. 通过左边菜单单击【游戏存储】>【表管理】进入到游戏存储服务开通页面。



在表管理页面顶部，可免费开通存储服务，专享1GB存储免费体验资源。

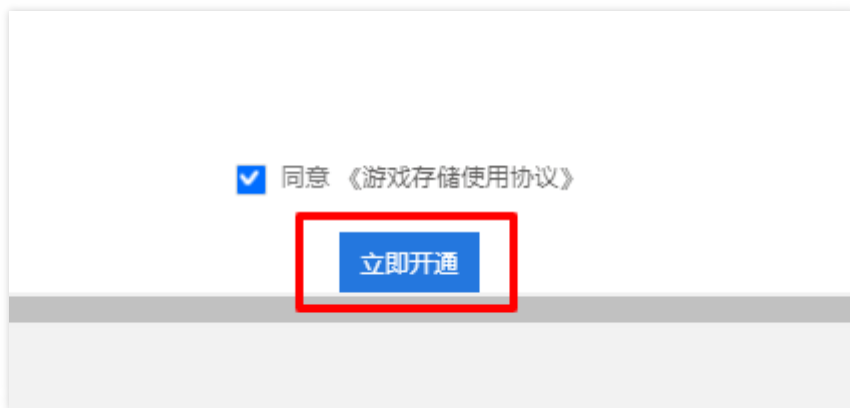


2. 单击【我要开通】进入到开通详情页面。



按照您的需求选择存储产品、地域、网络类型。

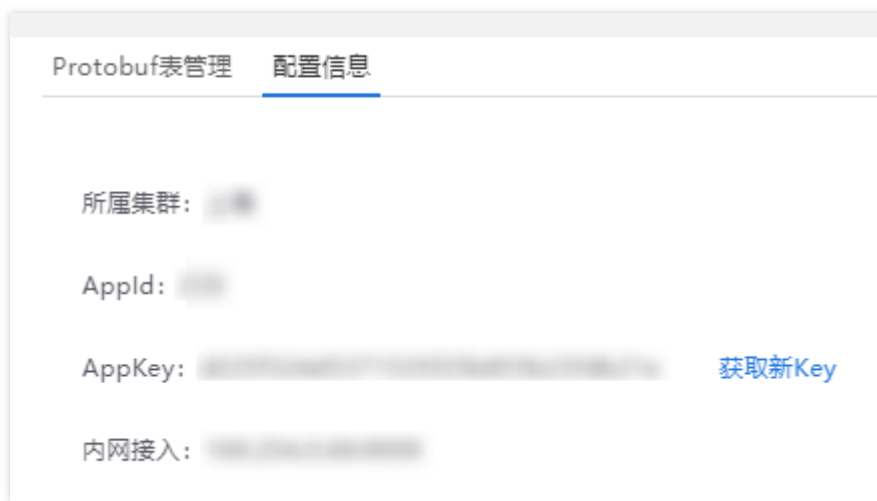
阅读使用协议后勾选同意协议，单击【立即开通】即可成功开通服务。



3. 开通服务后，显示下图界面，单击【配置信息】，可查看配置信息。



4. 查看配置信息，配置信息为系统返回的游戏所属集群、AppId、AppKey 和内网接入地址。开通服务完成，下一步即可进行表的管理。



批量加表

最近更新时间：2019-01-15 16:12:24

操作场景

本文指导您在开通服务成功后，如何为您的游戏创建 TcaplusDB 表。

前提条件

您的游戏已开通游戏存储服务。

如未开通游戏存储服务，请参考 [开通服务](#) 操作文档完成服务开通。

操作步骤

TcaplusDB 表管理常见操作为批量加表、批量改表、批量删表、批量扩缩容和批量回档。

TcaplusDB 支持批量加表操作，操作步骤如下：

1. 单击【批量加表】，进入加表页面。



2. 选择部署单元，若当前没有部署单元则需要新建部署单元，单击【新建部署单元】，按需要填写名称即可。

上一步

下一步

以下是一个示例 proto 文件

```
// tb_online.proto
syntax = "proto2";
package myTcaplusTable;

import "tcapluservice.optionv1.proto";

message tb_online {
  option(tcapluservice.tcaplus_primary_key) = "uin,name,region";

  required int64 uin = 1;
  required string name = 2;
  required int32 region = 3;

  required int32 gamesvrid = 4;
  optional int32 logintime = 5 [default = 1];
}
```

```
repeated int64 lockid = 6 [packed = true];
optional bool is_available = 7 [default = false];
optional pay_info pay = 8;
}

message pay_info {

  required int64 pay_id = 1;
  optional uint64 total_money = 2;
  optional uint64 pay_times = 3;
  optional pay_auth_info auth = 4;

  message pay_auth_info {
    required string pay_keys = 1;
    optional int64 update_time = 2;
  }
}
```

- 单击【本地文件】，选择本地文件上传。若已上传过文件，可单击【从历史文件导入】快速添加文件。上传文件后，单击【下一步】。

⚠ 注意：

文件仅支持 Proto 格式且大小不能超过2M。

1 上传文件 ———— 2 设置表信息 ———— 3 确认表信息

部署单元: × [新建部署单元](#)

添加Protobuf文件 **本地文件** **从历史文件导入**

仅支持Proto格式且文件大小不能超过2M

名称	大小(字节)	操作
test.proto	813	查看详情 删除

4. 设置表信息，选择需要设置的表，根据表需求设置【容量】、【预留读】和【预留写】，单击【下一步】。

1 上传文件 ———— 2 设置表信息 ———— 3 确认表信息

test(1) 已选择 1/1 个表 0 GB 容量 预留读 0 CU 预留写 0 CU 0 元/天

<input checked="" type="checkbox"/> 表名	容量(GB)	预留读(CU)	预留写(CU)	价格
<input checked="" type="checkbox"/> tb_online	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
总计	0GB	0CU	0CU	0元/天

5. 确认添加的表信息，确认无误后，单击【创建】。

1 上传文件 2 设置表信息 3 确认表信息

test(1) 共添加 1 个表 1 GB 容量 预留读 80 CU 预留写 26 CU 1.85 元/天

表名	容量(GB)	预留读 (CU)	预留写 (CU)	价格
tb_online	1	80	26	1.85

总计 1GB 80CU 26CU 1.85元/天

上一步 **创建**

6. 创建表成功，系统返回创建成功提示。

1 上传文件 2 设置表信息 3 确认表信息

test(1) 共添加 1 个表 1 GB 容量 预留读 80 CU 预留写 26 CU 1.85 元/天

表名	容量(GB)	预留读 (CU)	预留写 (CU)	价格
tb_online	1	80	26	1.85

总计 1GB 80CU 26CU 1.85元/天

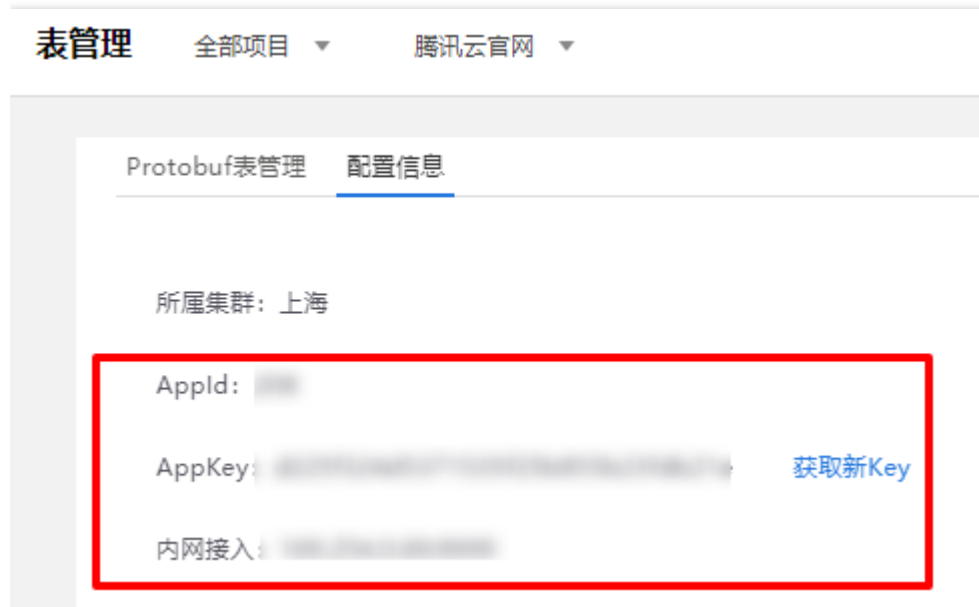
结果: 表名:tb_online,信息:创建表成功!

返回表管理

获取访问点信息

最近更新时间：2019-01-03 10:55:12

当添加表成功后，从【配置信息】选项卡中可以查看游戏访问点信息。



包括以下访问点信息：

- AppId - 应用所属 ID
- AppKey - 应用密码信息
- 内网接入 - 目录服务 Tcapdir 访问点
- RESTful 接入 - 通过 RESTful API (HTTP) 方式访问接入点

访问 TcaplusDB 表 通过 RESTful API 读写

最近更新时间：2019-06-26 17:23:06

本文为您介绍通过 RESTful API 方式访问 TcaplusDB 表的操作。

通过注册游戏、开通服务、添加表步骤后，您就有了属于自己的 TcaplusDB 业务数据表，通过获取到的访问点信息就能够对业务数据表进行读写访问。本示例将分别采用 SetRecord 和 GetRecord 接口对目标表进行写和读操作。

⚠ 注意：

相关操作需要在用户腾讯云账号下申请的 CVM 中进行。

在本例中，假设获取到如下接入点信息，并且在 ZoneId 为1的部署单元中创建了表 tb_online。

- AppId : 2
- AppKey : 3aa84dd773826cd655e9f24a249d68bb
- RESTful 接入点 : 10.123.9.70:31002
- ZoneId : 1

写操作

通过如下 HTTP 请求向表中写入一条数据。对应的请求 url、HTTP 请求头、请求数据和回包如下：

```
HTTP Method: POST
req_url:
http://10.123.9.70:31002/ver1.0/apps/2/zones/1/tables/tb_online/records
HTTP header:
[
"x-tcaplus-target:Tcaplus.SetRecord",
"x-tcaplus-version:Tcaplus3.32.0",
"x-tcaplus-pwd-md5:c3eda5f013f92c81dda7afcdc273cf82",
"x-tcaplus-result-flag:1",
"x-tcaplus-data-version-check:1",
"x-tcaplus-data-version:-1",
"x-tcaplus-idl-type:Protobuf"
]
Data:
{
"ReturnValues": "user define data",
```

```

"Record": {
  "name": "tcaplus_user",
  "lockid": [
    50,
    60,
    70,
    80,
    90,
    100
  ],
  "pay": {
    "pay_times": 184,
    "total_money": 100000,
    "pay_id": 12000,
    "auth": {
      "pay_keys": "njaklhewunfasdjnuonawef",
      "update_time": 1544018596
    }
  },
  "region": 10,
  "uin": 1024,
  "is_available": true,
  "gamesvrid": 5000,
  "logintime": 100
}
}

```

Resp Status 200

cost:9 ms

Response

```

{
  "ReturnValues": "aaaaaaaaa",
  "RecordVersion": 1,
  "ErrorMsg": "Succeed",
  "ErrorCode": 0,
  "Record": {
    "name": "tcaplus_user",
    "lockid": [
      50,
      60,
      70,
      80,
      90,
      100
    ],

```



```
"pay": {
  "pay_times": 184,
  "total_money": 100000,
  "pay_id": 12000,
  "auth": {
    "pay_keys": "njaklhewunfasdjnuonawef",
    "update_time": 1544018596
  }
},
"region": 10,
"uin": 1024,
"is_available": true,
"gamesvrid": 5000,
"logintime": 100
}
}
```

读操作

通过如下 HTTP 请求将上一步写入的记录读取出来。对应的请求 url、HTTP 请求头和回包如下：

```
HTTP Method: GET
req_url:
http://10.123.9.70:31002/ver1.0/apps/2/zones/1/tables/tb_online/records?keys={"region": 10, "name":
"tcaplus_user", "uin": 1024}
HTTP 请求Header:
[
  "x-tcaplus-target:Tcaplus.GetRecord",
  "x-tcaplus-version:Tcaplus3.32.0",
  "x-tcaplus-pwd-md5:c3eda5f013f92c81dda7afc2c273cf82",
  "x-tcaplus-idl-type:protobuf"
]
-----
应答状态 200
Response:
{
  "RecordVersion": 2,
  "Record": {
    "name": "tcaplus_user",
    "lockid": [
      50,
      60,
      70,
```

```
80,  
90,  
100  
],  
"pay": {  
  "pay_times": 184,  
  "total_money": 100000,  
  "pay_id": 12000,  
  "auth": {  
    "pay_keys": "njaklhewunfasdjnuonawef",  
    "update_time": 1544018596  
  }  
},  
"region": 10,  
"uin": 1024,  
"is_available": true,  
"gamesvid": 5000,  
"logintime": 100  
},  
"ErrorMsg": "Succeed",  
"ErrorCode": 0  
}
```

示例程序

以下通过简单的 Python 代码来实现上面所演示的两个功能。更多的表操作请参考 [Tcaplus RESTful API 接口文档](#)。

```
#!/usr/bin/env python  
# -*- coding: utf-8 -*-  
# author : tcaplus  
# created : 2018.11.12  
#####  
import sys  
import time  
import OS  
import json  
import pycurl  
import StringIO  
import hashlib  
from urllib import unquote, quote
```

```
#####
#用户的app和表信息
CGI_URL = "http://xx.xx.xx.xx:xxxxx/ver1.0"
APP_ID_INT = 2
ZONE_ID_INT = 1
GAME_KEY_STR = "xxxxxxxxxxxx"
TABLE_NAME_STR = "tb_online"
#####

def op_get(url, header):
    '''发送HTTP Get请求'''
    ret = None
    curl_fp = None

    try:
        curl = pycurl.Curl()
        curl.setopt(pycurl.HTTPHEADER, header)
        curl.setopt(pycurl.CONNECTTIMEOUT, 60)
        curl.setopt(pycurl.TIMEOUT, 300)
        curl.setopt(pycurl.CUSTOMREQUEST, "GET")
        curl_fp = StringIO.StringIO()
        curl.setopt(pycurl.URL, url.encode('utf-8'))
        curl.setopt(curl.WRITEFUNCTION, curl_fp.write)
        curl.perform()
        ret = json.loads(curl_fp.getvalue())
        print "Resp Status %d" %(curl.getinfo(pycurl.RESPONSE_CODE))
        return ret
    except pycurl.error, err_msg:
        print err_msg
    finally:
        if curl_fp:
            curl_fp.close()
            curl.close()

def op_put(url, header, content):
    '''发送http put请求'''
    ret = None

    try:
        curl = pycurl.Curl()
        curl.setopt(pycurl.HTTPHEADER, header)
        curl.setopt(pycurl.FOLLOWLOCATION, 1)
        curl.setopt(pycurl.MAXREDIRS, 5)
        curl.setopt(pycurl.CONNECTTIMEOUT, 60)
        curl.setopt(pycurl.TIMEOUT, 300)
        curl.setopt(pycurl.HTTPPROXYTUNNEL, 1)
    
```

```
curl_fp = StringIO.StringIO()
curl.setopt(pycurl.CUSTOMREQUEST, "PUT")
curl.setopt(curl.POSTFIELDS, content)
curl.setopt(pycurl.URL, str(url))
curl.setopt(curl.WRITEFUNCTION, curl_fp.write)
curl.perform()

ret = json.loads(curl_fp.getvalue())
print "Resp Status %d" %(curl.getinfo(pycurl.RESPONSE_CODE))
except pycurl.error, err_msg:
print err_msg
finally:
if curl_fp:
    curl_fp.close()
    curl.close()
return ret

def test_GetRecord():
    #填充key字段
    keys = {"uin":1024,
"name":"tcaplus_user",
"region":10}

    #填充 HTTP header
    header = []
    header.append("x-tcaplus-target:Tcaplus.GetRecord")
    header.append("x-tcaplus-version:Tcaplus3.32.0")
    header.append("x-tcaplus-pwd-md5:%s" %(hashlib.md5(GAME_KEY_STR).hexdigest()))
    header.append("x-tcaplus-idl-type:protobuf")

    #设置请求url
    #select属性需要urlencode处理
    req = '%s/apps/%d/zones/%d/tables/%s/records?keys=%s' \
    %(CGI_URL, APP_ID_INT, ZONE_ID_INT, TABLE_NAME_STR, quote(json.dumps(keys)))

    print "req_url:\n%s" %req
    print "HTTP header:"
    print header
    #发送get请求
    ret = op_get(req, header)
    print "Response:"
    print ret

def test_SetRecord():
    #填充 HTTP header
```

```

header = []
header.append("x-tcaplus-target:Tcaplus.SetRecord")
header.append("x-tcaplus-version:Tcaplus3.32.0")
header.append("x-tcaplus-pwd-md5:%s" %(hashlib.md5(GAME_KEY_STR).hexdigest()))
header.append("x-tcaplus-result-flag:1")
header.append("x-tcaplus-data-version-check:1")
header.append("x-tcaplus-data-version:-1")
header.append("x-tcaplus-idl-type:Protobuf")

str_user_buff = "test user buff"
print "user buff len:%s" %(len(str_user_buff))

#需要设置的数据记录
data = {}
data["Record"] = {}
data["Record"]["uin"] = 1024
data["Record"]["name"] = "tcaplus_user"
data["Record"]["region"] = 10
data["Record"]["gamesvrid"] = 5000
data["Record"]["logintime"] = 100
data["Record"]["lockid"] = [50, 60, 70, 80, 90, 100]
data["Record"]["is_available"] = True
data["Record"]["pay"] = {}
data["Record"]["pay"]["pay_id"] = 12000
data["Record"]["pay"]["pay_times"] = 184
data["Record"]["pay"]["total_money"] = 100000
data["Record"]["pay"]["auth"] = {}
data["Record"]["pay"]["auth"]["pay_keys"] = "njaklhewunfasdjnuonawef"
data["Record"]["pay"]["auth"]["update_time"] = 1544018596
data["ReturnValues"] = str_user_buff

#填充请求
req = '%s/apps/%d/zones/%d/tables/%s/records' \
%(CGI_URL, APP_ID_INT, ZONE_ID_INT, TABLE_NAME_STR)
req_data_str = json.dumps(data)
print "req_url:\n %s" %req
print "Http header:"
print header
print "Data:"
print '-' * 40
print data
print '-' * 40
ret = op_put(req, header, req_data_str)
print "Response"
print ret
    
```

```
if __name__ == '__main__':  
    test_SetRecord()  
    print '*' * 40  
    test_GetRecord()
```

通过 client 工具读写

最近更新时间：2019-04-16 12:07:43

以下是通过 client 工具访问 TcaplusDB 表的步骤。

tcaplus_client 是在 TcaplusServiceAPI 发布包 bin 目录中的一个 TcaplusDB 表访问工具，它本身也是 TcaplusServiceAPI 的一个应用程序。

Linux x86_64 平台的 TcaplusServiceAPI 发布包将包含64位 Linux 版本的 tcaplus_client 工具，Windows x64 平台的 TcaplusServiceAPI 发布包将包含64位 Windows 版本的 tcaplus_client 工具，接下来的演示将基于 Linux x86_64 版本的工具进行。

⚠ 注意：

相关操作需要在用户腾讯云账号下申请的 CVM 中进行。

在本例中，假设用户获取到如下接入点信息，并且在 ZoneId 为1的部署单元中创建了表 tb_online

- AppId : 2
- AppKey : 3aa84dd773826cd655e9f24a249d68bb
- tcapdir 接入点 : 10.125.32.21:9999
- ZoneId : 1

权限操作

首先需要给 tcaplus_client 工具赋予可执行权限，当直接执行 ./tcaplus_client 不带任何参数时，会打印连接所需的参数信息，用户可以根据自己的游戏业务进行填写。

```
# ./tcaplus_client
```

```
-----  
invalid parameters, please start the client as following:
```

```
./tcaplus_client -a app_id -z zone_id -s signature -d dir_server_url [-t table_name] [-l log_file.xml] [-T tdr_file.tdr] [-e execute_command]
```

```
the params in [] are optional, and their order is not important.
```

```
-a(--ap_id) App ID
```

```
-z(--zone_id) ZONE ID -s(--signature) PASSWORD
```

```
-d(--dir) dir server addr
```

```
-t(--table) table to add
```

```
-l(--log) log file name that must be client_log.xml, and log class name be client
```

```
-T(--tdr) tdr filename
```

```
-e(--execute) content following should be with quotes.
```

```
e.g. ./tcaplus_client -a 2 -z 3 -s "FE6533875C8385C3" -d 172.25.40.181:9999 -T table_test.tdr
```

连接

使用命令连接 TcaplusDB。

```
./tcaplus_client -a 2 -z 1 -s "3aa84dd773826cd655e9f24a249d68bb" -d 10.125.32.21:9999
```

```
+-----+  
| tcaplus_client x86_64 build at Wed Jan 18 22:08:38 CST 2017 |  
||  
| Welcome! |  
+-----+
```

```
tcaplus>
```

在提示符之后输入 `help`，可以看到进一步的帮助信息，通过 `> help 具体命令` 可以查看具体的使用方法。

```
tcaplus>help
```

```
-----  
show get server status related information. executing help show for details.
```

```
dir add dir server url. if no dir_server_url added ,
```

```
commands will fail when executing.
```

```
desc output current table struction description
```

```
count output table row count
```


select query data

update update record. **if** record does not exist then **add** it **or update** it **if exists**.

insert insert a new record (not implemented)

delete delete record(s)

bson **execute** bson query

dump dump records from tcaplus **to file** with csv format

load load records from csv **file and** import the records **to** tcplus

clean clean table

quit exit the client

help follow a **command to get** details.

e.g. **help show**, **help dir** etc.

note: tdr **mode** means starting client with **-T**, **and** none tdr **mode** starting client without **-T**

接下来将演示如何分别通过 Update 和 Select 命令执行写和读操作。

写操作

使用 update 命令写入记录。

```
tcaplus>update tb_online set gamesvrid=4099, logintime=101 where uin=1024 and name="tcaplus_u  
ser" and region=10;
```

```
-----  
success.  
-----
```

```
update time: 5593 us
```

读操作

使用 select 命令将刚刚写入的数据读出来。

```
tcaplus>select gamesvrid, logintime from tb_online where uin=1024 and name="tcaplus_user" and re  
gion=10;
```

```
+-----+-----+-----+-----+-----+  
| uin | name | region | recDataVersion | gamesvrid | logintime |  
+-----+-----+-----+-----+-----+  
| 1024 | tcaplus_user | 10 | 2 | 4099 | 101 |  
+-----+-----+-----+-----+-----+
```

totally 1 record(s) responded.

query time 8686 us