

数据库智能管家 DBbrain

常见问题

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

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

【商标声明】

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

【服务声明】

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

文档目录

常见问题

MySQL 性能问题

产品常见问题

常见问题

MySQL 性能问题

最近更新时间：2022-07-31 17:03:10

功能说明

App 管理员可以根据群组 ID 获取群组成员的资料。

接口调用说明

适用的群组类型

群组类型 ID	是否支持此 REST API
Private	支持，同新版本中的 Work（好友工作群）
Public	支持
ChatRoom	支持，同新版本中的 Meeting（临时会议群）
AVChatRoom	不支持
Community（社群）	支持，使用 Next 字段分批获取

即时通信 IM 内置上述群组类型，详情介绍请参见 [群组系统](#)。

说明：

因 Community（社群）人数较多，分页获取方式改用 Next 分批方法。

请求 URL 示例



```
https://xxxxxx/v4/group_open_http_svc/get_group_member_info?sdkappid=88888888&ident
```

请求参数说明

下表仅列出调用本接口时涉及修改的参数及其说明，更多参数详情请参考 [REST API 简介](#)。

参数	说明
https	请求协议为 HTTPS，请求方式为 POST
xxxxxx	SDKAppID 所在国家/地区对应的专属域名

	中国： <code>console.tim.qq.com</code> 新加坡： <code>adminapisgp.im.qcloud.com</code> 首尔： <code>adminapikr.im.qcloud.com</code> 法兰克福： <code>adminapiger.im.qcloud.com</code> 印度： <code>adminapiind.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_group_member_info</code>	请求接口
<code>sdkappid</code>	创建应用时即时通信 IM 控制台分配的 SDKAppID
<code>identifier</code>	必须为 App 管理员帐号，更多详情请参见 App 管理员
<code>usersig</code>	App 管理员帐号生成的签名，具体操作请参见 生成 UserSig
<code>random</code>	请输入随机的32位无符号整数，取值范围0 - 4294967295
<code>contenttype</code>	请求格式固定值为 <code>json</code>

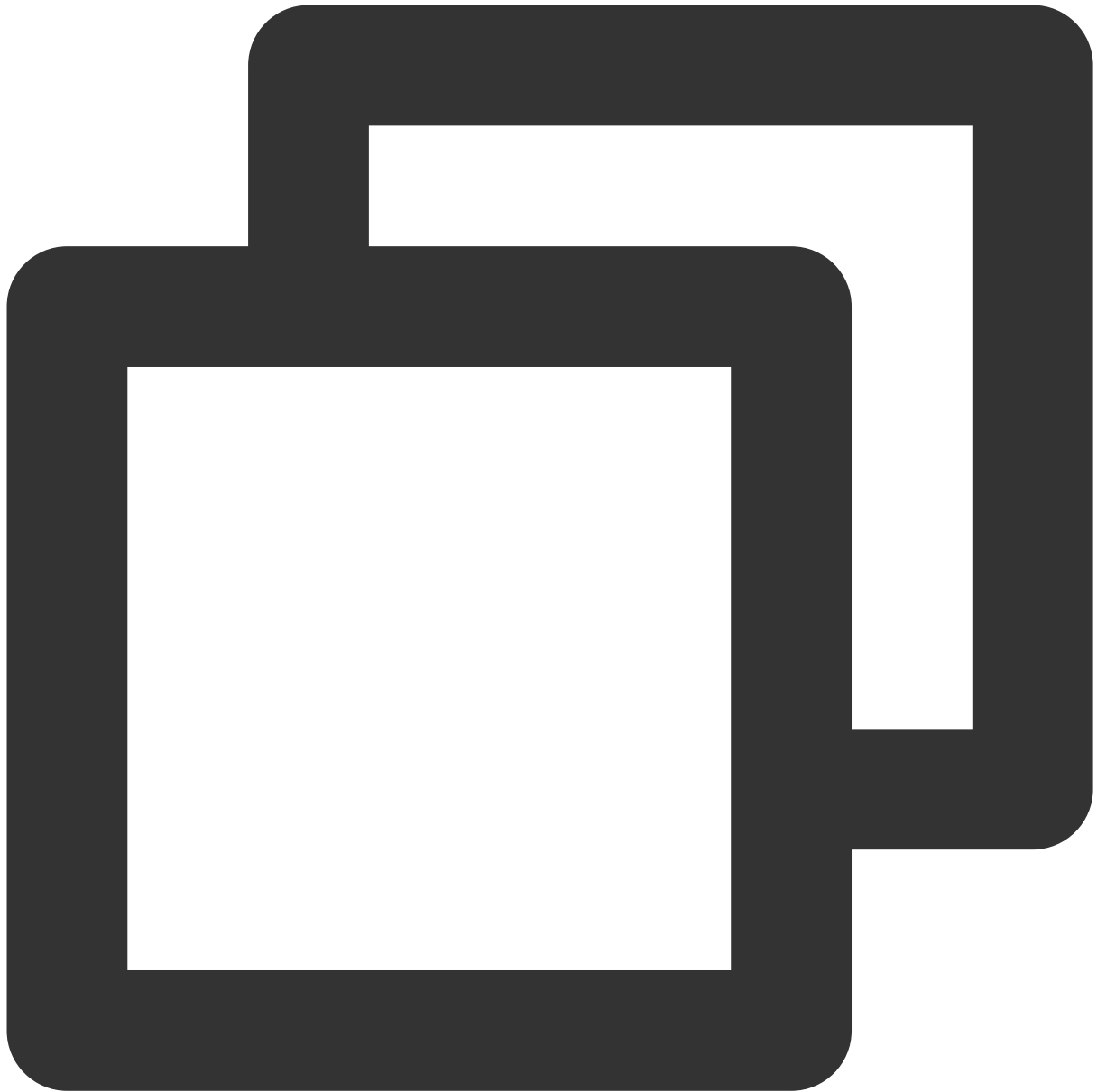
最高调用频率

200次/秒。

请求包示例

基础形式

用来获取群成员详细信息（群成员资料和群成员维度自定义字段），请求中只包含群 ID。



```
{
  "GroupId": "@TGS#1NVTZEAE4" // 群组 ID (必填)
}
```

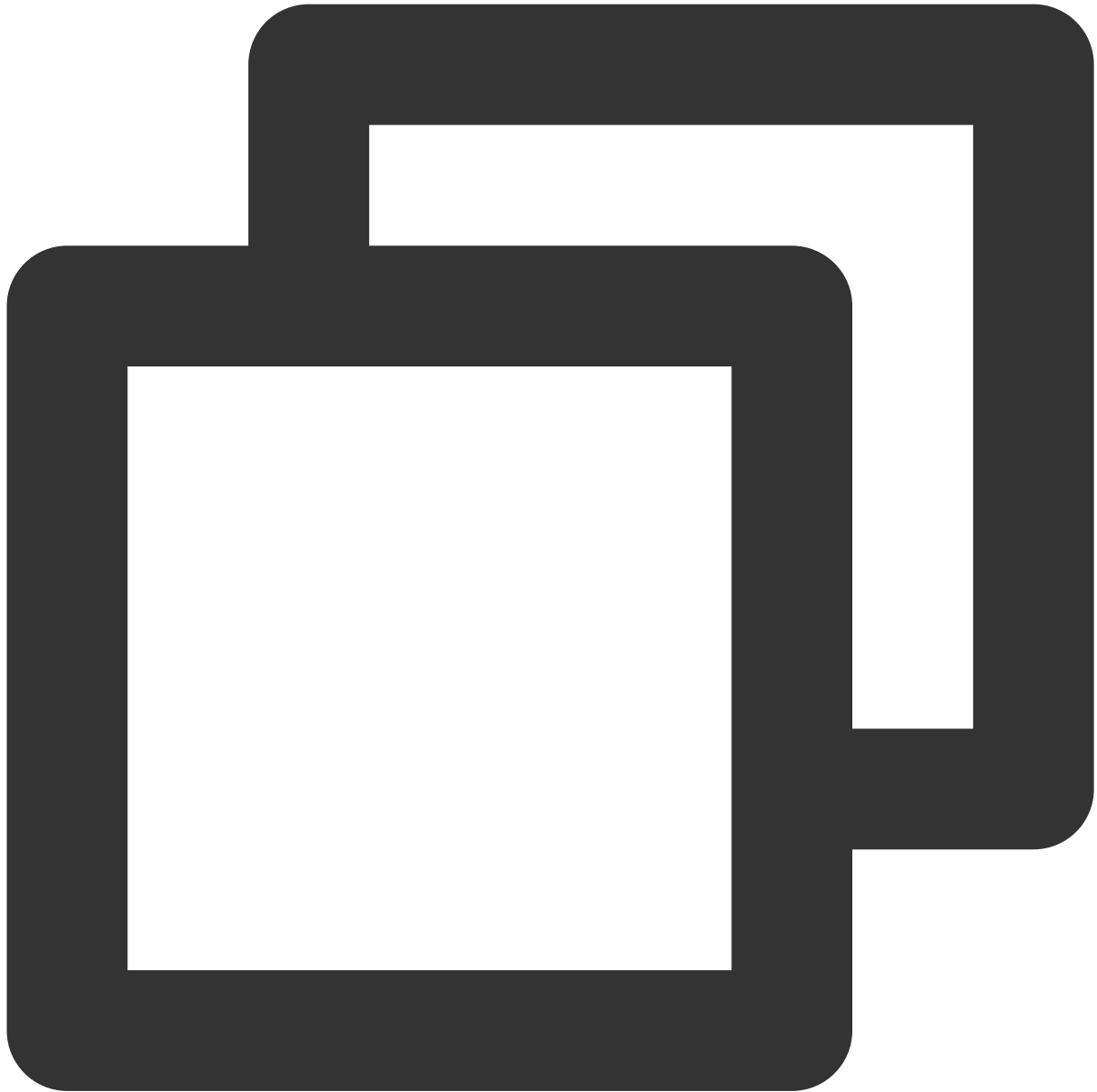
分页获取

可以使用 **Limit** 和 **Offset** 两个值用于控制分页拉取：

Limit 限制回包中 **MemberList** 数组中成员的个数，不得超过200，建议100。

Offset 控制从群成员中的第多少个成员开始拉取信息。对于分页请求（页码数字从1开始），每一页的 **Offset** 值应当为：**（页码数- 1）×每页展示的群成员数量**。

例如：假设需要分页拉取，每页展示 20 个，则第一页的请求参数应当为：`{"Limit" : 20, "Offset" : 0}`，第二页的请求参数应当为 `{"Limit" : 20, "Offset" : 20}`，依此类推。



```
{  
  "GroupId": "@TGS#1NVTZEAE4", // 群组 ID (必填)  
  "Limit": 100, // 最多获取多少个成员的资料  
  "Offset": 0 // 从第多少个成员开始获取资料  
}
```

分批获取

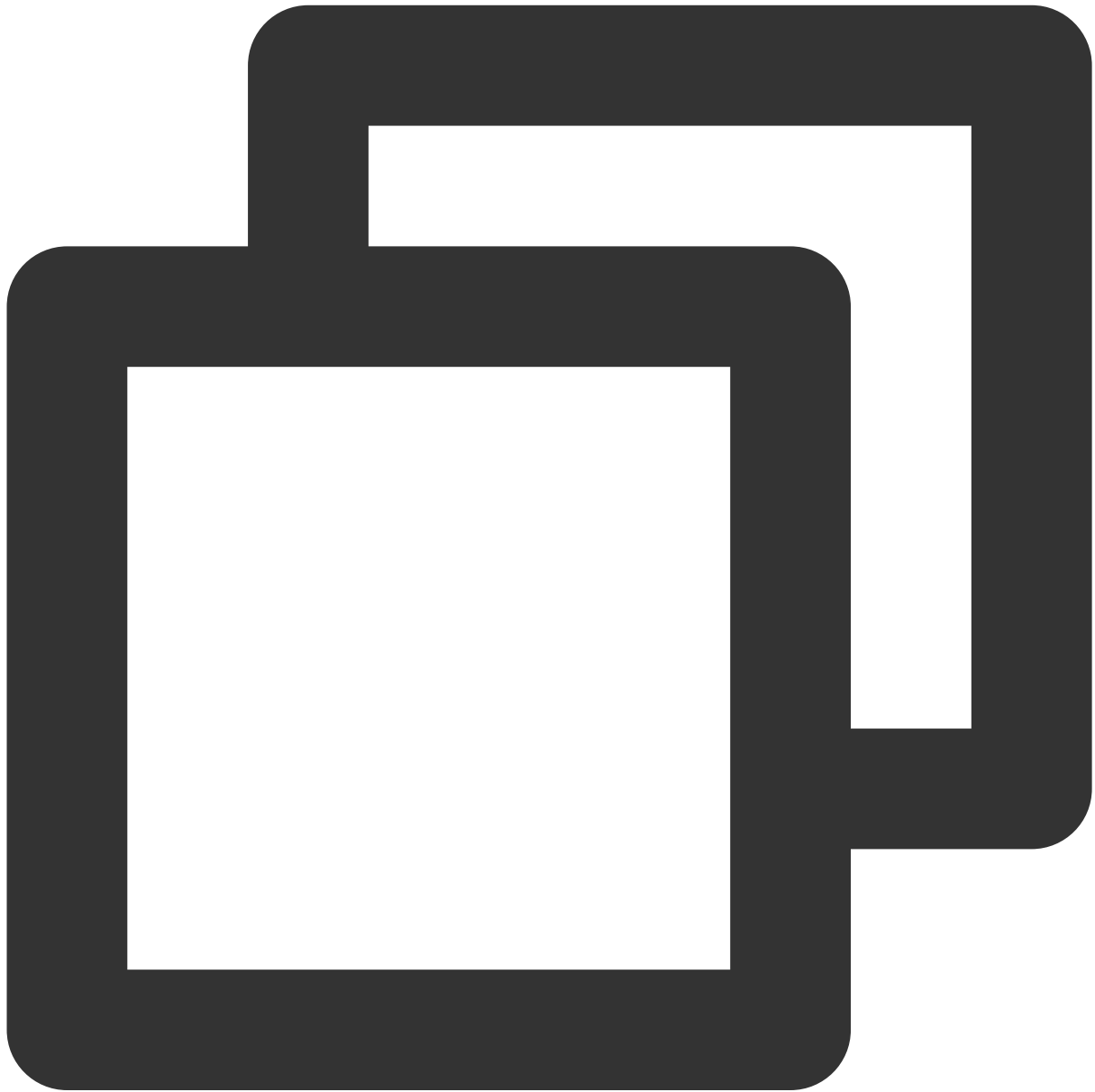
使用 Limit 和 Next 两个值用于控制分页拉取：

Limit 限制回包中 MemberList 数组中成员的个数，不得超过100。

Next 控制从某一个成员位置拉取后续的信息。第一次请求时，客户端请求参数 Next 传""；最后一次请求服务端返回的 Next 为""时，代表拉取结束；

中间的请求，客户端的 Next 均使用上一次服务端返回的 Next。类似 redis 的 scan 游标查询。

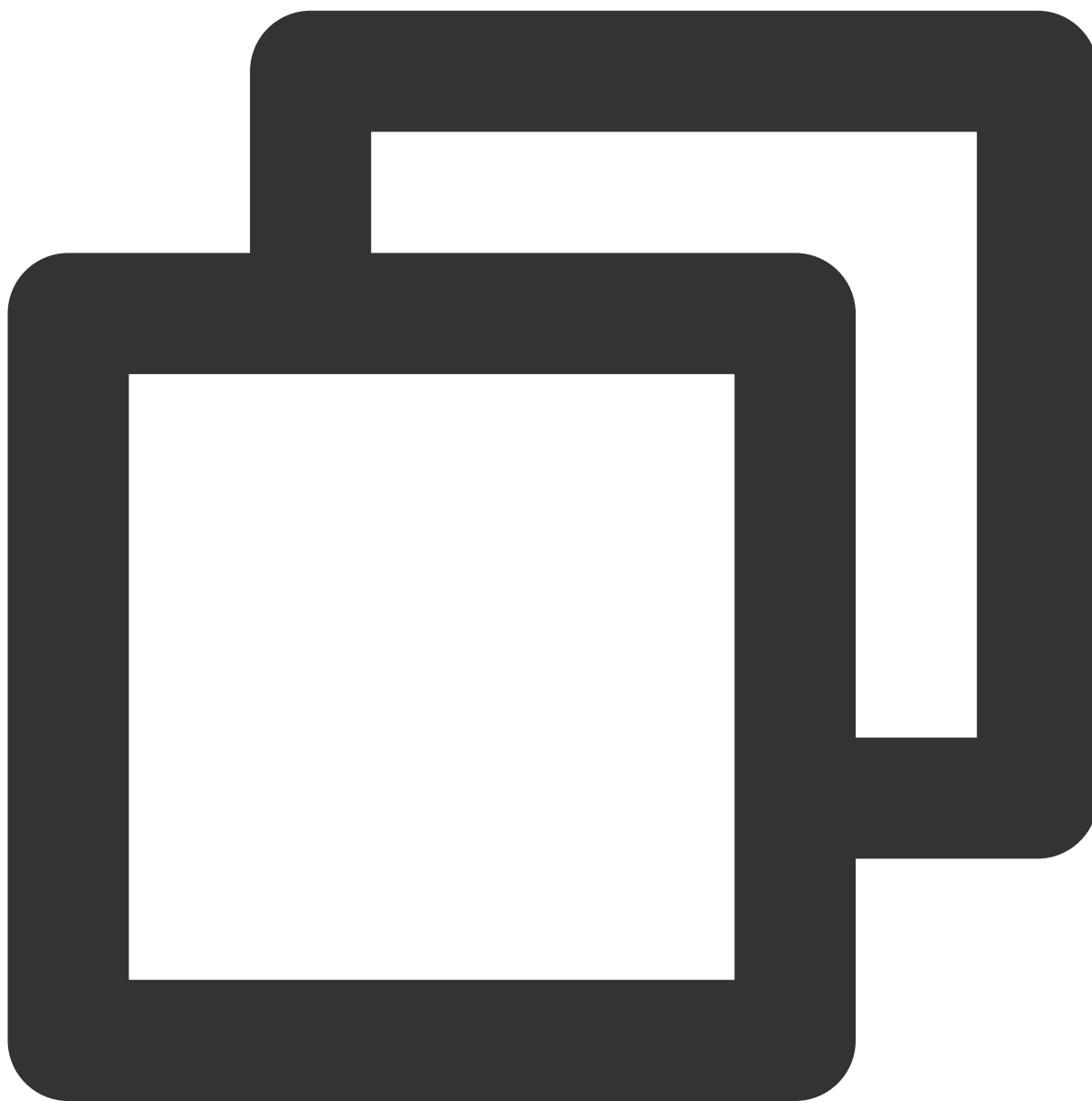
例如：假设需要分批拉取，第一次的请求参数应当为：`{"Limit" : 20, "Next" : ""}`，服务端返回



```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
```

```
"Next": "144115265295492787",  
"MemberList": [  
    ....
```

第二次请求参数应当为 `{"Limit" : 20, "Next" : "144115265295492787"}`。
以此类推，直到服务端的应答包返回Next为""，代表无后续成员信息，客户端应结束查询。

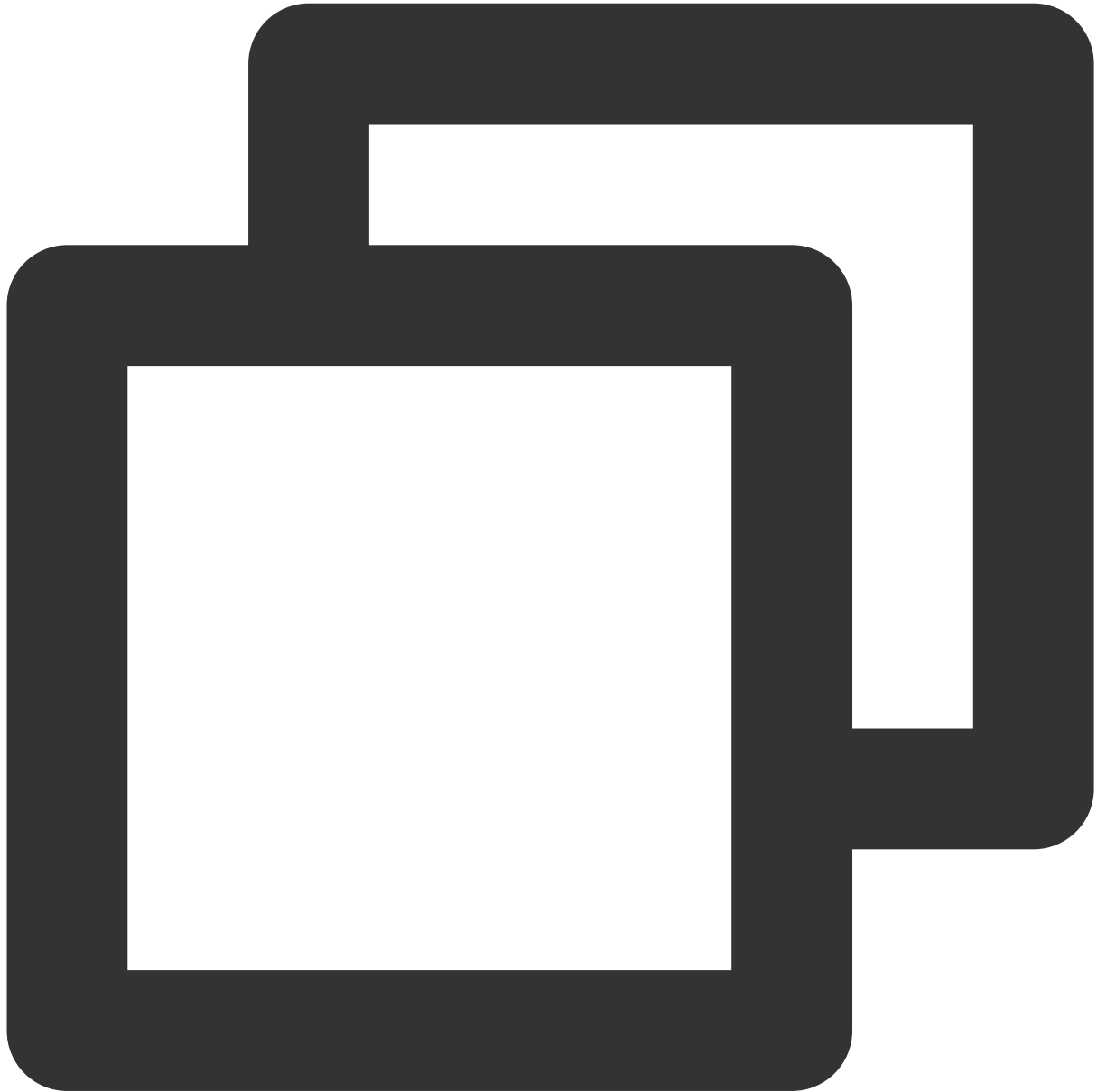


```
{  
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // 群组 ID (必填)  
  "Limit": 100, // 最多获取多少个成员的资料  
  "Next": "" // 从上次拉取结束的位置开始拉取
```

```
}
```

指定拉取的信息

通过 MemberInfoFilter 过滤器字段选择需要拉取的字段。没有在过滤器中指明的字段将不被拉取。

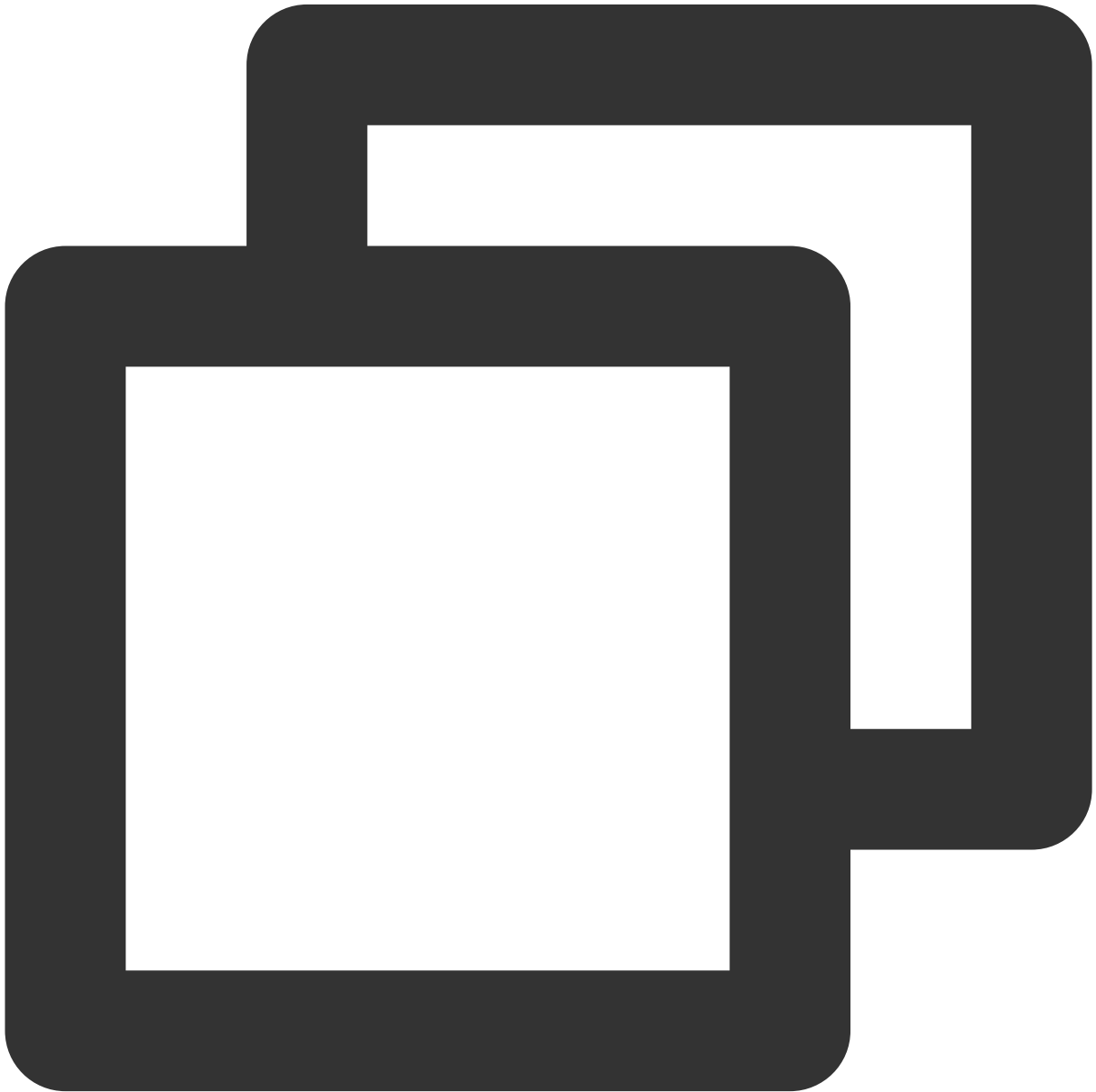


```
{  
  "GroupId": "@TGS#1NVTZEAE4", // 群组 ID (必填)  
  "MemberInfoFilter": [ // 需要获取哪些信息 (Member_Account 被默认包含在其中), 如果没有该字  
    "Role",  
    "JoinTime",  
    "MsgSeq",
```

```
"MsgFlag",  
"LastSendMsgTime",  
"MuteUntil",  
"NameCard"  
]  
}
```

拉取指定身份成员

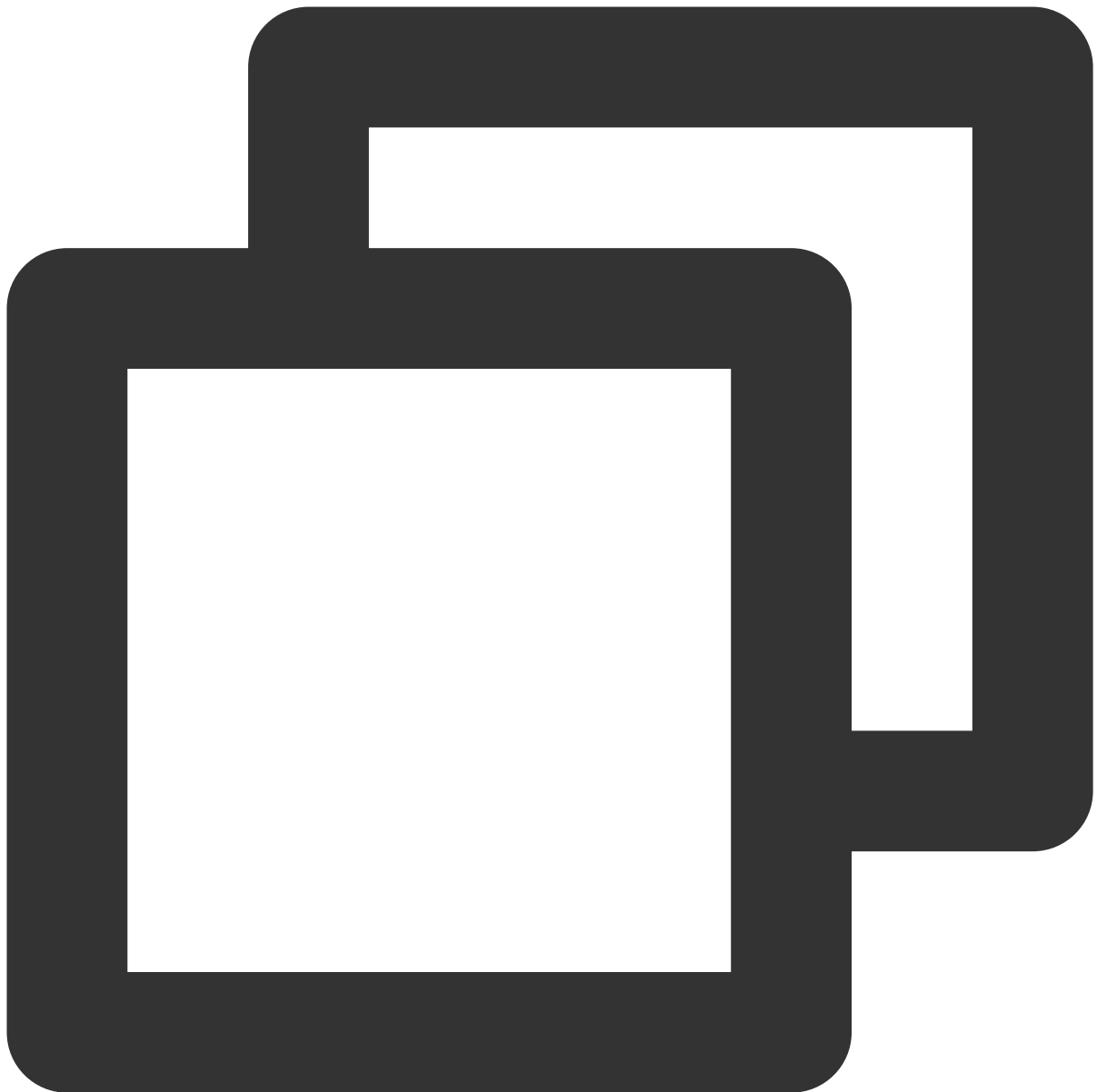
通过 MemberRoleFilter 过滤器字段选择需要拉取资料的成员身份。没有在过滤器中指明则代表拉取任何身份的成员的资料。



```
{
  "GroupId": "@TGS#37AB3PAEC", // 群组 ID (必填)
  "MemberRoleFilter": [ // 群成员身份过滤器
    "Owner",
    "Member"
  ]
}
```

拉取群成员自定义字段

通过 `AppDefinedDataFilter_GroupMember` 过滤器选取需要拉取的成员自定义字段。没有在过滤器中指定的字段将不被拉取。



```
{  
  "GroupId": "@TGS#37AB3PAEC", // 群组 ID (必填)  
  "AppDefinedDataFilter_GroupMember": [ // 群成员自定义字段过滤器  
    "MemberDefined2" // 群成员自定义字段key  
  ]  
}
```

ALL IN ONE



```
{
  "GroupId": "@TGS#1NVTZEAE4", // 群组 ID (必填)
  "MemberInfoFilter": [ // 需要获取哪些信息, 如果没有该字段则为群成员全部资料
    "Role",
    "JoinTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "MuteUntil",
    "NameCard"
  ],
}
```

```

"MemberRoleFilter": [ // 群成员身份过滤器
    "Owner",
    "Member"
],
"AppDefinedDataFilter_GroupMember": [ // 群成员自定义字段过滤器
    "MemberDefined2", // 群成员自定义字段 Key
    "MemberDefined1"
],
"Limit": 100, // 最多获取多少个成员的资料
"Offset": 0 // 从第多少个成员开始获取
}
    
```

请求包字段说明

字段	类型	属性	说明
GroupId	String	必填	需要拉取成员信息的群组的 ID
MemberInfoFilter	Array	选填	需要获取哪些信息，如果没有该字段则为群成员全部资料，成员信息字段详情请参阅 群成员资料
MemberRoleFilter	Array	选填	拉取指定身份的群成员资料。如没有填写该字段，默认为所有身份成员资料，成员身份可以为：“Owner”，“Admin”，“Member”
AppDefinedDataFilter_GroupMember	Array	选填	默认情况是没有的。该字段用来群成员维度的自定义字段过滤器，指定需要获取的群成员维度的自定义字段，群成员维度的自定义字段详情请参阅 自定义字段
Limit	Integer	选填	一次最多获取多少个成员的资料，不得超过6000。如果不填，则获取群内全部成员的信息
Offset	Integer	选填	从第几个成员开始获取，如果不填则默认为0，表示从第一个成员开始获取
Next	String	选填	上一次拉取到的成员位置，社群必填，社群不支持 Offset 参数，使用 Next 参数，首次调用填写“”，续拉使用返回中的 Next 值

应答包体示例

基本形式和分页拉取



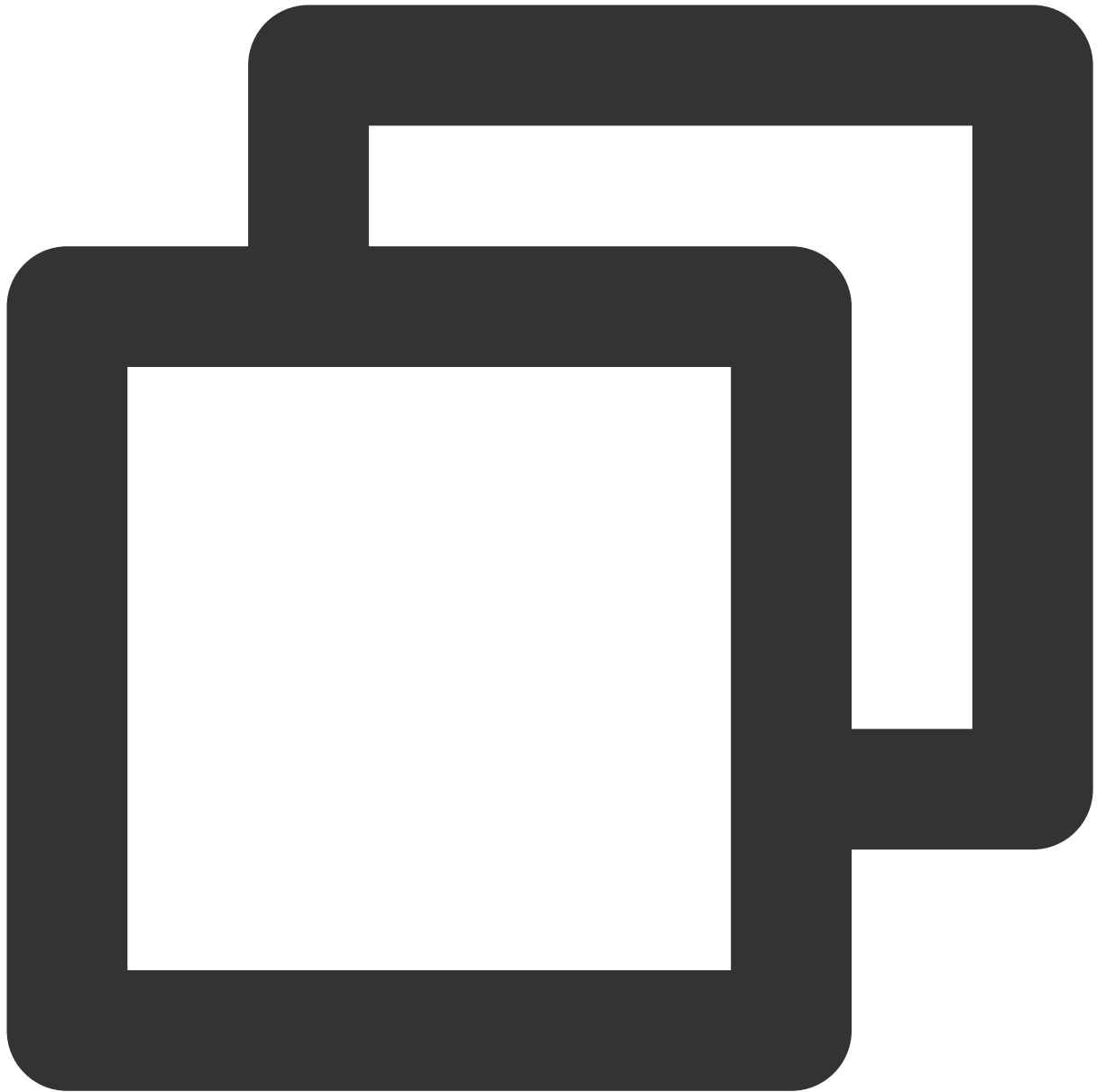
```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "144115265295492787", // 仅社群会返回该字段
  "MemberNum": 2, // 本群组的群成员总数
  "MemberList": [ // 群成员列表
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // 入群时间
    }
  ]
}
```

```

        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500, // 最后一次发消息的时间
        "MuteUntil": 1431069882, // 禁言截至时间 (秒数)
        "AppMemberDefinedData": [ //群成员自定义字段
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    },
    {
        "Member_Account": "peter",
        "Role": "Member ",
        "JoinTime": 1425976500,
        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500,
        "MuteUntil": 0, // 0表示未被禁言, 否则为禁言的截止时间
        "AppMemberDefinedData": [ // 群成员自定义字段
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    }
]
}
}

```

拉取指定字段



```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "MemberNum": 2, // 本群组的群成员总数
  "MemberList": [ // 群成员列表
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // 入群时间
      "MsgSeq": 1233,
    }
  ]
}
```

```
"MsgFlag": "AcceptAndNotify",
"LastSendMsgTime": 1425976500, // 最后一次发消息的时间
"MuteUntil": 1431069882, // 禁言截至时间 (秒数)
},
{
  "Member_Account": "peter",
  "Role": "Member ",
  "JoinTime": 1425976500,
  "MsgSeq": 1233,
  "MsgFlag": "AcceptAndNotify",
  "LastSendMsgTime": 1425976500,
  "MuteUntil": 0, // 0表示未被禁言, 否则为禁言的截止时间
}
]
}
```

拉取指定身份成员



```
{
  "ActionStatus": "OK", // 返回成功
  "ErrorCode": 0, // 返回码
  "MemberList": [
    {
      "JoinTime": 1450680436, // 成员加入时间
      "LastSendMsgTime": 0, // 成员最后发消息时间
      "Member_Account": "Test_1", // 成员帐号
      "MsgFlag": "AcceptNotNotify", // 成员消息屏蔽类型
      "MsgSeq": 1, // 成员已读消息 seq
      "NameCard": "", // 成员名片
    }
  ]
}
```

```
    "Role": "Owner", // 成员身份
    "MuteUntil": 0 // 0表示未被禁言, 否则为禁言的截止时间
  },
  {
    "JoinTime": 1450680436,
    "LastSendMsgTime": 0,
    "Member_Account": "Test_6",
    "MsgFlag": "AcceptNotNotify",
    "MsgSeq": 1,
    "NameCard": "",
    "Role": "Admin",
    "MuteUntil": 0
  }
],
"MemberNum": 8 // 本群组, 群成员总数
}
```

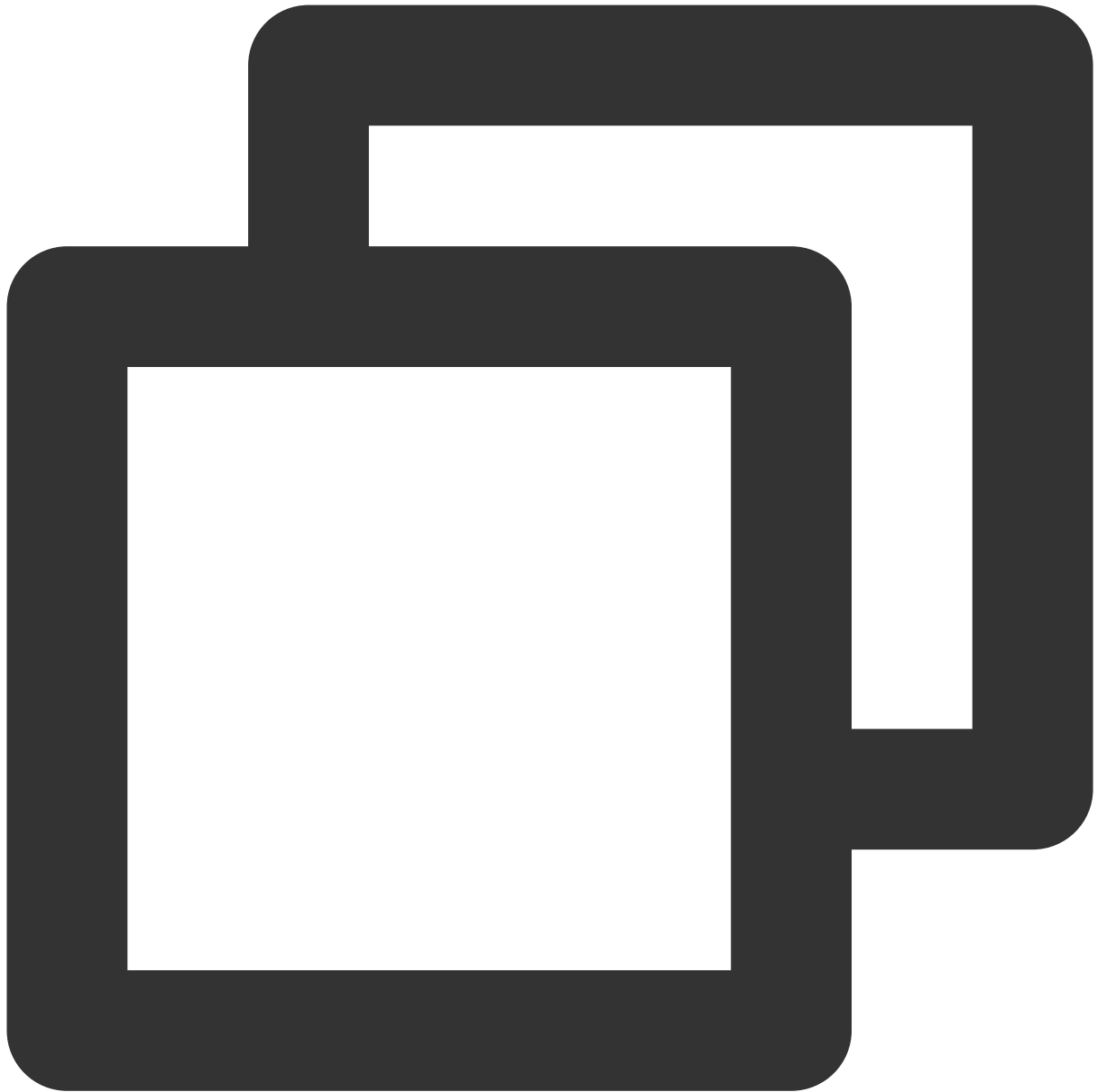
拉取群成员自定义字段



```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "MemberNum": 2, // 本群组的群成员总数
  "MemberList": [ // 群成员列表
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // 入群时间
      "MsgSeq": 1233,
    }
  ]
}
```

```
"MsgFlag": "AcceptAndNotify",
"LastSendMsgTime": 1425976500, // 最后一次发消息的时间
"MuteUntil": 1431069882, // 禁言截至时间 (秒数)
  "AppMemberDefinedData": [ // 群成员自定义字段
    {
      "Key": "MemberDefined2",
      "Value": "ModifyDefined2"
    }
  ]
},
{
  "Member_Account": "peter",
  "Role": "Member",
  "JoinTime": 1425976500,
  "MsgSeq": 1233,
  "MsgFlag": "AcceptAndNotify",
  "LastSendMsgTime": 1425976500,
  "MuteUntil": 0, // 0表示未被禁言, 否则为禁言的截止时间
  "AppMemberDefinedData": [ // 群成员自定义字段
    {
      "Key": "MemberDefined2",
      "Value": "ModifyDefined2"
    }
  ]
}
]
}
```

ALL IN ONE



```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "144115265295492787", // 仅社群会返回该字段
  "MemberNum": 2, // 本群组的群成员总数
  "MemberList": [ // 群成员列表
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // 入群时间
    }
  ]
}
```

```

        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500, // 最后一次发消息的时间
        "MuteUntil": 1431069882, // 禁言截至时间 (秒数)
        "AppMemberDefinedData": [ // 群成员自定义字段
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    },
    {
        "Member_Account": "peter",
        "Role": "Member",
        "JoinTime": 1425976500,
        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500,
        "MuteUntil": 0, // 0表示未被禁言, 否则为禁言的截止时间
        "AppMemberDefinedData": [ // 群成员自定义字段
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    }
]
}
}

```

应答包字段说明

字段

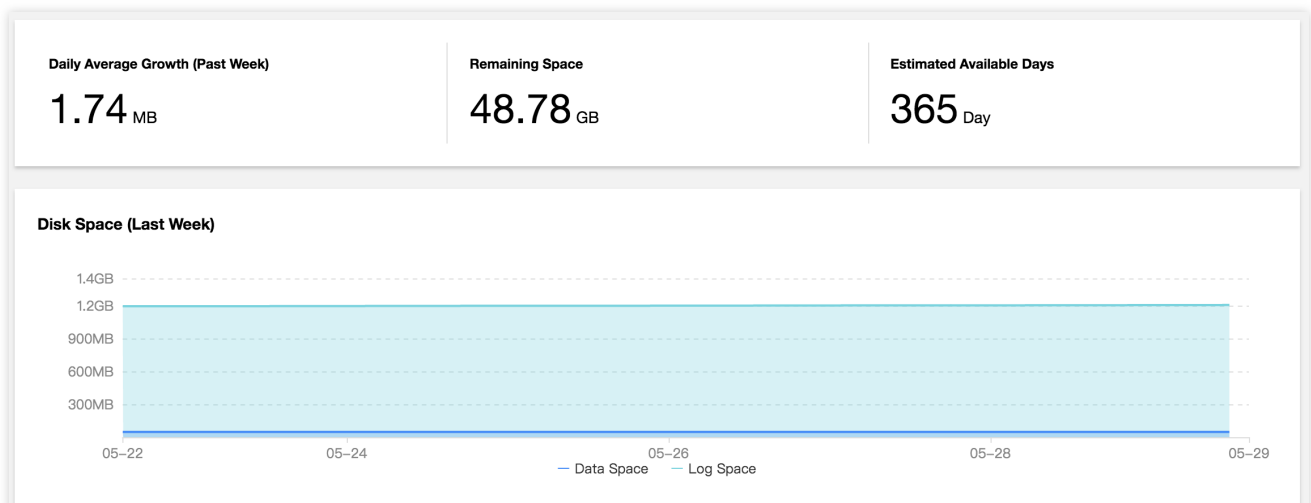
ActionStatus

ErrorCode

ErrorInfo

登录 [DBbrain 控制台](#)，在左侧导航选择**诊断优化**，在上方选择对应数据库，然后选择**空间分析**页。

在空间分析页可查看近一周的日均增长量对比、剩余磁盘空间、预计可用天数，以及近一周的磁盘空间趋势表。同也可以查看实例中数据库下各表的占用空间详情和碎片情况。



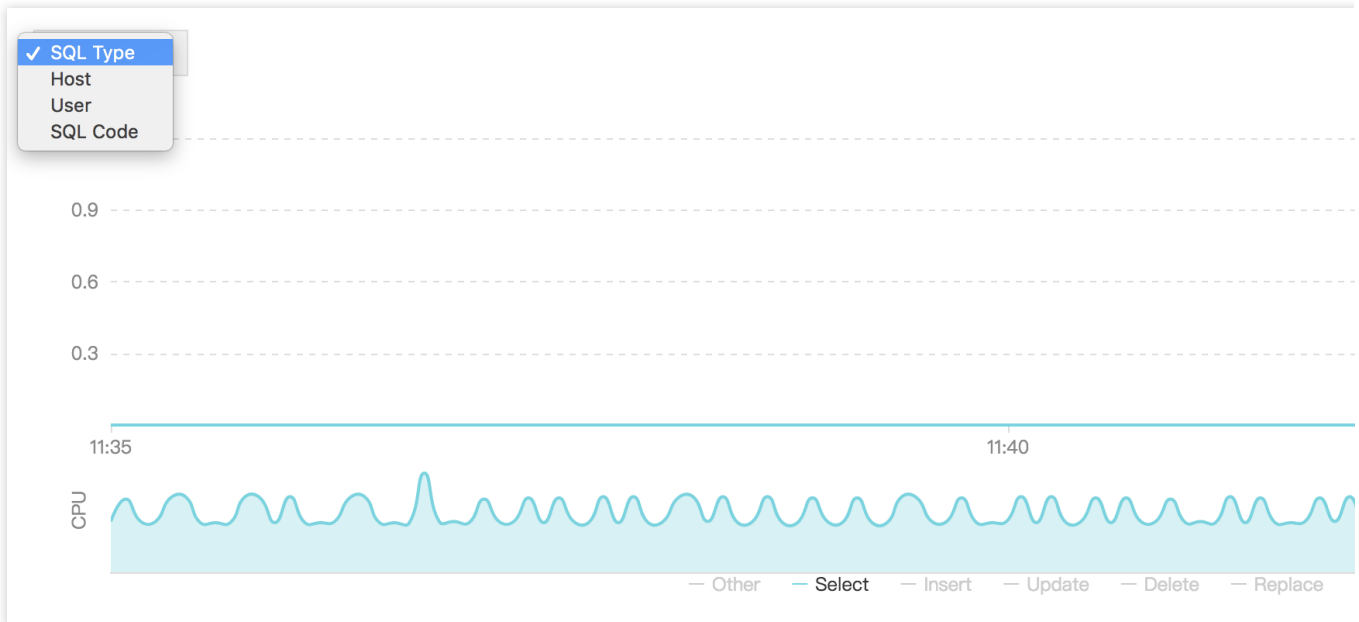
如何分析 MySQL 全量 SQL 执行轨迹？

登录 [DBbrain 控制台](#)，在左侧导航选择**诊断优化**，在上方选择对应数据库，然后选择**审计日志分析**页。

1. 单击视图右上角的**创建分析任务**，选择时间段，单击**确定**。
2. 在任务列表单击**查看 SQL 分析**，进入 SQL 分析页。

No.	Type	Health Level ▾	Creation Time ↕	Time Range
408	Scheduled T...	Healthy	2020-05-27 00:01:16	2020-05-25 23:59:00 ~ 2020-05-26 23:59:00

3. 在 SQL 分析页，可选择 SQL Type、Host、User 等维度的视图，并可选择时间段拉伸视图来查看具体时间点的数



4. 单击某行 SQL 模板，在右侧会弹出 SQL 语句的详情。

在分析页，可查看和复制具体 SQL 语句，根据给出的优化建议或说明来优化 SQL 语句。

在统计页，可查看该类 SQL 在 Host、User、SQL Code 维度的统计分析和执行时间轨迹。

MySQL 实例故障或异常时，如何自助诊断优化？

1. 登录 [DBbrain 控制台](#)，在左侧导航选择**诊断优化**，在上方选择对应数据库，然后选择**异常诊断**页。
2. “诊断提示”栏展示诊断事件历史记录概要信息，包括等级、开始时间、诊断项、持续时长。DBbrain 会定期对实例进行健康巡检。



3. 单击**查看详情**或“诊断提示”栏的诊断项可进入诊断详情页，在视图单击诊断事件，在下方会显示该事件的详情，包括事件概要、现象描述、智能分析以及专家建议，根据专家建议进行优化，即可解决数据库异常，提升实例性能。

Event Details

Item	CPU Utilization	Time Range	2020-06-02 16:42:01
Risk level	Alarm	Duration	7 minutes
Overview	monitoring metrics "cpu_use_rate" alarm, the current value 47.53		

Description Intelligent Analysis Optimization Advise

MSG_SQL_OPT

Database

SQL Statement

```
select id, pay_date, pay_hour, item_id, item_name,
       buyer_group_name, store_code, total_order_num,
       return_sale, item_buy_num, user_count, item_id
from t_order_item_sales_hour
where pay_date = '2020-05-24'
```

```
and item_id = 357221
```

Table `t_order_item_sales_hour`

Advice one Create Index

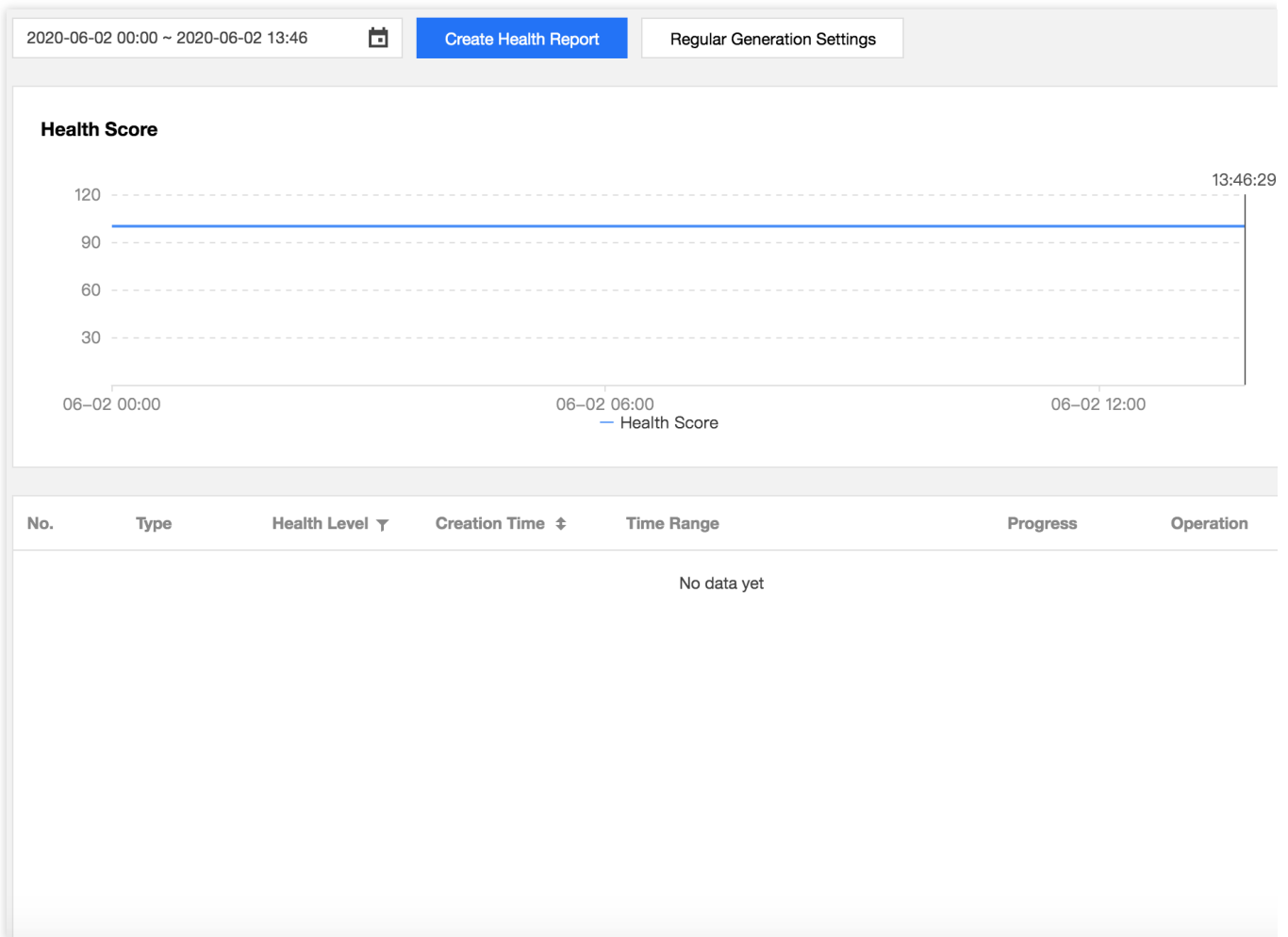
```
alter table `y` . `t_order_item_sales_hour` add index index_0(`pay_date`)
```

如何定期获取 MySQL 健康报告？

登录 [DBbrain 控制台](#)，在左侧导航选择**诊断优化**，在上方选择对应数据库，然后选择**健康报告**页，可查看选择时间段的健康得分趋势以及问题概要。

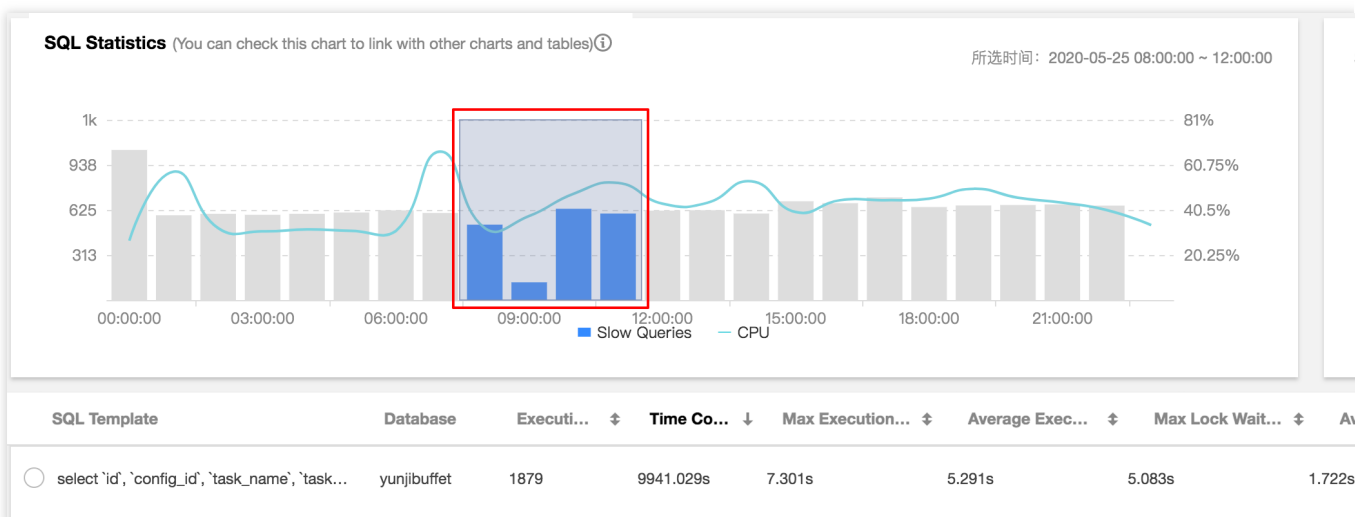
设置报告时间范围，然后单击**创建健康报告**，任务完成后可以查看或下载该时段的健康报告。

单击**定期生成设置**，可配置自动生成健康报告的时间周期。

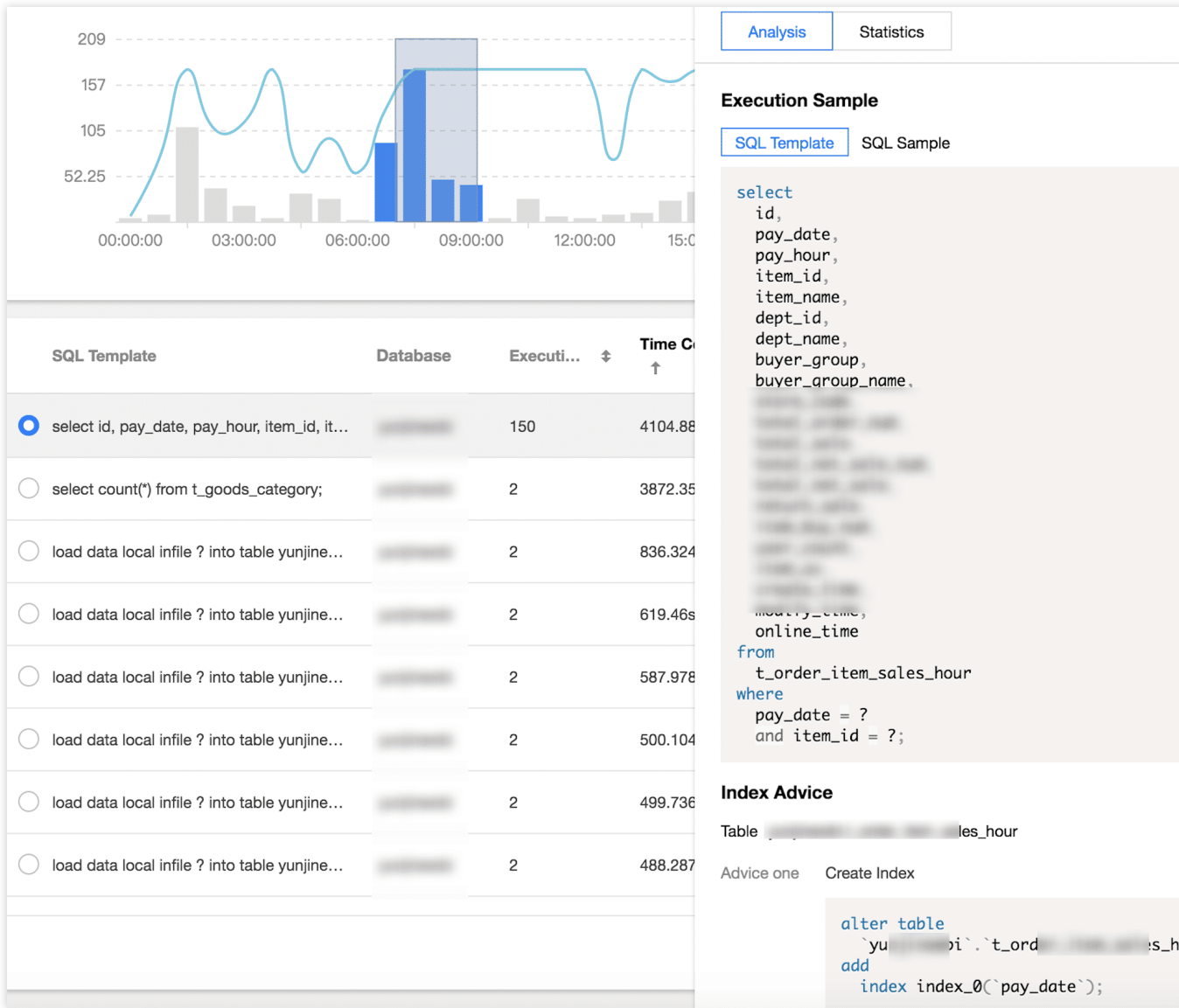


如何查看和优化 MySQL 慢日志？

1. 登录 [DBbrain 控制台](#)，在左侧导航选择**诊断优化**，在上方选择对应数据库，然后选择**慢 SQL 分析**页，“SQL 统计”栏展示实例的慢查询数和 CPU 使用率。
2. 单击或拉选 SQL 统计图表的慢查询，下方会显示聚合 SQL 模板以及执行信息，各列数据均支持正序或逆序排序。右侧的耗时分布中会展示所选时间段内的 SQL 总体耗时分布情况。



3. 单击某条聚合的 SQL 模板行，右侧边会弹出 SQL 的优化建议和统计信息，可根据优化建议改写 SQL 或者增加适当索引，即可提升 SQL 执行效率，提高数据库性能。



| MemberNum | Integer | 本群组的群成员总数 |

| MemberList | Array | 获取到的群成员列表，其中包含了全部或者指定的群成员信息，成员信息字段详情请参阅 [群成员资料](#) |

| AppMemberDefinedData | Array | 返回的群成员自定义字段信息 |

| Next | String | 下一次请求应该传的 Next 值，仅查询 Community（社群）时会返回该字段 |

错误码说明

除非发生网络错误（例如502错误），否则该接口的 HTTP 返回码均为200。真正的错误码，错误信息是通过应答包体中的 ErrorCode、ErrorInfo 来表示的。

公共错误码（60000到79999）参见 [错误码](#) 文档。

本 API 私有错误码如下：

错误码	描述
10002	服务器内部错误，请重试
10003	请求命令字非法
10004	参数非法，请根据错误描述检查请求是否正确
10007	操作权限不足，请确认操作者是否是 App 管理员或者是否有权限读取请求中的字段
10010	群组不存在，或者曾经存在过，但是目前已经被解散
10015	群组 ID 非法，请检查群组 ID 是否填写正确
10018	应答包长度超过最大包长（1MB），群成员数据过多，请尝试使用 Limit 和 Offset 分页拉取群成员数据

接口调试工具

通过 [REST API 在线调试工具](#) 调试本接口。

参考

修改群成员资料（[v4/group_open_http_svc/modify_group_member_info](#)）。

产品常见问题

最近更新时间：2022-09-01 18:34:46

DBbrain 的兼容性如何？

支持多款 SQL、NoSQL、NewSQL 数据库类型，目前支持云数据库 MySQL（不含单节点 - 基础型）、云原生数据库 TDSQL-C（TDSQL-C for MySQL）、用户自建数据库 MySQL、云数据库 Redis、云数据库 MongoDB。

DBbrain 具备哪些功能？

目前提供实例监控、审计日志分析（包含 SQL 分析和安全风险识别）、慢 SQL 分析、实时会话、健康报告等功能。

DBbrain 的入口有哪些？

DBbrain 提供 PC 端和移动端两种访问方式。PC 端的产品入口在 [DBbrain 控制台](#)，移动端可通过“腾讯云助手”微信小程序或“腾讯云数据库”官方微信使用。