阿里云
专有云Enterprise版
业务实时监控服务ARMS

开发指南

产品版本：V3.3.0

文档版本：20180312

为了无法计算的价值 | 阿里云

法律声明

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

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

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	**禁止**： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	**警告**： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	**说明**： 您也可以通过按**Ctrl** + **A**选中全部文件。
>	多级菜单递进。	**设置** > **网络** > **设置网络类型**
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。
courier字体	命令。	执行 cd /d C:/windows 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	ipconfig *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	swich *{stand | slave}*

目录

法律声明.....I
通用约定.....I
1 ARMS Open API概述.....1
2 请求参数.....2
3 Postman请求示例.....3
4 返回码.....4
4.1 errMsg数据集不存在.....4
4.2 errMsg查询结果为空.....4
4.3 Code 500.....5
4.4 Code 200.....5
5 接口说明.....6
5.1 数据集查询接口.....6
5.2 下钻数据集查询接口.....7
5.3 下钻数据集多维度查询接口.....7
5.4 Where In条件数据集查询接口.....8
5.5 通用数据集查询接口.....8

1 ARMS Open API概述

本手册介绍如何用ARMS RESTful API进行数据集的查询。

主要包含以下内容：

- 参数说明：ARMS API规格，包含API路径、方法、输入参数以及输出参数的详细说明。
- 返回码（Response Codes）：API调用错误情况下返回的Response介绍。
- 详细的接口信息及使用示例：包括通用数据集查询、数据集下钻查询、数据集多维度查询、whereIn条件数据集查询接口。

2 请求参数

本节介绍ARMS API请求字段具体含义。

说明：

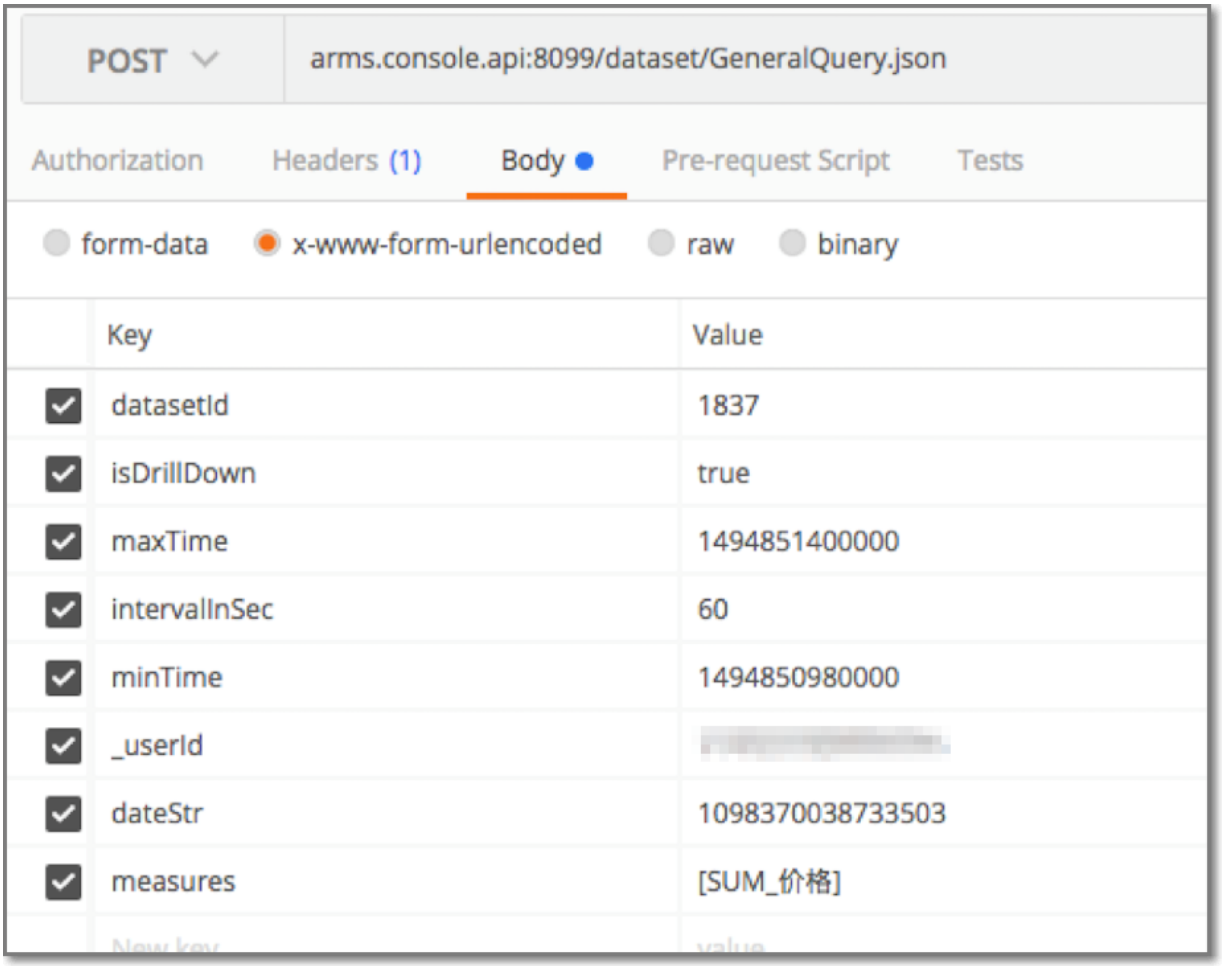
List[String] 表示List的toString()之后的字符串。

字段名称	字段类型	字段含义	是否必须	备注
datasetId	Long	数据集ID	是	
minTime	Long	查询数据的起始时间	是	
maxTime	Long	查询数据的截止时间	是	
intervalInSec	Integer	数据片的时间间隔	是	
measures	List[String]	查询指标列表	否	列表最长支持3个元素。空则返回所有指标数据
isDrillDown	Boolean	是否下钻	否	
dimensions	List[String]	查询维度	否	
whereInKey	String	Where In条件查询Key字段	否	
whereInValues	List[String]	Where In条件查询Key字段	否	
requiredDims	List[String]	通用数据集的必选维度	否	
optionalDims	List[String]	通用数据集的可选维度	否	
orderByKey	String	orderBy指标	否	
limit	Integer	限制的返回个数	否	
reduceTail	Boolean	是否把limit之外的数据合并到一起	否	
_userId	String	用户id		私有云中，_userId与用户名相同，例如ARMS管理员用户ID为arms_admin

3 Postman请求示例

在使用ARMS Open API前，您可以通过第三方软件Postman进行测试，确保数据集数据可以获取。

图 3-1: Postman请求示例



4 返回码

本节介绍API响应内容以及相关解决办法。

API返回通用响应结果包含如下内容。

名称	说明
code	错误代码
success	是否成功（true | false）
message	错误消息
data	ARMS查询结果具体信息

4.1 errMsg数据集不存在

图 4-1: 数据集不存在

```
{
  "code": 200,
  "data": "{\"actualSizeFromDB\":0,\"errMsg\":\"数据集ID不存在！\",\
  "message": "success",
  "success": true
}
```

解决办法：请在ARMS中查询该数据集ID是否拼写正确，该数据集ID是否属于该用户。

4.2 errMsg查询结果为空

图 4-2: 查询结果为空

```
{
  "code": 200,
  "data": "{\"actualSizeFromDB\":0,\"errMsg\":\"查询结果为空\",\
  "message": "success",
  "success": true
}
```

解决办法：

1. 确定在请求的开始时间和结束时间之间是否存在数据。
2. 如果选择了维度，检查该维度是否存在数据。

3. userId设置错误，私有云中_userId与用户名相同。

4.3 Code 500

ARMS内部错误。

解决办法：请联系ARMS开发人员。

4.4 Code 200

访问处理成功。

5 接口说明

本节提供ARMS 各个接口的说明及示例。

以下示例使用Java Apache HTTP Client以及Python脚本方式发送HTTP Post请求获取数据。您可以根据自己需求选择其他语言，例如Ruby等。

5.1 数据集查询接口

调用此接口来查询通用数据集数据。本文提供Java和Python示例。

Java Apache HTTP请求示例

```
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.http.message.BasicNameValuePair;
import org.junit.Test;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

public class GeneralQueryTest {

    public static final String url = "http://***.**.**.**:8099/dataset/GeneralQuery.json";

    @Test
    public void GeneralQueryTest() throws Exception {
        HttpClient client = HttpClientBuilder.create().build();
        HttpPost post = new HttpPost(url);

        // add header
        post.setHeader("Content-Type", "application/x-www-form-urlencoded");

        // add post parameters
        List<NameValuePair> urlParameters = new ArrayList<>();
        urlParameters.add(new BasicNameValuePair("datasetId", "1837"));
        urlParameters.add(new BasicNameValuePair("maxTime", "1494851400000"));
        urlParameters.add(new BasicNameValuePair("intervalInSec", "60"));
        urlParameters.add(new BasicNameValuePair("minTime", "1494850980000"));
        urlParameters.add(new BasicNameValuePair("_userId", "11************596"));
        List<String> measures = new ArrayList<>();
        measures.add("COUNT__line");
        measures.add("SUM_price");
        urlParameters.add(new BasicNameValuePair("measures", measures.toString()));
        post.setEntity(new UrlEncodedFormEntity(urlParameters));

        // send post
        HttpResponse response = client.execute(post);

        // analyse result
```

```
        BufferedReader rd = new BufferedReader(
                new InputStreamReader(response.getEntity().getContent()));

        StringBuffer result = new StringBuffer();
        String line = "";
        while ((line = rd.readLine()) != null) {
            result.append(line);
        }
    }
}
```

Python请求示例

```
#!/usr/bin/python
#-*-coding:utf-8 -*-

import httplib, urllib

try:
        params = urllib.urlencode({'datasetId': '1837',
                                   'isDrillDown': 'false',
                                   'maxTime': '1494851400000',
                                   'intervalInSec': '60',
                                   'minTime': '1494850980000',
                                   '_userId': '11***********96',
                                   'measures': '[COUNT__line, SUM_price]'
                                  })
        headers = {"Content-type": "application/x-www-form-urlencoded",
                   "Accept": "text/plain"}

        conn = httplib.HTTPConnection("***:**:**:**",8099)
        conn.request("POST", "/dataset/GeneralQuery.json", params, headers)
        response = conn.getresponse()
        print response.read()
except Exception,e:
        print str(e)
finally:
        if conn:
                conn.close()
```

5.2 下钻数据集查询接口

请在数据集查询接口示例代码基础上修改一行代码。

```
urlParameters.add(new BasicNameValuePair("isDrillDown", "true"));
```

5.3 下钻数据集多维度查询接口

请在数据集查询接口示例代码基础上增加以下代码。

```
Map<String, String> dimension = new LinkedMap();
dimension.put("value", "hz");
dimension.put("key", "区域");
dimension.put("type", "ALL");
List<Map<String, String>> dimensions = Lists.newArrayList();
dimensions.add(dimension);
```

5.4 Where In条件数据集查询接口

请在数据集查询接口示例代码的基础上，增加两行代码。

```
urlParameters.add(new BasicNameValuePair("whereInKey","区域");
List<String> whereInValues = new ArrayList<>();
whereInValues.add("hz");
whereInValues.add("bj");
urlParameters.add(new BasicNameValuePair("whereInValues", whereInValues.toString()));
```

5.5 通用数据集查询接口

请在数据集查询接口示例代码的基础上，增加以下代码。

```
Map<String, String> optionalDim = Maps.newTreeMap();
optionalDim.put("value", "hz");
optionalDim.put("key", "区域");
optionalDim.put("type", "ALL");
List<Map<String, String>> optionalDims = Lists.newArrayList();
optionalDims.add(optionalDim);
```