

# RestFul API 2.0 接口列表

TcaplusDB · 自研 NoSQL 数据库

Exported on 02/07/2021

# Table of Contents

<b>1</b>	<b>RESTful API 2.0 for Generic表</b>	<b>15</b>
1.1	RESTful API 2.0说明	15
1.2	RESTful API 使用前置	15
1.2.1	TcaplusDB环境准备	15
1.2.2	TcaplusDB表创建	15
1.2.3	测试机器准备	15
1.2.4	TcaplusDB样例表	15
1.3	RESTful API 2.0 接口列表	16
1.4	[PB] InsertRecord	17
1.4.1	简介	17
1.4.2	请求语法	17
1.4.2.1	Http请求	17
1.4.2.2	Http头	17
1.4.2.3	Data参数	18
1.4.2.4	完整请求示例	19
1.4.3	返回语法	20
1.4.3.1	返回参数说明	20
1.4.3.2	返回示例	20
1.4.4	错误码	21
1.5	[PB] BatchInsertRecords	21
1.5.1	简介	21
1.5.2	请求语法	21
1.5.2.1	Http请求	21
1.5.2.2	Http头	22
1.5.2.3	Data	23
1.5.2.4	完整请求示例	25
1.5.3	返回语法	26
1.5.4	返回参数说明	26
1.5.5	返回示例	27
1.5.5.1	成功返回示例	27
1.5.5.2	失败返回示例	28
1.5.6	错误码	28

1.6	[PB] GetRecord2.0 .....	28
1.6.1	简介 .....	29
1.6.2	请求语法 .....	29
1.6.2.1	Http请求 .....	29
1.6.2.2	Http头 .....	29
1.6.2.3	Data .....	30
1.6.2.4	完整请求示例 .....	31
1.6.3	返回语法 .....	31
1.6.3.1	返回参数说明 .....	31
1.6.3.2	返回示例 .....	32
1.6.4	错误码 .....	32
1.7	[PB] PartkeyGetRecord2.0 .....	32
1.7.1	简介 .....	32
1.7.2	请求语法 .....	33
1.7.2.1	Http请求 .....	33
1.7.2.2	Http请求头参数 .....	33
1.7.2.3	Data .....	34
1.7.2.4	完整请求示例 .....	35
1.7.3	返回语法 .....	35
1.7.3.1	返回参数说明 .....	35
1.7.3.2	返回示例 .....	36
1.7.4	错误 .....	37
1.8	[PB] BatchGetRecords .....	37
1.8.1	简介 .....	37
1.8.2	请求语法 .....	37
1.8.2.1	Http请求 .....	37
1.8.2.2	Http头 .....	37
1.8.2.3	Data .....	38
1.8.2.4	完整请求示例 .....	39
1.8.3	返回语法 .....	40
1.8.3.1	返回参数 .....	40
1.8.3.2	返回示例 .....	41
1.8.4	错误 .....	42
1.9	[PB] DeleteRecord2.0 .....	42
1.9.1	简介 .....	42

1.9.2	请求语法 .....	43
1.9.2.1	Http请求 .....	43
1.9.2.2	Http头 .....	43
1.9.2.3	Data.....	44
1.9.2.4	完整请求示例 .....	45
1.9.3	返回语法 .....	45
1.9.3.1	返回参数说明 .....	45
1.9.3.2	返回示例 .....	46
1.9.4	错误 .....	46
1.10	[PB] DeleteRecordByPartKey .....	46
1.10.1	简介 .....	47
1.10.2	请求语法 .....	47
1.10.2.1	Http请求 .....	47
1.10.2.2	Http请求头 .....	47
1.10.2.3	Data参数 .....	48
1.10.2.4	完整请求示例 .....	49
1.10.3	返回语法 .....	49
1.10.3.1	返回参数 .....	49
1.10.3.2	返回示例 .....	49
1.10.4	错误 .....	50
1.11	[PB] BatchDeleteRecords .....	50
1.11.1	简介 .....	50
1.11.2	请求语法 .....	50
1.11.2.1	Http请求 .....	50
1.11.2.2	Http头 .....	50
1.11.2.3	Data参数 .....	51
1.11.2.4	完整请求示例 .....	52
1.11.3	返回语法 .....	53
1.11.3.1	返回参数说明 .....	53
1.11.3.2	返回示例 .....	54
1.11.4	错误 .....	55
1.12	[PB] UpdateRecord2.0 .....	55
1.12.1	简介 .....	55
1.12.2	请求语法 .....	56
1.12.2.1	Http请求 .....	56

1.12.2.2 Http头 .....	56
1.12.2.3 Data.....	57
1.12.2.4 完整请求示例 .....	58
1.12.3 返回语法 .....	58
1.12.3.1 返回参数说明 .....	58
1.12.3.2 返回示例 .....	59
1.13 [PB] BatchUpdateRecords.....	59
1.13.1 简介 .....	60
1.13.2 请求语法 .....	60
1.13.2.1 Http请求.....	60
1.13.2.2 Http头.....	60
1.13.2.3 Data.....	61
1.13.2.4 完整请求示例 .....	63
1.13.3 返回语法 .....	64
1.13.3.1 返回参数说明 .....	64
1.13.3.2 返回示例 .....	65
1.13.4 错误 .....	66
1.14 [Pb] ReplaceRecord2.0 .....	66
1.14.1 简介 .....	67
1.14.2 请求语法 .....	67
1.14.2.1 Http请求.....	67
1.14.2.2 Http头.....	67
1.14.2.3 Data.....	68
1.14.2.4 完整请求示例 .....	69
1.14.3 返回语法 .....	69
1.14.3.1 返回参数说明 .....	69
1.14.3.2 返回示例 .....	70
1.14.4 错误 .....	71
1.15 [PB] BatchReplaceRecords.....	71
1.15.1 简介 .....	71
1.15.2 请求语法 .....	71
1.15.2.1 Http请求.....	71
1.15.2.2 Http头.....	71
1.15.2.3 Data.....	72
1.15.2.4 完整请求示例 .....	74

1.15.3 返回语法 .....	75
1.15.3.1 返回参数说明 .....	75
1.15.3.2 返回示例 .....	76
1.15.4 错误 .....	76
1.16 [PB] FieldGetRecord2.0 .....	77
1.16.1 简介 .....	77
1.16.2 请求语法 .....	77
1.16.2.1 Http请求 .....	77
1.16.2.2 Http头 .....	77
1.16.2.3 Data .....	78
1.16.2.4 使用json格式表示记录相关信息。参数说明： .....	78
1.16.2.5 完整请求示例 .....	79
1.16.2.6 返回参数说明 .....	79
1.16.2.7 返回示例 .....	80
1.16.3 错误 .....	80
1.17 [PB] BatchFieldGetRecords .....	80
1.17.1 简介 .....	80
1.17.2 请求语法 .....	81
1.17.2.1 Http请求 .....	81
1.17.2.2 Http头 .....	81
1.17.2.3 Data .....	82
1.17.2.4 完整请求示例 .....	83
1.17.3 返回语法 .....	84
1.17.3.1 返回参数说明 .....	84
1.17.3.2 返回示例 .....	84
1.17.4 错误 .....	85
1.18 [PB] FieldIncRecord .....	85
1.18.1 简介 .....	85
1.18.2 请求语法 .....	86
1.18.2.1 Http请求 .....	86
1.18.2.2 Http头 .....	86
1.18.2.3 Data .....	87
1.18.2.4 完整请求参数 .....	88
1.18.3 返回参数 .....	88
1.18.3.1 返回参数说明 .....	88

1.18.3.2 返回示例 .....	88
1.19 [PB] FieldUpdateRecord .....	89
1.19.1 简介 .....	89
1.19.2 请求语法 .....	89
1.19.2.1 Http请求 .....	89
1.19.2.2 Http头 .....	90
1.19.2.3 Data .....	91
1.19.2.4 完整请求示例 .....	91
1.19.3 返回语法 .....	92
1.19.3.1 返回参数说明 .....	92
1.19.3.2 返回示例 .....	92
1.19.4 错误 .....	93
1.20 [PB] GetRecordCount .....	93
1.20.1 简介 .....	93
1.20.2 请求语法 .....	93
1.20.2.1 Http请求 .....	93
1.20.2.2 Http头 .....	93
1.20.2.3 Data .....	94
1.20.2.4 完整请求示例 .....	95
1.20.3 返回语法 .....	95
1.20.3.1 返回参数说明 .....	95
1.20.3.2 返回示例 .....	95
1.20.4 错误 .....	95
1.21 [PB] IndexQuery2.0 .....	95
1.21.1 简介 .....	96
1.21.2 请求语法 .....	96
1.21.2.1 Http请求 .....	96
1.21.2.2 Http头 .....	96
1.21.2.3 Data .....	97
1.21.2.4 完整请求示例 .....	98
1.21.3 返回语法 .....	98
1.21.3.1 返回参数说明 .....	98
1.21.3.2 返回示例 .....	99
1.21.4 错误 .....	99
1.21.5 支持的SQL语法 .....	99

1.21.5.1 条件查询 .....	99
1.21.5.2 分页查询.....	100
1.21.5.3 聚合查询 .....	100
1.21.5.4 部分字段查询 .....	100
1.21.6 不支持的 SQL语法 .....	101
1.21.6.1 不支持聚合查询与非聚合查询混用 .....	101
1.21.6.2 有限支持Order by查询.....	101
1.21.6.3 不支持group by查询.....	101
1.21.6.4 不支持having查询 .....	101
1.21.6.5 不支持多表联合查询 .....	101
1.21.6.6 不支持嵌套的SELECT查询.....	101
1.21.6.7 不支持别名 .....	102
1.21.6.8 其它不支持的查询 .....	102
1.22 [PB] Traverse2.0 .....	102
1.22.1 简介 .....	102
1.22.2 请求语法 .....	102
1.22.2.1 Http请求 .....	102
1.22.2.2 Http头 .....	103
1.22.2.3 Data.....	104
1.22.2.4 完整请求示例 .....	104
1.22.3 返回语法 .....	107
1.22.3.1 返回参数说明 .....	107
1.22.3.2 返回示例 .....	107
1.22.4 错误 .....	110
<b>2 RESTful API 2.0 for List表.....</b>	<b>111</b>
2.1 RESTful API 2.0说明 .....	111
2.2 RESTful API 使用前置 .....	111
2.2.1 TcaplusDB环境准备 .....	111
2.2.2 TcaplusDB表创建 .....	111
2.2.3 测试机器准备 .....	111
2.2.4 TcaplusDB List样例表.....	111
2.3 [PB] ListInsertRecord .....	112
2.3.1 简介 .....	112
2.3.2 请求语法 .....	113



2.3.2.1	Http请求 .....	113
2.3.2.2	Http头 .....	113
2.3.2.3	Data.....	114
2.3.2.4	完整请求示例 .....	115
2.3.3	返回语法 .....	116
2.3.3.1	返回参数说明 .....	116
2.3.3.2	返回示例 .....	116
2.3.4	错误码 .....	117
2.4	[PB] ListBatchInsertRecord .....	117
2.4.1	简介 .....	117
2.4.2	请求语法 .....	117
2.4.2.1	Http请求.....	117
2.4.2.2	Http头.....	118
2.4.2.3	Data.....	119
2.4.2.4	完整请求示例 .....	121
2.4.3	返回语法 .....	122
2.4.3.1	返回参数说明 .....	122
2.4.3.2	返回示例 .....	123
2.4.4	错误码 .....	125
2.5	[PB] ListGetRecord2.0 .....	125
2.5.1	简介 .....	125
2.5.2	请求语法 .....	125
2.5.2.1	Http请求.....	125
2.5.2.2	Http头.....	125
2.5.2.3	Data.....	127
2.5.2.4	完整请求示例 .....	127
2.5.3	返回语法 .....	128
2.5.3.1	返回参数说明 .....	128
2.5.3.2	返回示例 .....	128
2.5.4	错误码 .....	129
2.6	[PB] ListBatchGetRecord .....	129
2.6.1	简介 .....	129
2.6.2	请求语法 .....	129
2.6.2.1	Http请求.....	129
2.6.2.2	Http头.....	129

2.6.2.3	Data.....	131
2.6.2.4	完整请求示例 .....	131
2.6.3	返回语法 .....	132
2.6.3.1	返回参数说明 .....	132
2.6.3.2	返回示例 .....	133
2.6.4	错误码 .....	134
2.6.5	1. 简介.....	134
2.6.6	2.请求语法 .....	135
2.6.6.1	2.1 Http请求 .....	135
2.6.6.2	2.2 Http头 .....	135
2.6.6.3	2.3 Data.....	136
2.6.7	3. 返回语法.....	137
2.6.8	5.返回参数 .....	137
2.6.9	6错误 .....	138
2.6.10	7.示例 .....	138
2.6.11	8. 还可以看.....	138
2.7	[PB] ListGetAllRecords2.0 .....	138
2.7.1	简介 .....	139
2.7.2	请求语法 .....	139
2.7.2.1	Http请求.....	139
2.7.2.2	Http头.....	139
2.7.2.3	Data.....	140
2.7.2.4	完整请求示例 .....	141
2.7.3	返回语法 .....	141
2.7.3.1	返回参数说明 .....	141
2.7.3.2	返回示例 .....	142
2.7.4	错误码 .....	143
2.8	[PB] ListDeleteRecord2.0 .....	143
2.8.1	简介 .....	143
2.8.2	请求语法 .....	143
2.8.2.1	Http请求.....	143
2.8.2.2	Http头.....	143
2.8.2.3	Data.....	145
2.8.2.4	完整请求示例 .....	145
2.8.3	返回语法 .....	145

2.8.3.1	返回参数说明 .....	145
2.8.3.2	返回示例 .....	146
2.8.4	错误码 .....	146
2.9	[PB] ListBatchDeleteRecords2.0 .....	147
2.9.1	简介 .....	147
2.9.2	请求语法 .....	147
2.9.2.1	Http请求 .....	147
2.9.2.2	Http头 .....	147
2.9.2.3	Data .....	148
2.9.2.4	完整请求示例 .....	149
2.9.3	返回语法 .....	149
2.9.3.1	返回参数说明 .....	149
2.9.3.2	返回示例 .....	150
2.9.4	错误码 .....	151
2.10	[PB] ListDeleteAllRecords2.0 .....	151
2.10.1	简介 .....	151
2.10.2	请求语法 .....	151
2.10.2.1	Http请求 .....	151
2.10.2.2	Http头 .....	152
2.10.2.3	Data .....	153
2.10.2.4	完整请求示例 .....	154
2.10.3	返回语法 .....	154
2.10.3.1	返回参数说明 .....	154
2.10.3.2	返回示例 .....	154
2.10.4	错误码 .....	155
2.10.5	1. 简介 .....	155
2.10.6	2. 请求语法 .....	155
2.10.6.1	2.1 Http请求 .....	155
2.10.6.2	2.2 Http头 .....	155
2.10.6.3	2.3 Data .....	156
2.10.7	3. 请求参数 .....	156
2.10.7.1	3.1 Http头参数 .....	156
2.10.8	4. 返回语法 .....	157
2.10.9	5. 返回参数 .....	157
2.10.10	6 错误 .....	158

2.10.11 7.示例 .....	158
2.10.12 8. 还可以看.....	158
2.11 [PB] ListUpdateRecord .....	158
2.11.1 简介 .....	158
2.11.2 请求语法 .....	159
2.11.2.1 Http请求.....	159
2.11.2.2 Http头.....	159
2.11.2.3 Data.....	160
2.11.2.4 完整请求示例 .....	161
2.11.3 返回语法 .....	162
2.11.3.1 返回参数说明 .....	162
2.11.3.2 返回示例 .....	162
2.11.4 错误码 .....	163
2.12 [PB] ListBatchUpdateRecord.....	163
2.12.1 简介 .....	163
2.12.2 请求语法 .....	163
2.12.2.1 Http请求.....	163
2.12.2.2 Http头.....	164
2.12.2.3 Data.....	165
2.12.2.4 完整请求示例 .....	167
2.12.3 返回语法 .....	167
2.12.3.1 返回参数说明 .....	167
2.12.3.2 返回示例 .....	168
2.12.4 错误码 .....	169
2.13 [PB] ListTraverse2.0 .....	170
2.13.1 简介 .....	170
2.13.2 请求语法 .....	170
2.13.2.1 Http请求.....	170
2.13.2.2 Http头.....	171
2.13.2.3 Data.....	172
2.13.2.4 完整请求示例 .....	172
2.13.3 返回语法 .....	175
2.13.3.1 返回参数说明 .....	175
2.13.3.2 返回示例 .....	175
2.13.4 错误 .....	179

3	[PB]常见错误码 .....	180
---	-----------------	-----

本章节主要包括以下几个部分：

# 1 RESTful API 2.0 for Generic表

## 1.1 RESTful API 2.0说明

目前RESTful API 版本已升级到2.0，针对TcaplusDB Generic表，已开放19个接口，涵盖单笔，批量，索引查询等方面。

## 1.2 RESTful API 使用前置

### 1.2.1 TcaplusDB环境准备

针对腾讯云环境，直接用腾讯云账号进TcaplusDB控制台进行相关操作即可。

针对TcaplusDB本地Docker环境，部署步骤请参考：[Docker环境部署](#)<sup>1</sup>。

### 1.2.2 TcaplusDB表创建

针对腾讯云环境，创建TcaplusDB表请参考：[腾讯云环境创建表](#)<sup>2</sup>。

针对TcaplusDB本地Docker环境，创建TcaplusDB表请参考：[Docker环境表创建](#)<sup>3</sup>。

### 1.2.3 测试机器准备

针对腾讯云环境，请申请一台CVM实例，VPC环境与TcaplusDB的VPC环境保持一致。

针对TcaplusDB本地Docker环境，可以准备一台开发测试机，如果涉及跨机访问Docker环境，需要注意连通策略，同时Docker部署以本地Host模式为主。

### 1.2.4 TcaplusDB样例表

表名: **game\_players**, proto文件名: **game\_players.proto**

---

<sup>1</sup> <https://github.com/tencentyun/tcaplusdb-documents/blob/main/docker/TcaplusDB%E5%85%A5%E9%97%A8-Docker%E9%83%A8%E7%BD%B2%E7%AF%87.md>

<sup>2</sup> <https://cloud.tencent.com/document/product/596/38808>

<sup>3</sup> <https://github.com/tencentyun/tcaplusdb-documents/tree/main/tcapluscli>

```

syntax = "proto3";                                // Specify the version of the protocol buffers language

import "tcapluservice.optionv1.proto"; // Use the public definitions of TcaplusDB by importing them.

message game_players { // Define a TcaplusDB table with message

    // Specify the primary keys with the option tcapluservice.tcaplus_primary_key
    // The primary key of a TcaplusDB table has a limit of 4 fields
    option(tcapluservice.tcaplus_primary_key) = "player_id, player_name, player_email";

    // Specify the primary key indexes with the option tcapluservice.tcaplus_index
    option(tcapluservice.tcaplus_index) = "index_1(player_id, player_name)";
    option(tcapluservice.tcaplus_index) = "index_2(player_id, player_email)";

    // Value Types supported by TcaplusDB
    // int32, int64, uint32, uint64, sint32, sint64, bool, fixed64, sfixed64, double, fixed32, sfixed32,
    float, string, bytes
    // Nested Types Message

    // primary key fields
    uint64 player_id = 1;
    string player_name = 2;
    string player_email = 3;

    // Ordinary fields
    int32 game_server_id = 4;
    repeated string login_timestamp = 5;
    repeated string logout_timestamp = 6;
    bool is_online = 7;

    payment pay = 8;
}

message payment {

    int64 pay_id = 1;
    uint64 amount = 2;
    int64 method = 3;
}

```

## 1.3 RESTful API 2.0 接口列表

Generic表 RESTful API 2.0主要包括19个接口，有单笔和批量。接口列表如下：



## 1.4 [PB] InsertRecord

支持版本：3.50.0

### 1.4.1 简介

在表中插入单笔数据，如果数据已存在则返回错误。

```
POST http://{Tcaplus_REST_URL}
```

### 1.4.2 请求语法

#### 1.4.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.4.2.2 Http头

```
x-tcaplus-target:Tcaplus.InsertRecord
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.InsertRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)

名称	是否必填	限制条件	说明
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.InsertRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.4.2.3 Data参数

使用json格式表示记录相关信息，示例如下：

```
{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4",
    "game_server_id":4,
    "login_timestamp":[],
    "logout_timestamp":[],
    "is_online":false,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

#### 1.4.2.4 完整请求示例

注意：示例中相关参数请替换成业务方TcaplusDB实际参数。

```
curl -i -XPOST -H 'x-tpplus-target: Tcaplus.InsertRecord' -H 'x-tpplus-app-id: 70' -H 'x-tpplus-zone-id: 1' -H 'x-tpplus-protocol-version: 2.0' -H 'x-tpplus-table-name: game_players' -H 'x-tpplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tpplus-result-flag: 2' -H 'x-tpplus-version: Tcaplus3.50.0' -H 'x-tpplus-data-version-check: 3' -H 'x-tpplus-id1-type: protobuf' http://172.17.32.12 -d '{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4",
    "game_server_id":4,
    "login_timestamp":[],
    "logout_timestamp":[],
    "is_online":false,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}'
```

### 1.4.3

## 返回语法

#### 1.4.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

#### 1.4.3.2 返回示例

##### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "Record": {
    "player_id": 4,
    "player_name": "4",
    "player_email": "4",
    "game_server_id": 4,
    "login_timestamp": [],
    "logout_timestamp": [],
    "is_online": false,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

错误返回

```
{
  "ErrorCode": -1293,
  "ErrorMsg": "Record exists"
}
```

#### 1.4.4 错误码

参考 [\[PB\]常见错误码](#)<sup>4</sup>

### 1.5 [PB] BatchInsertRecords

支持版本：3.50.0

#### 1.5.1 简介

在PB表中批量添加数据，可指定插入方式。批量插入接口非原子接口，允许部分成功，部分失败，用户需根据返回值来判断成功的记录数或失败的记录数。

```
POST http://{Tcaplus_REST_URL}
```

#### 1.5.2 请求语法

##### 1.5.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

<sup>4</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 1.5.2.2 Http头

```
x-tcaplus-target:Tcaplus.BatchInsertRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.BatchInsertRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.BatchInsertRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.5.2.3 Data

使用json格式表示记录相关信息。参数说明：

- MultiRecords: 记录集
- Record: 必选, 子记录数据

```

{
  "MultiRecords": [{
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "55",
      "game_server_id": 55,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 55,
        "amount": 55,
        "method": 55
      }
    }
  }, {
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }
}]
}

```



## 1.5.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.BatchInsertRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 3' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "MultiRecords": [{
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"5",
      "game_server_id":5,
      "login_timestamp":[],
      "logout_timestamp":[],
      "is_online":false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"55",
      "game_server_id":55,
      "login_timestamp":[],
      "logout_timestamp":[],
      "is_online":false,
      "pay": {
        "pay_id": 55,
        "amount": 55,
        "method": 55
      }
    }
  }, {
    "Record": {
      "player_id":6,
      "player_name":"6",
      "player_email":"6",
      "game_server_id":6,
      "login_timestamp":[],
      "logout_timestamp":[],
      "is_online":false,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }
}]
```

```
}'  
  }  
}
```

### 1.5.3 返回语法

### 1.5.4 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

## 1.5.5 返回示例

### 1.5.5.1 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 1,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "55",
      "game_server_id": 55,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 55,
        "amount": 55,
        "method": 55
      }
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 6,
        "amount": 6,
```

```

        "method": 6
      }
    }
  },
  "SucceedNum": 3
}

```

### 1.5.5.2 失败返回示例

```

{
  "ErrorCode": -1293,
  "ErrorMsg": "tcapsvr_fail_record_exist",
  "SucceedRecords": [],
  "SucceedNum": 0,
  "FailedRecords": [{
    "Record": {
      "player_email": "5",
      "player_id": 5,
      "player_name": "5"
    }
  }, {
    "Record": {
      "player_email": "55",
      "player_id": 5,
      "player_name": "5"
    }
  }, {
    "Record": {
      "player_email": "6",
      "player_id": 6,
      "player_name": "6"
    }
  }
],
  "FailedNum": 3
}

```

## 1.5.6 错误码

参考 [\[PB\]常见错误码](#)<sup>5</sup>

## 1.6 [PB] GetRecord2.0

支持版本：3.50.0

<sup>5</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 1.6.1 简介

单笔记录查询接口，如果记录不存在，则返回261错误码。支持返回指定字段，通过Select参数指定要返回的字段名。

```
POST http://{Tcaplus_REST_URL}
```

## 1.6.2 请求语法

### 1.6.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 1.6.2.2 Http头

```
x-tcaplus-target:Tcaplus.GetRecord
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.GetRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0

名称	是否必填	限制条件	说明
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.GetRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb498233aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.6.2.3 Data

使用json格式表示记录相关信息。参数说明：

- Select：可选，参数指定返回的字段，不传时默认返回记录所有字段。
- Record：必须，参数指定**主键字段**用于惟一记录获取。

注意：如果Data带有Select参数，Select不能是主键字段名 (因为主键字段已经默认会返回)

```
{
  "Record": {
    "player_id":5,
    "player_name":"5",
    "player_email":"5"
  },
  "Select":["game_server_id", "is_online", "pay"]
}
```

#### 1.6.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.GetRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1'
-H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5:
0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H
'x-tcaplus-data-version-check: 3' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "player_id":5,
    "player_name":"5",
    "player_email":"5"
  },
  "Select":["game_server_id", "is_online", "pay"]
}'
```

### 1.6.3 返回语法

#### 1.6.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnVersion	记录版本号
Record	json格式, 成功的记录返回数据

### 1.6.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 1,
  "Record": {
    "game_server_id": 5,
    "is_online": false,
    "pay": {
      "pay_id": 5,
      "amount": 5,
      "method": 5
    },
    "player_email": "5",
    "player_id": 5,
    "player_name": "5"
  }
}
```

#### 失败示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

## 1.6.4 错误码

参考 [\[PB\]常见错误码](#)<sup>6</sup>

## 1.7 [PB] PartkeyGetRecord2.0

支持版本：3.50.0

### 1.7.1 简介

根据主键索引获取相关记录，主键索引在表定义proto文件中定义。主键索引是基于主键字段设计的一种索引。接口同时支持过滤字段返回，通过Select参数指定要返回的字段名；并可控制返回的记录条数，通过参数Limit指定。

<sup>6</sup> <https://iwiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>



```
POST http://{Tcaplus_REST_URL}
```

## 1.7.2 请求语法

### 1.7.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 1.7.2.2 Http请求头参数

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.GetRecordByPartKey
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.GetRecordByPartKey
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.7.2.3 Data

使用json格式表示记录相关信息。参数说明:

- Select: 可选, 过滤返回的字段, 指定字段名, **不能为主键字段名**。
- Limit: 可选, 限制每次请求返回的记录条数
- Record: 必须, 指定主键索引字段

示例：

```
{
  "Limit": 1,
  "Record": {
    "player_id": 5,
    "player_name": "5"
  },
  "Select": ["game_server_id", "is_online", "pay"]
}
```

### 1.7.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.GetRecordByPartKey' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 3' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "player_id":5,
    "player_name":"5"
  },
  "Select":["game_server_id", "is_online", "pay"]
}'
```

## 1.7.3 返回语法

### 1.7.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	记录版本号
MultiRecords	json格式, 成功的记录返回数据数组, 数组元素为Record记录数据
TotalNum	成功的记录数

### 1.7.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "MultiRecords": [{
    "RecordVersion": 1,
    "Record": {
      "game_server_id": 55,
      "is_online": false,
      "pay": {
        "pay_id": 55,
        "amount": 55,
        "method": 55
      },
      "player_email": "55",
      "player_id": 5,
      "player_name": "5"
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "game_server_id": 5,
      "is_online": false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      },
      "player_email": "5",
      "player_id": 5,
      "player_name": "5"
    }
  }],
  "TotalNum": 2
}
```

#### 失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

### 1.7.4 错误

参考 [\[PB\]常见错误码<sup>7</sup>](#)

## 1.8 [PB] BatchGetRecords

支持版本：3.50.0

### 1.8.1 简介

根据主键批量获取记录，满足批量查询场景，支持返回部分字段

```
POST http://{Tcaplus_REST_URL}
```

### 1.8.2 请求语法

#### 1.8.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.8.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.BatchGetRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0

<sup>7</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

名称	是否必填	限制条件	说明
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.BatchGetRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

### 1.8.2.3 Data

使用json格式表示记录相关信息。参数说明：

- Select: 可选, 过滤要返回的字段, 不能包含主键字段 (默认返回)
- MultiRecords: 必须, 记录集
- Record: 必须, 记录主键字段

示例：

```
{
  "Select": ["game_server_id", "is_online", "pay"],
  "MultiRecords": [{
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5"
    }
  }, {
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6"
    }
  }]
}
```

#### 1.8.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.BatchGetRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Select":["game_server_id", "is_online", "pay"],
  "MultiRecords": [{
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"5"
    }
  }, {
    "Record": {
      "player_id":6,
      "player_name":"6",
      "player_email":"6"
    }
  }]
}'
```

### 1.8.3 返回语法

#### 1.8.3.1 返回参数

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数



### 1.8.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 1,
    "Record": {
      "game_server_id": 5,
      "is_online": false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      },
      "player_email": "5",
      "player_id": 5,
      "player_name": "5"
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "game_server_id": 6,
      "is_online": false,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      },
      "player_email": "6",
      "player_id": 6,
      "player_name": "6"
    }
  }],
  "SucceedNum": 2
}
```

## 失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "txhdb_record_not_exist",
  "SucceedRecords": [],
  "SucceedNum": 0,
  "FailedRecords": [{
    "Record": {
      "player_email": "5",
      "player_id": 5,
      "player_name": "5"
    }
  }, {
    "Record": {
      "player_email": "6",
      "player_id": 6,
      "player_name": "6"
    }
  }],
  "FailedNum": 2
}
```

## 1.8.4 错误

参考 [\[PB\]常见错误码](#)<sup>8</sup>

## 1.9 [PB] DeleteRecord2.0

支持版本：3.50.0

## 1.9.1 简介

通过指定的主键key删除记录，如果记录不存在，则返回相关错误信息。

```
POST http://{Tcaplus_REST_URL}
```

<sup>8</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 1.9.2 请求语法

### 1.9.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 1.9.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.DeleteRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.DeleteRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:3
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.9.2.3 Data

使用json格式表示记录相关信息。参数说明：

- Record: 必须, 记录数据, 包含全部主键字段

示例：

```
{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4"
  }
}
```

### 1.9.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.DeleteRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4"
  }
}'
```

## 1.9.3 返回语法

### 1.9.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
Record	成功的记录数

### 1.9.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 1,
  "Record": {
    "player_id": 4,
    "player_name": "4",
    "player_email": "4",
    "game_server_id": 4,
    "login_timestamp": [],
    "logout_timestamp": [],
    "is_online": false,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

#### 失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

### 1.9.4 错误

参考 [PB]常见错误码<sup>9</sup>

## 1.10 [PB] DeleteRecordByPartKey

支持版本：3.50.0

---

<sup>9</sup> <https://iwiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 1.10.1 简介

根据表主键索引删除部分记录。

```
POST http://{Tcaplus_REST_URL}
```

### 1.10.2 请求语法

#### 1.10.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.10.2.2 Http请求头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.DeleteRecordByPartKey
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.DeleteRecordByPartKey
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.10.2.3 Data参数

以JSON形式传参。参数说明：

- Record: 必选，根据指定的主键索引字段来删除记录

```
{
  "Record": {
    "player_id":5,
    "player_name":"5"
  }
}
```



#### 1.10.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.DeleteRecordByPartKey' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 3' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
    "Record": {
        "player_id":5,
        "player_name":"5"
    }
}'
```

#### 1.10.3 返回语法

##### 1.10.3.1 返回参数

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
TotalNum	删除成功的记录条数

##### 1.10.3.2 返回示例

###### 成功返回示例

```
{
    "ErrorCode": 0,
    "ErrorMsg": "Succeed",
    "TotalNum": 1
}
```

###### 失败返回示例

```
{
    "ErrorCode": 261,
    "ErrorMsg": "Record does not exist"
}
```

### 1.10.4 错误

参考 [\[PB\]常见错误码](#)<sup>10</sup>

## 1.11 [PB] BatchDeleteRecords

支持版本：3.50.0

### 1.11.1 简介

根据表主键批量删除数据

```
POST http://{Tcaplus_REST_URL}
```

### 1.11.2 请求语法

#### 1.11.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.11.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.BatchDeleteRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0

<sup>10</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

名称	是否必填	限制条件	说明
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.InsertRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.11.2.3 Data参数

使用json格式表示记录相关信息。参数：

- MultiRecords: 必须, 删除记录的集合表示
- Record: 必须, 删除记录的主键字段

示例:

```
{
  "MultiRecords": [{
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"5"
    }
  }, {
    "Record": {
      "player_id":6,
      "player_name":"6",
      "player_email":"6"
    }
  }]
}
```

#### 1.11.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.BatchDeleteRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 3' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "MultiRecords": [{
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"5"
    }
  }, {
    "Record": {
      "player_id":6,
      "player_name":"6",
      "player_email":"6"
    }
  }]
}'
```

### 1.11.3 返回语法

#### 1.11.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
SucceedRecords	json格式的返回数据, 详见文中Data部分
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 1.11.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 1,
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }],
  "SucceedNum": 2
}
```

## 失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "txhdb_record_not_exist",
  "SucceedRecords": [],
  "SucceedNum": 0,
  "FailedRecords": [{
    "Record": {
      "player_email": "5",
      "player_id": 5,
      "player_name": "5"
    }
  }, {
    "Record": {
      "player_email": "6",
      "player_id": 6,
      "player_name": "6"
    }
  }],
  "FailedNum": 2
}
```

## 1.11.4 错误

参考 [\[PB\]常见错误码](#)<sup>11</sup>

## 1.12 [PB] UpdateRecord2.0

支持版本：3.50.0

## 1.12.1 简介

添加一条新记录或者用新记录替换旧纪录。如果数据库中存在记录和新记录的key信息相同，那么就用新记录替换旧记录。如果不存在，那么返回错误信息。

```
POST http://{Tcaplus_REST_URL}
```

<sup>11</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 1.12.2 请求语法

### 1.12.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 1.12.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.UpdateRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>



名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.UpdateRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.12.2.3 Data

使用json格式表示记录相关信息。参数说明：

- Record: 必选，要更新的记录数据

```
{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4",
    "game_server_id":4,
    "login_timestamp":["123456"],
    "logout_timestamp":["123456"],
    "is_online":true,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

#### 1.12.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.UpdateRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 3' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4",
    "game_server_id":4,
    "login_timestamp":["123456"],
    "logout_timestamp":["123456"],
    "is_online":true,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}'
```

#### 1.12.3 返回语法

##### 1.12.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	返回更新后记录的版本号
Record	json格式, 成功的记录数据

### 1.12.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 2,
  "Record": {
    "player_id": 4,
    "player_name": "4",
    "player_email": "4",
    "game_server_id": 4,
    "login_timestamp": ["123456"],
    "logout_timestamp": ["123456"],
    "is_online": true,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

#### 失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

## 错误

参考 [\[PB\] 常见错误码](#)<sup>12</sup>

## 1.13 [PB] BatchUpdateRecords

支持版本：3.50.0

---

<sup>12</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pagelId=237730186>

### 1.13.1 简介

批量更新记录（记录不存在报错，记录存在则修改）。

```
POST http://{Tcaplus_REST_URL}
```

### 1.13.2 请求语法

#### 1.13.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.13.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.BatchUpdateRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf

名称	是否必填	限制条件	说明
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.BatchUpdateRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.13.2.3 Data

使用json格式表示记录相关信息。参数说明：

- MultiRecords: 更新的记录集合
- Record: 更新的记录数据

```
{
  "MultiRecords": [{
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": ["123456"],
      "logout_timestamp": ["123456"],
      "is_online": true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": ["123456"],
      "logout_timestamp": ["123456"],
      "is_online": true,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  ]
}
```

### 1.13.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.BatchUpdateRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 3' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "MultiRecords": [{
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"5",
      "game_server_id":5,
      "login_timestamp":["123456"],
      "logout_timestamp":["123456"],
      "is_online":true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "Record": {
      "player_id":6,
      "player_name":"6",
      "player_email":"6",
      "game_server_id":6,
      "login_timestamp":["123456"],
      "logout_timestamp":["123456"],
      "is_online":true,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }]
}'
```

### 1.13.3 返回语法

#### 1.13.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
SucceedRecords	json格式的返回数据, 详见文中Data部分
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数



### 1.13.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 5,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": ["123456"],
      "logout_timestamp": ["123456"],
      "is_online": true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "RecordVersion": 5,
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": ["123456"],
      "logout_timestamp": ["123456"],
      "is_online": true,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }],
  "SucceedNum": 2
}
```

## 失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "txhdb_record_not_exist",
  "SucceedRecords": [{
    "RecordVersion": 6,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": ["123456"],
      "logout_timestamp": ["123456"],
      "is_online": true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }],
  "SucceedNum": 1,
  "FailedRecords": [{
    "Record": {
      "player_email": "6",
      "player_id": 66,
      "player_name": "6"
    }
  }],
  "FailedNum": 1
}
```

## 1.13.4 错误

参考 [\[PB\]常见错误码](#)<sup>13</sup>

## 1.14 [Pb] ReplaceRecord2.0

支持版本：3.50.0

<sup>13</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 1.14.1 简介

添加一条新记录或者用新记录替换旧纪录。如果记录已存在，则更新；如果记录不存在，则插入新记录。

```
POST http://{Tcaplus_REST_URL}
```

### 1.14.2 请求语法

#### 1.14.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.14.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ReplaceRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ReplaceRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb498233aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.14.2.3 Data

使用json格式表示记录相关信息。

```
{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4",
    "game_server_id":4,
    "login_timestamp":["123456","234"],
    "logout_timestamp":["123456","234"],
    "is_online":true,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

#### 1.14.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ReplaceRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 3' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "player_id":4,
    "player_name":"4",
    "player_email":"4",
    "game_server_id":4,
    "login_timestamp":["123456","234"],
    "logout_timestamp":["123456","234"],
    "is_online":true,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}'
```

#### 1.14.3 返回语法

##### 1.14.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

### 1.14.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 0,
  "Record": {
    "player_id": 4,
    "player_name": "4",
    "player_email": "4",
    "game_server_id": 4,
    "login_timestamp": ["123456"],
    "logout_timestamp": ["123456"],
    "is_online": true,
    "pay": {
      "pay_id": 4,
      "amount": 4,
      "method": 4
    }
  }
}
```

#### 失败返回示例

如果记录不存在, 则插入记录。

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 1,
  "Record": {
    "player_id": 4,
    "player_name": "44",
    "player_email": "44",
    "game_server_id": 44,
    "login_timestamp": ["123456", "234"],
    "logout_timestamp": ["123456", "234"],
    "is_online": true,
    "pay": {
      "pay_id": 44,
      "amount": 44,
      "method": 44
    }
  }
}
```

### 1.14.4 错误

参考 [\[PB\]常见错误码](#)<sup>14</sup>

## 1.15 [PB] BatchReplaceRecords

支持版本：3.50.0

### 1.15.1 简介

批量更新记录（记录不存在新增，记录存在则修改）

```
POST http://{Tcaplus_REST_URL}
```

### 1.15.2 请求语法

#### 1.15.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80  
http://172.17.0.22  
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001  
http://9.135.8.93:31001
```

#### 1.15.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.BatchReplaceRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)

<sup>14</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

名称	是否必填	限制条件	说明
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.BatchReplaceRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb498233aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.15.2.3 Data

使用json格式表示记录相关信息。



```
{
  "MultiRecords": [{
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  ]
}
```

## 1.15.2.4 完整请求示例

```

curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.BatchReplaceRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 2' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "MultiRecords": [{
    "Record": {
      "player_id":5,
      "player_name":"5",
      "player_email":"5",
      "game_server_id":5,
      "login_timestamp":["123456","234"],
      "logout_timestamp":["123456","234"],
      "is_online":true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "Record": {
      "player_id":6,
      "player_name":"6",
      "player_email":"6",
      "game_server_id":6,
      "login_timestamp":["123456","234"],
      "logout_timestamp":["123456","234"],
      "is_online":true,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  ]
}
}'

```

### 1.15.3 返回语法

#### 1.15.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
SucceedRecords	json格式的返回数据, 详见文中Data部分
ReturnValues	用户设置的保留数据, 随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 1.15.3.2 返回示例

成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 3,
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }, {
    "RecordVersion": 3,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 5,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }],
  "SucceedNum": 2
}
```

### 1.15.4 错误

参考 [\[PB\]常见错误码](#)<sup>15</sup>

<sup>15</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 1.16 [PB] FieldGetRecord2.0

支持版本：3.50.0

### 1.16.1 简介

查询对应记录，获取指定字段返回。与GetRecord返回指定字段相比（Select参数），区别为：FieldGetRecord在服务端过滤，网络流量相对较小，推荐用户在有类似部分字段返回场景时用此接口。

```
POST http://{Tcaplus_REST_URL}
```

### 1.16.2 请求语法

#### 1.16.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.16.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.FieldGetRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf

名称	是否必填	限制条件	说明
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.FieldGetRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.16.2.3 Data

#### 1.16.2.4 使用json格式表示记录相关信息。参数说明：

- FieldPath: 必须, 指定返回的字段, 类似GetRecord的Select参数
- Record: 必须, 指定记录的主键字段

```
{
  "FieldPath": ["game_server_id", "pay"],
  "Record": {
    "player_id": 5,
    "player_name": "5",
    "player_email": "5"
  }
}
```

#### 1.16.2.5 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.FieldGetRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "FieldPath": ["game_server_id", "pay"],
  "Record": {
    "player_id": 5,
    "player_name": "5",
    "player_email": "5"
  },
  "Select": ["game_server_id", "is_online", "pay"]
}'
```

## 返回语法

#### 1.16.2.6

##### 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	记录版本号
Record	json格式, 返回的指定字段的记录数据

### 1.16.2.7 返回示例

成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 5,
  "Record": {
    "game_server_id": 5,
    "pay": {
      "pay_id": 5,
      "amount": 5,
      "method": 5
    }
  }
}
```

失败返回示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

## 1.16.3 错误

参考 [\[PB\]常见错误码](#)<sup>16</sup>

## 1.17 [PB] BatchFieldGetRecords

支持版本：3.50.0

### 1.17.1 简介

批量返回指定字段的记录数。

```
POST http://{Tcaplus_REST_URL}
```

---

<sup>16</sup> <https://iwiki.oa.tencent.com/pages/viewpage.action?pagelId=237730186>



## 1.17.2 请求语法

### 1.17.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 1.17.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.BatchFieldGetRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例:

```
x-tcaplus-target:Tcaplus.BatchFieldGetRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.17.2.3 Data

使用json格式表示记录相关信息。参数说明：

- FieldPath: 必须，指定返回的字段
- MultiRecords: 批量记录集合
- Record: 记录的主键字段

示例：

```
{
  "FieldPath": ["game_server_id", "pay"],
  "MultiRecords": [{
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5"
    }
  }, {
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6"
    }
  }]
}
```

#### 1.17.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.BatchFieldGetRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "FieldPath": ["game_server_id", "pay"],
  "MultiRecords": [{
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5"
    }
  }, {
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6"
    }
  }]
}'
```

### 1.17.3 返回语法

#### 1.17.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
MultiRecords	json数组, 返回成功的记录数据
TotalNum	返回成功的记录条数

#### 1.17.3.2 返回示例

成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "MultiRecords": [{
    "RecordVersion": 0,
    "Record": {
      "game_server_id": 5,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "RecordVersion": 0,
    "Record": {
      "game_server_id": 6,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }],
  "TotalNum": 2
}
```

## 失败返回示例

```
{
  "ErrorCode": -2560,
  "ErrorMsg": "some records fetch fail, check your keys.x is right or not",
  "MultiRecords": [{
    "RecordVersion": 0,
    "Record": {
      "game_server_id": 6,
      "pay": {
        "pay_id": 6,
        "amount": 6,
        "method": 6
      }
    }
  }],
  "TotalNum": 2
}
```

## 1.17.4 错误

参考 [\[PB\]常见错误码](#)<sup>17</sup>

## 1.18 [PB] FieldIncRecord

支持版本：3.50.0

## 1.18.1 简介

自增记录中选中的对应字段，只作用于数值类型字段。常用于玩家经验值增加，等级增加等场景，避免先查再更新，减少网络交互。

```
POST http://{Tcaplus_REST_URL}
```

<sup>17</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 1.18.2 请求语法

### 1.18.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 1.18.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.FieldIncRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.FieldIncRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.18.2.3 Data

使用json格式表示记录相关信息。参数说明：

- FieldPath: 必须，指定要自增的数值类型字段名
- Record: 必须，记录的主键字段，及自增字段要自增的值

示例：

```
{
  "FieldPath": ["game_server_id"],
  "Record": {
    "player_id":5,
    "player_name":"5",
    "player_email":"5",
    "game_server_id": 1
  }
}
```

### 1.18.2.4 完整请求参数

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.FieldIncRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "FieldPath": ["game_server_id"],
  "Record": {
    "player_id":5,
    "player_name":"5",
    "player_email":"5",
    "game_server_id": 1
  }
}'
```

## 1.18.3 返回参数

### 1.18.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	返回记录版本
Record	json格式, 返回成功的记录

### 1.18.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 6,
  "Record": {
    "game_server_id": 6
  }
}
```



## 失败返回示例

```
#示例1
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
#示例2
{
  "ErrorCode": -1811,
  "ErrorMsg": "Missing FieldPath when call FieldOperate Target.\u0000"
}
```

## 错误

参考 [\[PB\]常见错误码](#)<sup>18</sup>

## 1.19 [PB] FieldUpdateRecord

支持版本：3.50.0

## 1.19.1 简介

更新与指定字段相关联的信息

```
POST http://{Tcaplus_REST_URL}
```

## 1.19.2 请求语法

## 1.19.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

<sup>18</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 1.19.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.FieldUpdateRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>• 0: 操作成功后不返回数据</li> <li>• 1: 操作成功后返回和请求一致的数据</li> <li>• 2: 操作成功后返回本次update操作后的数据</li> <li>• 3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>• 1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>• 2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>• 3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.FieldUpdateRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.19.2.3 Data

使用json格式表示记录相关信息。参数：

- FieldPath: 必须，指定要更新的字段
- Record: 必须：指定要更新的记录的主键字段，及要更新的字段的值

示例：

```
{
  "FieldPath": ["is_online"],
  "Record": {
    "player_id":5,
    "player_name":"5",
    "player_email":"5",
    "is_online": false
  }
}
```

### 1.19.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.FieldUpdateRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "FieldPath": ["is_online"],
  "Record": {
    "player_id":5,
    "player_name":"5",
    "player_email":"5",
    "is_online": false
  }
}'
```

### 1.19.3 返回语法

#### 1.19.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
Record	json格式, 返回操作成功的更新字段值
RecordVersion	返回记录版本号

#### 1.19.3.2 返回示例

成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 7,
  "Record": {
    "is_online": false
  }
}
```

失败返回示例

```
#示例1
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
#示例2
{
  "ErrorCode": -1811,
  "ErrorMsg": "Missing FieldPath when call FieldOperate Target.\u0000"
}
```

## 1.19.4 错误

参考 [\[PB\]常见错误码](#)<sup>19</sup>

## 1.20 [PB] GetRecordCount

支持版本：3.50.0

### 1.20.1 简介

获取表中数据总的记录条数。

```
POST http://{Tcaplus_REST_URL}
```

### 1.20.2 请求语法

#### 1.20.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.20.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.GetRecordCount
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名

<sup>19</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

名称	是否必填	限制条件	说明
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>• 0: 操作成功后不返回数据</li> <li>• 1: 操作成功后返回和请求一致的数据</li> <li>• 2: 操作成功后返回本次update操作后的数据</li> <li>• 3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>• 1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>• 2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>• 3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.GetRecordCount
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.20.2.3 Data

无Data参数。

### 1.20.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.GetRecordCount' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d ''
```

## 1.20.3 返回语法

### 1.20.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
TotalNum	表的记录条数

### 1.20.3.2 返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "TotalNum": 7
}
```

## 1.20.4 错误

参考 [\[PB\]常见错误码](#)<sup>20</sup>

## 1.21 [PB] IndexQuery2.0

支持版本：3.50.0

<sup>20</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 1.21.1 简介

通过全局二级索引查询记录，支持SQL语法如附录。使用此接口需要提前在腾讯云控制台或本地Docker版的Web平台针对表添加全局索引字段。

```
POST http://{Tcaplus_REST_URL}
```

### 1.21.2 请求语法

#### 1.21.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 1.21.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.IndexQuery
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf



名称	是否必填	限制条件	说明
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.IndexQuery
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.21.2.3 Data

使用json格式表示记录相关信息。参数信息：

- Query: 必须, 指定查询的SQL语句。

示例：以game\_server\_id索引字段作为 查询条件

```
#示例1,
{
  "Query": "select player_id, player_name,player_email, game_server_id, pay.amount from game_players
where game_server_id>0 limit 1 offset 0"
}
```

#### 1.21.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.IndexQuery' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{"Query": "select player_id, player_name,player_email, game_server_id, pay.amount from game_players where game_server_id>0 limit 1 offset 0"}'
```

#### 1.21.3 返回语法

##### 1.21.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
MultiRecords	返回的记录集合
Record	返回的记录数据, 根据SQL中查询的字段返回
TotalNum	返回匹配的记录条数

### 1.21.3.2 返回示例

#### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "MultiRecords": [{
    "RecordVersion": 1,
    "Record": {
      "game_server_id": 55,
      "pay": {
        "amount": 55
      },
      "player_email": "55",
      "player_id": 5,
      "player_name": "5"
    }
  ]},
  "TotalNum": 1
}
```

#### 失败返回示例

```
{"ErrorCode": -13841, "ErrorMsg": "proxy_err_query_index_field_not_exist"}
```

### 1.21.4 错误

参考 [\[PB\]常见错误码](#)<sup>21</sup>

### 1.21.5 支持的SQL语法

#### 1.21.5.1 条件查询

支持 =, >, >=, <, <=, !=, between, in, not in, like, not like, and, or , 比如:

---

<sup>21</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

```
select * from table where a > 100 and b < 1000;

select * from table where a between 1 and 100 and b < 1000;

select * from table where str like "test";

select * from table where a > 100 or b < 1000;
```

注意：between 查询时，between a and b，对应的查询范围为[a, b]，比如 between 1 and 100, 是会包含 1 和 100 这两个值的，即查询范围为[1,100]。

注意：like 查询是支持模糊匹配，其中"%"通配符，匹配 0 个或者多个字符；“\_”通配符，匹配 1 个字符。

#### 1.21.5.2 分页查询

支持 limit offset 分页查询。比如：

```
select * from table where a > 100 limit 100 offset 0;
```

注意：当前 limit 必须与 offset 搭配使用，即不支持 limit 1 或者 limit 0,1 这种。

#### 1.21.5.3 聚合查询

当前支持的聚合查询包括：sum, count, max, min, avg，比如：

```
select sum(a), count(*), max(a), min(a), avg(a) from table where a > 1000;
```

注意：聚合查询不支持 limit offset，即 limit offset 不生效；

注意：目前只有 count 支持 distinct，即 select count(distinct(a)) from table where a > 1000; 其他情况均不支持 distinct。

#### 1.21.5.4 部分字段查询

```
select a, b from table where a > 1000;
```

对于 pb 表，还支持查询嵌套字段的值，用点分方式，类似：

```
select field1.field2.field3, a, b from table where a > 1000;
```

## 1.21.6

### 不支持的 SQL 语法

#### 1.21.6.1 不支持聚合查询与非聚合查询混用

```
select *, a, b from table where a > 1000;  
  
select sum(a), a, b from table where a > 1000;  
  
select count(*), * from table where a > 1000;
```

#### 1.21.6.2 有限支持Order by查询

```
select * from table where a > 1000 order by a;
```

#### 1.21.6.3 不支持group by查询

```
select * from table where a > 1000 group by a;
```

#### 1.21.6.4 不支持having查询

```
select sum(a) from table where a > 1000 group by a having sum(a) > 10000;
```

#### 1.21.6.5 不支持多表联合查询

```
select * from table1 where table1.a > 1000 and table1.a = table2.b;
```

#### 1.21.6.6 不支持嵌套的SELECT查询

```
select * from table where a > 1000 and b in (select b from table where b < 5000);
```

### 1.21.6.7 不支持别名

```
select sum(a) as sum_a from table where a > 1000;
```

### 1.21.6.8 其它不支持的查询

- 不支持 join 查询；
- 不支持 union 查询；
- 不支持类似 select a+b from table where a > 1000 的查询；
- 不支持类似 select \* from table where a+b > 1000 的查询；
- 不支持类似 select \* from table where a >= b 的查询；
- 不支持其他未提到的查询。

## 1.22 [PB] Traverse2.0

支持版本：3.50.0

### 1.22.1 简介

遍历全表数据。支持Limit参数控制每次遍历返回的记录数，同时返回响应包中包含TraverseSession字段 和 Complete字段，遍历条件：

- Limit: 控制每次遍历返回的记录条数
- TraverseSession: 字符串，初始值: 0.0, 用于每次遍历判断遍历位置，类似于offset作用，本次返回的Session值，传入下次遍历Data参数中
- Complete: bool, 取值范围：False 或 True, 用于判断记录是否遍历完成

```
POST http://{Tcaplus_REST_URL}
```

### 1.22.2 请求语法

#### 1.22.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

## 1.22.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.Traverse
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>• 0: 操作成功后不返回数据</li> <li>• 1: 操作成功后返回和请求一致的数据</li> <li>• 2: 操作成功后返回本次update操作后的数据</li> <li>• 3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>• 1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>• 2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>• 3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.Traverse
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 1.22.2.3 Data

使用json格式表示记录相关信息。参数说明：

- Limit: 可选，限制每次遍历返回的数据条数
- ReturnValues: 可选，用户缓存，用于一些全局对象的传递，避免内存占用管理
- TravesSession: 必须，用于判断遍历循环Session的状态，类似于Offset概念，从响应包的TravesSession中

示例：

```
{
  "Limit": 1000,
  "TraverseSession": "0.0",
  "ReturnValues": "user buffers, to build a bridge between request and response, use case: global object that do not want to save in memory"
}
```

### 1.22.2.4 完整请求示例

一次性遍历示例

Limit设置比较大，一次就返回所有记录。适合表记录数小的场景

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.Traverse' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: game_players' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Limit": 1000,
  "TravesSession": "0.0",
  "ReturnValues": "user buffers, to build a bridge between request and response, use case: global object that do not want to save in memory"
}'
```



### 循环遍历示例

Limit设置适中，同时结合**TraverseSession**和**Complete**来进行循环遍历判断。以Bash脚本举例：

```
#!/bin/bash
PROTO_VERSION="2.0"
TCAPLUS_VERSION="Tcaplus3.50.0"
IDL_TYPE="protobuf"
CHECK_POLICY=3
result_flag=2
rest_url="http://172.17.32.17"
app_id=70
zone_id=1
table_name="game_players"
passwd="Test@2020"

#caculate the md5 value of password
md5_passwd=$(echo -n $passwd | md5sum | cut -d ' ' -f1)

function build_headers(){
    headers=(
        -H "x-tcaplus-target: ${1}"
        -H "x-tcaplus-app-id: ${app_id}"
        -H "x-tcaplus-zone-id: ${zone_id}"
        -H "x-tcaplus-protocol-version: ${PROTO_VERSION}"
        -H "x-tcaplus-table-name: ${table_name}"
        -H "x-tcaplus-pwd-md5: ${md5_passwd}"
        -H "x-tcaplus-result-flag: ${result_flag}"
        -H "x-tcaplus-version: ${TCAPLUS_VERSION}"
        -H "x-tcaplus-data-version-check: ${CHECK_POLICY}"
        -H "x-tcaplus-idl-type: ${IDL_TYPE}"
    )
}

function build_traverse_loop_data(){
    cat <<EOF
{
    "Limit": 1,
    "TraverseSession": "$1",
    "ReturnValues": "user buffers"
}
EOF
}

function traverse_loop() {
    target="Tcaplus.Traverse"
    build_headers "$target"
    complete="False"
    limit=1
    traverse_session="0.0"
    total_num=0
    while [ "$complete" == "False" ];do

        ret=$(curl -i -XPOST "${headers[@]}" "$rest_url" -d "$(build_traverse_loop_data $traverse_session)" |
grep -v HTTP | grep -v "json-1.0" | grep -v "content-length" | grep -v "^$" )

        echo "$ret"

        #获取complete值, False 或true, False: 表示数据尚未结束, true:表示遍历结束. 借助Python3的Json
        complete=$( echo "$ret" | python3 -c "import sys, json; print(json.load(sys.stdin)['Complete'])" )
    done
}
```

```

#获取TraverseSession值, 作为下一次遍历定位使用, 类似于Offset作用.
traverse_session=$( echo "$ret" | python3 -c "import sys, json; print(json.load(sys.stdin)
['TraverseSession'])" )

if [ "$complete" == "False" ];then

    total_num=$((total_num+1))
fi

done
echo "TotalRecords:$total_num"
}

declare -a headers

#traverse
traverse_loop

```

## 1.22.3 返回语法

### 1.22.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据, 随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 1.22.3.2 返回示例

判断Complete参数是否为true, 如果为false, 需要通过循环遍历方式继续请求。

```

{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "ReturnValues": "user buffers, to build a bridge between request and response, use case: global
object that do not want to save in memory",
  "MultiRecords": [{
    "RecordVersion": 1,
    "Record": {
      "player_id": 3,
      "player_name": "1",
      "player_email": "15",
      "game_server_id": 1,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 0,
        "amount": 0,
        "method": 0
      }
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "player_id": 4,
      "player_name": "44",
      "player_email": "44",
      "game_server_id": 44,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,
      "pay": {
        "pay_id": 44,
        "amount": 44,
        "method": 44
      }
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "player_id": 54,
      "player_name": "54",
      "player_email": "54",
      "game_server_id": 54,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 54,
        "amount": 54,
        "method": 54
      }
    }
  }, {
    "RecordVersion": 3,

```

```

    "Record": {
      "player_id": 4,
      "player_name": "4",
      "player_email": "4",
      "game_server_id": 4,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,
      "pay": {
        "pay_id": 4,
        "amount": 4,
        "method": 4
      }
    }
  }, {
    "RecordVersion": 7,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "5",
      "game_server_id": 6,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": false,
      "pay": {
        "pay_id": 5,
        "amount": 5,
        "method": 5
      }
    }
  }, {
    "RecordVersion": 1,
    "Record": {
      "player_id": 5,
      "player_name": "5",
      "player_email": "55",
      "game_server_id": 55,
      "login_timestamp": [],
      "logout_timestamp": [],
      "is_online": false,
      "pay": {
        "pay_id": 55,
        "amount": 55,
        "method": 55
      }
    }
  }, {
    "RecordVersion": 5,
    "Record": {
      "player_id": 6,
      "player_name": "6",
      "player_email": "6",
      "game_server_id": 6,
      "login_timestamp": ["123456", "234"],
      "logout_timestamp": ["123456", "234"],
      "is_online": true,

```

```
        "pay": {
          "pay_id": 6,
          "amount": 6,
          "method": 6
        }
      },
      "TotalNum": 7,
      "Complete": true,
      "TraverseSession": "80614.48569856"
    }
  }
```

### 1.22.4 错误

参考 [\[PB\]常见错误码](#)<sup>22</sup>

---

<sup>22</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 2 RESTful API 2.0 for List表

### 2.1 RESTful API 2.0说明

目前RESTful API 版本已升级到2.0，针对TcaplusDB List表，已开放12个接口，涵盖单笔，批量等方面。List表相比于Generic表，可用于一些特殊场景，其本质是链表结构，通过对链表元素建索引实现类似数组的O(1)时间复杂度访问及删除，队头&队尾O(1)时间复杂度插入。使用场景：

- 保存玩家前N封邮件
- 保存玩家前N条聊天记录

List表使用限制：

- 单个Key对应的List记录元素不超过10000个

### 2.2 RESTful API 使用前置

#### 2.2.1 TcaplusDB环境准备

针对腾讯云环境，直接用腾讯云账号进TcaplusDB控制台进行相关操作即可。

针对TcaplusDB本地Docker环境，部署步骤请参考：[Docker环境部署](#)<sup>23</sup>。

#### 2.2.2 TcaplusDB表创建

针对腾讯云环境，创建TcaplusDB表请参考：[腾讯云环境创建表](#)<sup>24</sup>。

针对TcaplusDB本地Docker环境，创建TcaplusDB表请参考：[Docker环境表创建](#)<sup>25</sup>。

#### 2.2.3 测试机器准备

针对腾讯云环境，请申请一台CVM实例，VPC环境与TcaplusDB的VPC环境保持一致。

针对TcaplusDB本地Docker环境，可以准备一台开发测试机，如果涉及跨机访问Docker环境，需要注意连通策略，同时Docker部署以本地Host模式为主。

#### 2.2.4 TcaplusDB List样例表

表名:tb\_online\_list ,proto文件名:tb\_online\_list.**proto**

<sup>23</sup><https://github.com/tencentyun/tcaplusdb-documents/blob/main/docker/TcaplusDB%E5%85%A5%E9%97%A8-Docker%E9%83%A8%E7%BD%B2%E7%AF%87.md>

<sup>24</sup> <https://cloud.tencent.com/document/product/596/38808>

<sup>25</sup> <https://github.com/tencentyun/tcaplusdb-documents/tree/main/tcapluscli>

```

syntax = "proto3";

package tcapluservice;

import "tcapluservice.optionv1.proto";

message tb_online_list {
    option(tcapluservice.tcaplus_primary_key) = "openid,tconndid,timekey";
    option(tcapluservice.tcaplus_customattr) = "TableType=LIST;ListNum=1900";

    int32 openid = 1; //QQUin
    int32 tconndid = 2;
    string timekey = 3;
    string gamesvrid = 4;
    int32 logintime = 5 ;
    repeated int64 lockid = 6;
    pay_info pay = 7;

    message pay_info {
        uint64 total_money = 1;
        uint64 pay_times = 2;
    }
}

```

## RESTful API 2.0 接口列表

Generic表 RESTful API 2.0主要包括12个接口，有单笔和批量。接口列表如下：

### 2.3 [PB] ListInsertRecord

**支持版本：3.50.0**

#### 2.3.1 简介

在List PB表中插入记录，index表示为该记录分配index值，支持队头和队尾插入，通过参数Index控制。

- Index = -1: 表示队尾插入
- Index = -2: 表示队头插入

```
POST http://{Tcaplus_REST_URL}
```



## 2.3.2 请求语法

### 2.3.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 2.3.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListInsertRecord
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListInsertRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf

名称	是否必填	限制条件	说明
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListInsertRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.3.2.3 Data

使用json格式表示记录相关信息。参数：

- Index: 表示插入位置, -1：队尾, -2：队头

```
{
  "Index": -1,
  "Record": {
    "openid": 10,
    "tconndid": 10,
    "timekey": "10",
    "gamesvrid": "2",
    "logintime": 0,
    "lockid": [],
    "pay": {
      "total_money": 0,
      "pay_times": 1
    }
  }
}
```

#### 2.3.2.4 完整请求示例

注意：示例中相关参数请替换成业务方TcaplusDB实际参数。

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListInsertRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 2' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 3' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.12 -d '{
  "Index": -1,
  "Record": {
    "openid": 10,
    "tconndid": 10,
    "timekey": "10",
    "gamesvrid": "2",
    "logintime": 0,
    "lockid": [],
    "pay": {
      "total_money": 0,
      "pay_times": 1
    }
  }
}'
```

### 2.3.3

## 返回语法

#### 2.3.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

#### 2.3.3.2 返回示例

##### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 998,
  "Index": 1,
  "Record": {
    "openid": 10,
    "tconndid": 10,
    "timekey": "10",
    "gamesvrid": "2",
    "logintime": 0,
    "lockid": [],
    "pay": {
      "total_money": 0,
      "pay_times": 1
    }
  }
}
```

错误返回

```
{
  "ErrorCode": -4109,
  "ErrorMsg": "tcapsvr_fail_invalid_index"
}
```

## 2.3.4 错误码

参考 [\[PB\]常见错误码](#)<sup>26</sup>

## 2.4 [PB] ListBatchInsertRecord

起始版本：3.50.0

### 2.4.1 简介

在List表中批量插入数据，可指定插入方式。通过 Index参数控制插入的位置：

- Index = -1: 表示队尾插入
- Index = -2: 表示队头插入

```
POST http://{Tcaplus_REST_URL}
```

### 2.4.2 请求语法

#### 2.4.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

<sup>26</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 2.4.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListBatchInsertRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListBatchInsertRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListBatchInsertRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.4.2.3 Data

使用json格式表示记录相关信息。Index表示插入的位置：

- Index: 表示插入位置, -1：队尾, -2：队头

```

{
  "MultiRecords": [{
    "Index": -1,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "2",
      "logintime": 2,
      "lockid": [],
      "pay": {
        "total_money": 2,
        "pay_times": 2
      }
    }
  }], {
    "Index": -1,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "3",
      "logintime": 3,
      "lockid": [],
      "pay": {
        "total_money": 3,
        "pay_times": 3
      }
    }
  }], {
    "Index": -1,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "4",
      "logintime": 4,
      "lockid": [],
      "pay": {
        "total_money": 4,
        "pay_times": 4
      }
    }
  }
  }
}]
}

```



#### 2.4.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tpcaplus-target: Tcaplus.ListBatchInsertRecords' -H 'x-tpcaplus-app-id: 70' -H 'x-tpcaplus-zone-id: 1' -H 'x-tpcaplus-protocol-version: 2.0' -H 'x-tpcaplus-table-name: tb_online_list' -H 'x-tpcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tpcaplus-result-flag: 3' -H 'x-tpcaplus-version: Tcaplus3.50.0' -H 'x-tpcaplus-data-version-check: 1' -H 'x-tpcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "MultiRecords": [{
    "Index": -1,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"2",
      "logintime":2,
      "lockid":[],
      "pay": {
        "total_money": 2,
        "pay_times": 2
      }
    }
  }],{
    "Index": -1,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"3",
      "logintime":3,
      "lockid":[],
      "pay": {
        "total_money": 3,
        "pay_times": 3
      }
    }
  }],{
    "Index": -1,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"4",
      "logintime":4,
      "lockid":[],
      "pay": {
        "total_money": 4,
        "pay_times": 4
      }
    }
  }
}]
}'
```

## 2.4.3 返回语法

### 2.4.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 2.4.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 23,
    "Index": 17,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "2",
      "logintime": 2,
      "lockid": [],
      "pay": {
        "total_money": 2,
        "pay_times": 2
      }
    }
  }, {
    "RecordVersion": 23,
    "Index": 18,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "3",
      "logintime": 3,
      "lockid": [],
      "pay": {
        "total_money": 3,
        "pay_times": 3
      }
    }
  }, {
    "RecordVersion": 23,
    "Index": 19,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "4",
      "logintime": 4,
      "lockid": [],
      "pay": {
        "total_money": 4,
        "pay_times": 4
      }
    }
  }
}],
}
```

```

    "SucceedNum": 3
  }

```

## 失败示例

```

{
  "ErrorCode": -1299,
  "ErrorMsg": "api_field_not_exist_error",
  "SucceedRecords": [{
    "RecordVersion": 26,
    "Index": 21,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "3",
      "logintime": 3,
      "lockid": [],
      "pay": {
        "total_money": 3,
        "pay_times": 3
      }
    }
  }, {
    "RecordVersion": 26,
    "Index": 22,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "4",
      "logintime": 4,
      "lockid": [],
      "pay": {
        "total_money": 4,
        "pay_times": 4
      }
    }
  }],
  "SucceedNum": 2,
  "FailedRecords": [{
    "Index": 1
  }],
  "FailedNum": 1
}

```

## 2.4.4 错误码

参考 [\[PB\]常见错误码](#)<sup>27</sup>

## 2.5 [PB] ListGetRecord2.0

起始版本：3.50.0

### 2.5.1 简介

在List PB表中查询对应Key下的指定下标的记录。

- Index: 指定要返回记录的下标。

```
POST http://{Tcaplus_REST_URL}
```

### 2.5.2 请求语法

#### 2.5.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 2.5.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListGetRecord
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

<sup>27</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListGetRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>• 0：操作成功后不返回数据</li> <li>• 1：操作成功后返回和请求一致的数据</li> <li>• 2：操作成功后返回本次update操作后的数据</li> <li>• 3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>• 1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>• 2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>• 3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

## 示例：

```
x-tcaplus-target:Tcaplus.ListGetRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.5.2.3 Data

使用json格式表示记录相关信息。指定要获取记录的主键及对应记录的下标。可以理解为1个主键对应了N条记录，这N条记录的主键是一样的，然后通过指定下标来获取对应位置的完整记录。

- Index: 指定要获取记录的下标
- Record: 指定要获取记录的主键

```
{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1,
  },
  "Index": 0
}
```

### 2.5.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListGetRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1,
  },
  "Index": 0
}'
```

## 2.5.3 返回语法

### 2.5.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	当前记录的版本号
Index	当前记录所在的下标
Record	Json格式, 成功的记录

### 2.5.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 27,
  "Index": 0,
  "Record": {
    "openid": 1,
    "tconndid": 1,
    "timekey": "1",
    "gamesvrid": "2",
    "logintime": 2,
    "lockid": [],
    "pay": {
      "total_money": 2,
      "pay_times": 2
    }
  }
}
```

#### 失败示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```



## 2.5.4 错误码

参考 [\[PB\]常见错误码](#)<sup>28</sup>

## 2.6 [PB] ListBatchGetRecord

起始版本：3.50.0

### 2.6.1 简介

在List PB表中批量查询指定Key下的记录。

- Index: 指定要返回记录的下标。

```
POST http://{Tcaplus_REST_URL}
```

### 2.6.2 请求语法

#### 2.6.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 2.6.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListBatchGetRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

<sup>28</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListBatchGetRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

## 示例：

```
x-tcaplus-target:Tcaplus.ListBatchGetRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.6.2.3 Data

使用json格式表示记录相关信息。指定要获取的Key, 及对应的多条记录的下标,

- Indexes: 必须, 数组, 指定要获取记录的下标, 支持指定多个
- Record: 指定要获取记录的主键

```
{
  "Indexes": [0,1],
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  }
}
```

### 2.6.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListBatchGetRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Indexes": [4,5],
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  }
}'
```

## 2.6.3 返回语法

### 2.6.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	当前记录的版本号
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 2.6.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 28,
    "Index": 4,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "1",
      "logintime": 1,
      "lockid": [],
      "pay": {
        "total_money": 1,
        "pay_times": 1
      }
    }
  }, {
    "RecordVersion": 28,
    "Index": 5,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "2",
      "logintime": 2,
      "lockid": [],
      "pay": {
        "total_money": 2,
        "pay_times": 2
      }
    }
  }],
  "SucceedNum": 2
}
```

## 失败示例

```
{
  "ErrorCode": -1299,
  "ErrorMsg": "api_field_not_exist_error",
  "SucceedRecords": [{
    "RecordVersion": 28,
    "Index": 4,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "1",
      "logintime": 1,
      "lockid": [],
      "pay": {
        "total_money": 1,
        "pay_times": 1
      }
    }
  }],
  "SucceedNum": 1,
  "FailedRecords": [{
    "Index": 0
  }, {
    "Index": 3
  }],
  "FailedNum": 2
}
```

## 2.6.4 错误码

参考 [\[PB\]常见错误码](#)<sup>29</sup>

支持版本：3.50.0

## 2.6.5 1. 简介

在List表中批量获取数据，根据记录主键获取

```
POST http://{Tcaplus_REST_URL}
```

<sup>29</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 2.6.6 2.请求语法

### 2.6.6.1 2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 2.6.6.2 2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListBatchGetRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例:

```
x-tcaplus-target:Tcaplus.ListBatchGetRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:game_players
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.6.6.3 2.3 Data

使用json格式表示记录相关信息。



```

{"MultiRecords": [{
  "RecordVersion": 1,
  "Record": {
    "openid": 200,
    "timekey": "timekey",
    "tconndid": 1,
    "svrid": "svrid",
  }
}, {
  "RecordVersion": 1,
  "Record": {
    "openid": 202,
    "timekey": "timekey",
    "tconndid": 1,
    "svrid": "svrid",
  }
}, {
  "RecordVersion": 1,
  "Record": {
    "openid": 209,
    "timekey": "timekey",
    "tconndid": 1,
    "svrid": "svrid"
  }
}]
}

```

### 2.6.7 3. 返回语法

```

{
  "ErrorCode":errCode,
  "ErrorMsg":errMsg,
  "SucceedRecords":RecordJsonObjs,
  "ReturnValues":RetVal,
  "SucceedNum" : int,
  "SucceedRecords" : RecordJsonObjs,
  "FailedRecords" : RecordJsonObjs,
  "FailedNum":int
}

```

### 2.6.8 5.返回参数

参数名	说明
ErrorCode	返回码

参数名	说明
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

## 2.6.9 6错误

参考 [\[PB\]常见错误码](#)<sup>30</sup>

## 2.6.10 7.示例

## 2.6.11 8.还可以看

原生

- [C++](#)<sup>31</sup>

RESTful

- HTTP
- Java
- PHP
- Python
- Go

CLI

## 2.7 [PB] ListGetAllRecords2.0

起始版本：3.50.0

<sup>30</sup> <https://iwiki.oa.tencent.com/pages/viewpage.action?pagelId=237730186>

<sup>31</sup> <https://iwiki.oa.tencent.com/pages/viewpage.action?pagelId=237725396>

## 2.7.1 简介

在List PB表中查询对应Key下的所有记录, 通过 Limit和Offset限制每次返回的记录条数。

- Limit: 限制返回的记录条数, 如果Limit为-1则返回所有记录数
- Offset: 指定记录偏移量

```
POST http://{Tcaplus_REST_URL}
```

## 2.7.2 请求语法

### 2.7.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 2.7.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListGetAllRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListGetAllRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0

名称	是否必填	限制条件	说明
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListGetAllRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.7.2.3 Data

使用json格式表示记录相关信息。指定

- Offset: 指定获取记录的偏移量
- Limit: 指定每次返回记录的条数, 为-1表示返回所有记录数
- Record: 指定要获取记录的主键

```
{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  },
  "Offset": 0,
  "Limit": -1
}
```

#### 2.7.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListGetAllRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  },
  "Offset": 0,
  "Limit": -1
}'
```

### 2.7.3 返回语法

#### 2.7.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	当前记录的版本号
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分

参数名	说明
FailedNum	失败的记录数

### 2.7.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [
    {
      "RecordVersion": 27,
      "Index": 3,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 4,
        "logintime": 4,
        "lockid": [],
        "pay": {
          "total_money": 4,
          "pay_times": 4
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 4,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 1,
        "logintime": 1,
        "lockid": [],
        "pay": {
          "total_money": 1,
          "pay_times": 1
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 5,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 2,
        "logintime": 2,
        "lockid": [],
        "pay": {
          "total_money": 2,
          "pay_times": 2
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 6,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 3,
        "logintime": 3,
        "lockid": [],
        "pay": {
          "total_money": 3,
          "pay_times": 3
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 7,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 4,
        "logintime": 4,
        "lockid": [],
        "pay": {
          "total_money": 4,
          "pay_times": 4
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 8,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 1,
        "logintime": 1,
        "lockid": [],
        "pay": {
          "total_money": 1,
          "pay_times": 1
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 9,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 2,
        "logintime": 2,
        "lockid": [],
        "pay": {
          "total_money": 2,
          "pay_times": 2
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 10,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 3,
        "logintime": 3,
        "lockid": [],
        "pay": {
          "total_money": 3,
          "pay_times": 3
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 11,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 4,
        "logintime": 4,
        "lockid": [],
        "pay": {
          "total_money": 4,
          "pay_times": 4
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 13,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 2,
        "logintime": 2,
        "lockid": [],
        "pay": {
          "total_money": 2,
          "pay_times": 2
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 14,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 3,
        "logintime": 3,
        "lockid": [],
        "pay": {
          "total_money": 3,
          "pay_times": 3
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 15,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 4,
        "logintime": 4,
        "lockid": [],
        "pay": {
          "total_money": 4,
          "pay_times": 4
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 16,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 1,
        "logintime": 1,
        "lockid": [],
        "pay": {
          "total_money": 1,
          "pay_times": 1
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 17,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 2,
        "logintime": 2,
        "lockid": [],
        "pay": {
          "total_money": 2,
          "pay_times": 2
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 18,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 3,
        "logintime": 3,
        "lockid": [],
        "pay": {
          "total_money": 3,
          "pay_times": 3
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 19,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 4,
        "logintime": 4,
        "lockid": [],
        "pay": {
          "total_money": 4,
          "pay_times": 4
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 20,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 1,
        "logintime": 1,
        "lockid": [],
        "pay": {
          "total_money": 1,
          "pay_times": 1
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 21,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 3,
        "logintime": 3,
        "lockid": [],
        "pay": {
          "total_money": 3,
          "pay_times": 3
        }
      }
    },
    {
      "RecordVersion": 27,
      "Index": 22,
      "Record": {
        "openid": 1,
        "tconndid": 1,
        "timekey": 1,
        "gamesvrid": 4,
        "logintime": 4,
        "lockid": [],
        "pay": {
          "total_money": 4,
          "pay_times": 4
        }
      }
    }
  ],
  "SucceedNum": 19
}
```

#### 失败示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

## 2.7.4 错误码

参考 [PB]常见错误码<sup>32</sup>

## 2.8 [PB] ListDeleteRecord2.0

起始版本：3.50.0

### 2.8.1 简介

在List PB表中删除对应Key下的指定下标的记录。

- Index: 指定要删除记录的下标。

```
POST http://{Tcaplus_REST_URL}
```

### 2.8.2 请求语法

#### 2.8.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

#### 2.8.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListDeleteRecord
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

<sup>32</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListDeleteRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListDeleteRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```



### 2.8.2.3 Data

使用json格式表示记录相关信息。指定要删除记录的主键及对应记录的下标。

- Index: 指定要删除记录的下标
- Record: 指定要删除记录的主键

```
{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1,
  },
  "Index": 0
}
```

### 2.8.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListDeleteRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Index": 3,
  "Record": {
    "openid":1,
    "tconndid":1,
    "timekey": "1"
  }
}'
```

## 2.8.3 返回语法

### 2.8.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	当前记录的版本号

参数名	说明
Index	当前记录所在的下标
Record	Json格式, 成功的记录

### 2.8.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": -1,
  "Index": 3,
  "Record": {
    "openid": 1,
    "tconndid": 1,
    "timekey": "1",
    "gamesvrid": "",
    "logintime": 0,
    "lockid": [],
    "pay": {
      "total_money": 0,
      "pay_times": 0
    }
  }
}
```

#### 失败示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

## 2.8.4 错误码

参考 [\[PB\]常见错误码](#)<sup>33</sup>

<sup>33</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 2.9 [PB] ListBatchDeleteRecords2.0

起始版本：3.50.0

### 2.9.1 简介

在List PB表中批量删除指定Key的记录，支持同时删除多个下标的记录

- Indexex: 指定要删除记录的下标。

```
POST http://{Tcaplus_REST_URL}
```

### 2.9.2 请求语法

#### 2.9.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80  
http://172.17.0.22  
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001  
http://9.135.8.93:31001
```

#### 2.9.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListBatchDeleteRecords  
x-tcaplus-app-id:{string}  
x-tcaplus-zone-id:{string}  
x-tcaplus-protocol-version:{string}  
x-tcaplus-table-name:{string}  
x-tcaplus-pwd-md5:{string}  
x-tcaplus-result-flag:result_flag  
x-tcaplus-version:{string}  
x-tcaplus-data-version-check:{string}  
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListBatchDeleteRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)

名称	是否必填	限制条件	说明
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0: 操作成功后不返回数据</li> <li>1: 操作成功后返回和请求一致的数据</li> <li>2: 操作成功后返回本次update操作后的数据</li> <li>3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListBatchDeleteRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.9.2.3 Data

使用json格式表示记录相关信息。指定要获取的Key, 及对应的多条记录的下标,

- Indexes: 必须, 数组, 指定要获取记录的下标, 支持指定多个

- Record: 指定要获取记录的主键

```
{
  "Indexes": [4,5],
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  }
}
```

#### 2.9.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListBatchDeleteRecords' -H 'x-tcaplus-app-id: 70' -H 'x-
tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-
tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version:
Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17
-d '{
  "Record": {
    "openid":1,
    "tconndid":1,
    "timekey": "1"
  },
  "Indexes": [4,5]
}'
```

### 2.9.3 返回语法

#### 2.9.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	当前记录的版本号
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分

参数名	说明
FailedNum	失败的记录数

### 2.9.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 30,
    "Index": 4,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "1",
      "logintime": 1,
      "lockid": [],
      "pay": {
        "total_money": 1,
        "pay_times": 1
      }
    }
  }, {
    "RecordVersion": 30,
    "Index": 5,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "2",
      "logintime": 2,
      "lockid": [],
      "pay": {
        "total_money": 2,
        "pay_times": 2
      }
    }
  }],
  "SucceedNum": 2
}
```

失败示例

```
{
  "ErrorCode": -4109,
  "ErrorMsg": "tcapsvr_fail_invalid_index"
}
```

## 2.9.4 错误码

参考 [\[PB\]常见错误码](#)<sup>34</sup>

## 2.10 [PB] ListDeleteAllRecords2.0

起始版本：3.50.0

### 2.10.1 简介

在List PB表中删除对应Key下的所有记录。

```
POST http://{Tcaplus_REST_URL}
```

### 2.10.2 请求语法

#### 2.10.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

<sup>34</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 2.10.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListDeleteAllRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

#### 参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListDeleteAllRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>



名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListDeleteAllRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.10.2.3 Data

使用json格式表示记录相关信息。指定

- Record: 指定要删除记录的主键

```
{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  }
}
```

#### 2.10.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListGetAllRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Record": {
    "openid": 1,
    "timekey": "1",
    "tconndid": 1
  }
}'
```

#### 2.10.3 返回语法

##### 2.10.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
RecordVersion	当前记录的版本号
TotalNum	删除的记录条数

##### 2.10.3.2 返回示例

成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": -1,
  "TotalNum": 16
}
```

失败示例

```
{
  "ErrorCode": 261,
  "ErrorMsg": "Record does not exist"
}
```

## 2.10.4 错误码

参考 [\[PB\]常见错误码](#)<sup>35</sup>

起始版本：3.50.0

### 2.10.5 1. 简介

在List PB表中 批量删除对应key的记录，删除该key下index数组中的记录

```
POST http://{Tcaplus_REST_URL}
```

## 2.10.6 2.请求语法

### 2.10.6.1 2.1 Http请求

```
http://9.135.8.93:31001
```

### 2.10.6.2 2.2 Http头

```
x-tcaplus-target:Tcaplus.ListDeleteAllRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

<sup>35</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 2.10.6.3 2.3 Data

使用json格式表示记录相关信息。

```
{
  "Record": {
    "openid": 103,
    "timekey": "timekey",
    "tconndid": 100,
    "svrid": "svrid"
  },
  "Indexes": [0]
}
```

## 2.10.7 3. 请求参数

### 2.10.7.1 3.1 Http头参数

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListDeleteAllRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应app编号
x-tcaplus-zone-id	是	无	对应zone编号
x-tcaplus-protocol-version	是	无	对应protocol版本号
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	MD5 of AppKey(Password)
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时，该版本号才会自增</li> <li>2：不检测记录版本号，强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号，将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

## 2.10.8

## 4.返回语法

```
{
  "ErrorCode":errCode,
  "ErrorMsg":errMsg,
  "SucceedRecords":RecordJsonObjs,
  "ReturnValues":RetVal,
  "SucceedNum":int,
  "SucceedRecords":RecordJsonObjs,
  "FailedRecords":RecordJsonObjs,
  "FailedNum":int
}
```

## 2.10.9 5.返回参数

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

### 2.10.10 6.错误

参考 [\[PB\]常见错误码](#)<sup>36</sup>

### 2.10.11 7.示例

### 2.10.12 8. 还可以看

原生

- [C++](#)<sup>37</sup>

RESTful

- HTTP
- Java
- PHP
- Python
- Go

CLI

## 2.11 [PB] ListUpdateRecord

支持版本：3.50.0

### 2.11.1 简介

在List PB表中更新记录，index指定要更新的记录下标。

```
POST http://{Tcaplus_REST_URL}
```

---

<sup>36</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

<sup>37</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237725396>

## 2.11.2 请求语法

### 2.11.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

### 2.11.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListUpdateRecord
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListUpdateRecord
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf

名称	是否必填	限制条件	说明
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListUpdateRecord
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.11.2.3 Data

使用json格式表示记录相关信息。参数：

- Index: 表示更新记录的下标



```
{
  "Index": 0,
  "Record": {
    "openid": 1,
    "tconndid": 1,
    "timekey": "1",
    "gamesvrid": "2",
    "logintime": 1,
    "lockid": [],
    "pay": {
      "total_money": 1,
      "pay_times": 1
    }
  }
}
```

#### 2.11.2.4 完整请求示例

注意：示例中相关参数请替换成业务方TcaplusDB实际参数。

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListUpdateRecord' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Index": 0,
  "Record": {
    "openid": 1,
    "tconndid": 1,
    "timekey": "1",
    "gamesvrid": "2",
    "logintime": 1,
    "lockid": [],
    "pay": {
      "total_money": 1,
      "pay_times": 1
    }
  }
}'
```

### 2.11.3 返回语法

#### 2.11.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据，随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式，成功的记录返回数据，详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式，失败的记录返回数据，详见文中Data部分
FailedNum	失败的记录数

#### 2.11.3.2 返回示例

##### 成功返回示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "RecordVersion": 5,
  "Index": 0,
  "Record": {
    "openid": 1,
    "tconndid": 1,
    "timekey": "1",
    "gamesvrid": "2",
    "logintime": 1,
    "lockid": [],
    "pay": {
      "total_money": 1,
      "pay_times": 1
    }
  }
}
```

错误返回

```
{
  "ErrorCode": -4109,
  "ErrorMsg": "tcapsvr_fail_invalid_index"
}
```

#### 2.11.4 错误码

参考 [\[PB\]常见错误码](#)<sup>38</sup>

## 2.12 [PB] ListBatchUpdateRecord

起始版本：3.50.0

### 2.12.1 简介

在List表中批量更新数据，指定更新记录的下标。

```
POST http://{Tcaplus_REST_URL}
```

### 2.12.2 请求语法

#### 2.12.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80
http://172.17.0.22
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001
http://9.135.8.93:31001
```

<sup>38</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 2.12.2.2 Http头

```
x-tcaplus-target:Tcaplus.ListBatchUpdateRecords
x-tcaplus-app-id:{string}
x-tcaplus-zone-id:{string}
x-tcaplus-protocol-version:{string}
x-tcaplus-table-name:{string}
x-tcaplus-pwd-md5:{string}
x-tcaplus-result-flag:result_flag
x-tcaplus-version:{string}
x-tcaplus-data-version-check:{string}
x-tcaplus-idl-type:protobuf
```

参数说明：

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListBatchUpdateRecords
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>0：操作成功后不返回数据</li> <li>1：操作成功后返回和请求一致的数据</li> <li>2：操作成功后返回本次update操作后的数据</li> <li>3：操作成功后返回tcapsvr端操作前的数据</li> </ul>

名称	是否必填	限制条件	说明
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>1：检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>2：不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>3：不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListBatchUpdateRecords
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.12.2.3 Data

使用json格式表示记录相关信息。

- Index: 要更新记录的下标

```
{
  "MultiRecords": [{
    "Index": 1,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"11",
      "logintime":1,
      "lockid":[],
      "pay": {
        "total_money": 1,
        "pay_times": 1
      }
    }
  }],{
    "Index": 2,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"22",
      "logintime":22,
      "lockid":[],
      "pay": {
        "total_money": 22,
        "pay_times": 22
      }
    }
  }
]
```

### 2.12.2.4 完整请求示例

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListBatchUpdateRecords' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "MultiRecords": [{
    "Index": 1,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"11",
      "logintime":1,
      "lockid":[],
      "pay": {
        "total_money": 1,
        "pay_times": 1
      }
    }
  }],{
    "Index": 2,
    "Record": {
      "openid":1,
      "tconndid":1,
      "timekey":"1",
      "gamesvrid":"22",
      "logintime":22,
      "lockid":[],
      "pay": {
        "total_money": 22,
        "pay_times": 22
      }
    }
  }
}]
}'
```

## 2.12.3 返回语法

### 2.12.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息

参数名	说明
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 2.12.3.2 返回示例

#### 成功示例

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "SucceedRecords": [{
    "RecordVersion": 7,
    "Index": 1,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "11",
      "logintime": 1,
      "lockid": [],
      "pay": {
        "total_money": 1,
        "pay_times": 1
      }
    }
  }, {
    "RecordVersion": 7,
    "Index": 2,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "22",
      "logintime": 22,
      "lockid": [],
      "pay": {
        "total_money": 22,
        "pay_times": 22
      }
    }
  }],
  "SucceedNum": 2
}
```



## 失败示例

```
{
  "ErrorCode": -1299,
  "ErrorMsg": "api_field_not_exist_error",
  "SucceedRecords": [{
    "RecordVersion": 26,
    "Index": 1,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "3",
      "logintime": 3,
      "lockid": [],
      "pay": {
        "total_money": 3,
        "pay_times": 3
      }
    }
  }, {
    "RecordVersion": 26,
    "Index": 2,
    "Record": {
      "openid": 1,
      "tconndid": 1,
      "timekey": "1",
      "gamesvrid": "4",
      "logintime": 4,
      "lockid": [],
      "pay": {
        "total_money": 4,
        "pay_times": 4
      }
    }
  }],
  "SucceedNum": 2,
  "FailedRecords": [{
    "Index": 8
  }],
  "FailedNum": 1
}
```

## 2.12.4 错误码

参考 [\[PB\]常见错误码](#)<sup>39</sup>

<sup>39</sup> <https://wiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

## 2.13 [PB] ListTraverse2.0

支持版本：3.50.0

### 2.13.1 简介

遍历全表数据。支持Limit参数控制每次遍历返回的记录数，同时返回响应包中包含TraverseSession字段 和 Complete字段，遍历条件：

- **Limit**: 控制每次遍历返回的记录条数
- **TraverseSession**: 字符串，初始值: 0.0, 用于每次遍历判断遍历位置，游标参数，本次返回的Session值，传入下次遍历Data参数中
- **Complete**: bool, 取值范围：False 或 True, 用于判断记录是否遍历完成
- **Select**: 指定每次遍历返回的字段

```
POST http://{Tcaplus_REST_URL}
```

### 2.13.2 请求语法

#### 2.13.2.1 Http请求

```
#腾讯云控制台 RESTful Endpoint, ip:80, 端口默认80  
http://172.17.0.22  
#TcaplusDB本地docker版 RESTful Endpoint, ip:31001, 端口默认31001  
http://9.135.8.93:31001
```

## 2.13.2.2 Http头

名称	是否必填	限制条件	说明
x-tcaplus-target	是	无	Tcaplus.ListTraverse
x-tcaplus-version	是	无	Tcaplus3.50.0
x-tcaplus-app-id	是	无	对应业务id编号(aka,集群接入id)
x-tcaplus-zone-id	是	无	对应zone编号(aka,表格组id)
x-tcaplus-protocol-version	是	无	对应protocol版本号,默认2.0
x-tcaplus-table-name	是	无	对应表名
x-tcaplus-pwd-md5	是	无	业务密码(aka,集群访问密码), 传入计算后的md5值
x-tcaplus-idl-type	是	无	protobuf
x-tcaplus-result-flag	否	无	<ul style="list-style-type: none"> <li>• 0: 操作成功后不返回数据</li> <li>• 1: 操作成功后返回和请求一致的数据</li> <li>• 2: 操作成功后返回本次update操作后的数据</li> <li>• 3: 操作成功后返回tcapsvr端操作前的数据</li> </ul>
x-tcaplus-data-version-check	否		<ul style="list-style-type: none"> <li>• 1: 检测记录版本号,只有当该版本号与服务器端的版本号相同时, 该版本号才会自增</li> <li>• 2: 不检测记录版本号, 强制把客户端的记录版本号写入到服务器中</li> <li>• 3: 不检测记录版本号, 将服务器端的版本号自增</li> </ul>
x-tcaplus-data-version	否		具体的version值

示例：

```
x-tcaplus-target:Tcaplus.ListTraverse
x-tcaplus-app-id:3
x-tcaplus-zone-id:1
x-tcaplus-protocol-version:2.0
x-tcaplus-table-name:tb_online_list
x-tcaplus-pwd-md5:4e81984efccfb4982333aeb1ff7968d5
x-tcaplus-result-flag:2
x-tcaplus-version:Tcaplus3.50.0
x-tcaplus-data-version-check: 3
x-tcaplus-idl-type:protobuf
```

### 2.13.2.3 Data

使用json格式表示记录相关信息。参数说明：

- Limit: 可选，限制每次遍历返回的数据条数
- Travesesession: 必须，用于判断遍历循环Session的状态，类似于Offset概念，从响应包的Travesesession中
- Select: 可选，指定返回的记录字段

示例：

```
{
  "Limit": 1000,
  "Travesesession": "0.0",
  "Select": ["gamesvrid"]
}
```

### 2.13.2.4 完整请求示例

一次性遍历示例

Limit设置比较大，一次就返回所有记录。适合表记录数小的场景

```
curl -i -XPOST -H 'x-tcaplus-target: Tcaplus.ListTraverse' -H 'x-tcaplus-app-id: 70' -H 'x-tcaplus-zone-id: 1' -H 'x-tcaplus-protocol-version: 2.0' -H 'x-tcaplus-table-name: tb_online_list' -H 'x-tcaplus-pwd-md5: 0972ad76decf4d11a69e2e0d9af335da' -H 'x-tcaplus-result-flag: 1' -H 'x-tcaplus-version: Tcaplus3.50.0' -H 'x-tcaplus-data-version-check: 1' -H 'x-tcaplus-idl-type: protobuf' http://172.17.32.17 -d '{
  "Limit": 1,
  "Travesesession": "0.0",
  "Select": ["gamesvrid"]
}'
```

### 循环遍历示例

Limit设置适中，同时结合[TraverseSession](#)和[Complete](#)来进行循环遍历判断。以Bash脚本举例：

```
#!/bin/bash
PROTO_VERSION="2.0"
TCAPLUS_VERSION="Tcaplus3.50.0"
IDL_TYPE="protobuf"
CHECK_POLICY=3
result_flag=2
rest_url="http://172.17.32.17"
app_id=70
zone_id=1
table_name="tb_online_list"
passwd="Test@2020"

#caculate the md5 value of password
md5_passwd=$(echo -n $passwd | md5sum | cut -d ' ' -f1)

function build_headers(){
    headers=(
        -H "x-tcaplus-target: ${1}"
        -H "x-tcaplus-app-id: ${app_id}"
        -H "x-tcaplus-zone-id: ${zone_id}"
        -H "x-tcaplus-protocol-version: ${PROTO_VERSION}"
        -H "x-tcaplus-table-name: ${table_name}"
        -H "x-tcaplus-pwd-md5: ${md5_passwd}"
        -H "x-tcaplus-result-flag: ${result_flag}"
        -H "x-tcaplus-version: ${TCAPLUS_VERSION}"
        -H "x-tcaplus-data-version-check: ${CHECK_POLICY}"
        -H "x-tcaplus-idl-type: ${IDL_TYPE}"
    )
}

function build_traverse_loop_data(){
    cat <<EOF
{
    "Limit": 1,
    "TraverseSession": "$1",
    "Select": ["gamesvrid"]
}
EOF
}

function traverse_loop() {
    target="Tcaplus.Traverse"
    build_headers "$target"
    complete="False"
    limit=1
    traverse_session="0.0"
    total_num=0
    while [ "$complete" == "False" ];do

        ret=$(curl -i -XPOST "${headers[@]}" "$rest_url" -d "$(build_traverse_loop_data $traverse_session)" |
grep -v HTTP | grep -v "json-1.0" | grep -v "content-length" | grep -v "^$" )

        echo "$ret"

        #获取complete值, False 或true, False: 表示数据尚未结束, true:表示遍历结束. 借助Python3的Json
        complete=$( echo "$ret" | python3 -c "import sys, json; print(json.load(sys.stdin)['Complete'])" )
    done
}
```

```

#获取TraverseSession值, 作为下一次遍历定位使用, 类似于Offset作用.
traverse_session=$( echo "$ret" | python3 -c "import sys, json; print(json.load(sys.stdin)
['TraverseSession'])" )

if [ "$complete" == "False" ];then

    total_num=$((total_num+1))
fi

done
echo "TotalRecords:$total_num"
}

declare -a headers

#traverse
traverse_loop

```

## 2.13.3 返回语法

### 2.13.3.1 返回参数说明

参数名	说明
ErrorCode	返回码
ErrorMsg	返回信息
ReturnValues	用户设置的保留数据, 随请求到达 tcaplus 并由应答原样带回
SucceedRecords	json格式, 成功的记录返回数据, 详见文中Data部分
SucceedNum	成功的记录数
FailedRecords	json格式, 失败的记录返回数据, 详见文中Data部分
FailedNum	失败的记录数

### 2.13.3.2 返回示例

判断Complete参数是否为true, 如果为false, 需要通过循环遍历方式继续请求。

一次性返回数据示例

Limit设置比较大的值

```

{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "MultiRecords": [{
    "RecordVersion": 3,
    "Index": 0,
    "Record": {
      "gamesvrid": "4",
      "openid": 4,
      "tconndid": 4,
      "timekey": "4"
    }
  }, {
    "RecordVersion": 3,
    "Index": 1,
    "Record": {
      "gamesvrid": "2",
      "openid": 2,
      "tconndid": 2,
      "timekey": "2"
    }
  }, {
    "RecordVersion": 3,
    "Index": 2,
    "Record": {
      "gamesvrid": "3",
      "openid": 3,
      "tconndid": 3,
      "timekey": "3"
    }
  }, {
    "RecordVersion": 7,
    "Index": 0,
    "Record": {
      "gamesvrid": "2",
      "openid": 1,
      "tconndid": 1,
      "timekey": "1"
    }
  }, {
    "RecordVersion": 7,
    "Index": 1,
    "Record": {
      "gamesvrid": "11",
      "openid": 1,
      "tconndid": 1,
      "timekey": "1"
    }
  }, {
    "RecordVersion": 7,
    "Index": 2,
    "Record": {
      "gamesvrid": "22",
      "openid": 1,
      "tconndid": 1,

```



```
        "timekey": "1"
      }
    }, {
      "RecordVersion": 7,
      "Index": 3,
      "Record": {
        "gamesvrid": "4",
        "openid": 1,
        "tconndid": 1,
        "timekey": "1"
      }
    }
  ],
  "TotalNum": 7,
  "Complete": true,
  "TraverseSession": "80630.48568512"
}
```

### 循环返回数据示例

单次Limit设置比较小 (Limit=1), 通过 循环遍历返回的结果, 下图所示, 2次遍历返回所有记录, 总记录条数和上面一次性返回的记录数保持一致, TotalNum都为7。

#Loop-第一次返回结果, TotalNum:3

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "MultiRecords": [{
    "RecordVersion": 3,
    "Index": 0,
    "Record": {
      "gamesvrid": "4",
      "openid": 4,
      "tconndid": 4,
      "timekey": "4"
    }
  }, {
    "RecordVersion": 3,
    "Index": 1,
    "Record": {
      "gamesvrid": "2",
      "openid": 2,
      "tconndid": 2,
      "timekey": "2"
    }
  }, {
    "RecordVersion": 3,
    "Index": 2,
    "Record": {
      "gamesvrid": "3",
      "openid": 3,
      "tconndid": 3,
      "timekey": "3"
    }
  }],
  "TotalNum": 3,
  "Complete": false,
  "TraverseSession": "80630.48566720"
}
```

#Loop-第二次返回结果, TotalNum:4

```
{
  "ErrorCode": 0,
  "ErrorMsg": "Succeed",
  "MultiRecords": [{
    "RecordVersion": 7,
    "Index": 0,
    "Record": {
      "gamesvrid": "2",
      "openid": 1,
      "tconndid": 1,
      "timekey": "1"
    }
  }, {
    "RecordVersion": 7,
    "Index": 1,
    "Record": {
```

```

        "gamesvrid": "11",
        "openid": 1,
        "tconndid": 1,
        "timekey": "1"
    }
}, {
    "RecordVersion": 7,
    "Index": 2,
    "Record": {
        "gamesvrid": "22",
        "openid": 1,
        "tconndid": 1,
        "timekey": "1"
    }
}, {
    "RecordVersion": 7,
    "Index": 3,
    "Record": {
        "gamesvrid": "4",
        "openid": 1,
        "tconndid": 1,
        "timekey": "1"
    }
}],
"TotalNum": 4,
"Complete": false,
"TraverseSession": "80630.48568384"
}

```

## 2.13.4 错误

参考 [\[PB\]常见错误码](#)<sup>40</sup>

<sup>40</sup> <https://iwiki.oa.tencent.com/pages/viewpage.action?pageId=237730186>

### 3 [PB]常见错误码

常见的错误码列表, 其它错误码用户可查看 Response 返回的 ErrMsg 属性看具体错误类型：

编号	错误码	描述
1	-1792	表处于只读模式, 请检查 RCU,
3	261	该记录不存在
4	-525	batchget 操作请求超时,
5	-781	batchget,
6	-1037	系统繁忙, 请联系管理员
7	-1293	记录已存在, 请不要重复插入
8	-1549	访问的表字段不存在
9	-2061	表字段类型错误
10	-3085	SetFieldName 操作指定了错误的字段
11	-3341	字段值大小超过其定义类型的限制
12	-4109	list 数据类型元素下标超过范围
14	-4621	请求缺少主键字段或索引字段
15	-6157	list 表元素个数超过定义范围,请设置元素淘汰
16	-6925	result_flag 设置错误, 请参考 SDK 中 result_flag 说明
17	-7949	请检查乐观锁, 请求记录版本号与实际记录版本号不一致
18	-11277	操作表的方法不存在
19	-16141	PB 表 GetRecord 操作失败, 请联系管理员
20	-16397	PB 表非主键字段值超过限定大小(256KB)
21	-16653	PB 表 FieldSetRecord 操作失败, 请联系管理员
22	-16909	PB 表 FieldIncRecord 操作失败, 请联系管理员
23	-275	主键字段个数超过限制, Generic 表限制数为 4,
24	-531	非主键字段个数超过限制, Generic 表限制数为 128,

编号	错误码	描述
25	-787	字段名称大小超过限制（32B）
26	-1043	字段值超过限制（256KB）
27	-1555	字段值的数据类型与其定义类型不匹配
28	-5395	请求中缺少主键
29	-9235	index 不存在
30	-12307	请求发送失败，网络过载，请联系管理员。
31	-12819	表不存在
32	-13843	后台网络异常，请求无法发送成功，如持续存在请联系管理员
33	-14099	插入的记录超过大小限制（1MB）
34	- 6673	请求参数无主键
35	- 6929	请求参数缺少主键字段
36	-4877	需要签名
37	-4365	超出最大field数
38	-34565	索引不存在
39	-3597	非法操作
40	-1805	超出最大行数
41	-2067	操作类型不匹配
42	-11539	非法请求 详细原因请参考返回信息字段
43	-279	鉴权错误
44	-2579	数据反序列化错误
45	-1811	非法参数
46	-2323	数据序列化错误
47		