



DAQTest 测试管理平台 RESTful API 接口说明书

北京诺约科技有限公司

PROMISE AUTO

前言

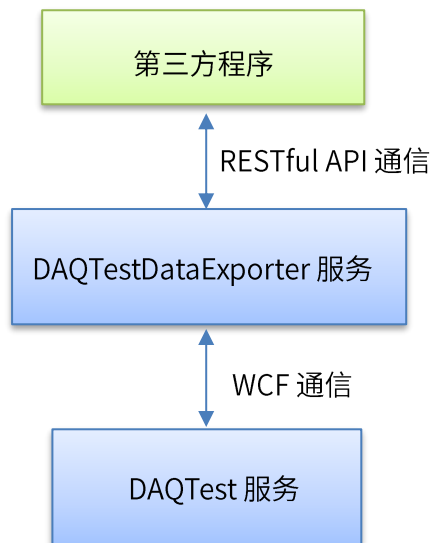
本说明书将简明地介绍测试管理平台 DAQTest 提供的对外接口。方便第三程序访问 DAQTest 内部的数据。

版本

第一版 2020年6月

第二版 2023年7月

系统构成



DAQTestDataExporter 服务

DAQTest 安装时会安装 DAQTestDataExporter 服务，该服务提供 RESTful API 服务，但缺省状态该服务为手动，停止状态。如果要使用 RESTful API,请在服务管理器里面，把 DAQTestDataExporter 服务改为自动，开始状态。

例子程序

DAQTestRESTfulAPIExample 为 RESTful API 调用例子程序，模拟第三方程序实时获取 DAQTest 的数据，可供参考。

程序功能：演示简单的获取测试数据的方法

开发环境：VS2015 或更新的版本，.NET 4.0 或更新的框架

开发语言：C#语言

API 接口的语言

RESTful API, 复杂数据以 json 字符串形式传递。

API 接口的功能

1. 登陆/登出服务器
2. 维持心跳
3. 获取运行试验列表
4. 打开测试 (获取运行试验通道列表, 各通道的信息), 关闭测试
5. 获取运行试验的数据范围, 各通道数据实时值, 缓存值

API 接口使用说明

1. 基本使用流程: 登录 Login, 获取运行中测试列表 GetRunningTest, 打开测试 OpenTest, 获取数据 GetTestNewestData, 关闭测试 CloseTest, 登出 Logout。
2. 除 Login 接口外所有接口需要带 sessionID (由 Login 调用成功时获取)
3. 原则上成功 Login, 执行各种操作最后需要显示调用 Logout 释放服务器资源, 即使遇到错误最后也需要显示调用 Logout
4. 所有接口返回 json 字符串, 必含信息: ErrorCode (错误码), ErrorMessage (错误描述), 可选信息: 根据具体接口而定
5. 心跳时间为 1 分钟, 1 分钟无任何操作服务器会关闭连接。如果调用时间间隔超过 1 分钟, 可通过调用 KeepAlive 接口维持心跳

API 接口错误代码

请参考 DAQTest 软件说明书中的附录章节。

API 函数说明

登录系统

下面的说明是基于第三方程序跟 DAQTestDataExporter 服务部署在同一台电脑说明。
如果部署在不同的电脑，请把 localhost 替换成 DAQTestDataExporter 服务部署的电脑 IP 地址。用户名和密码为 DAQTest 服务器中注册的用户名和密码。

<http://localhost:6003/DAQTest/Login?userName=admin&password=>

参数：

userName: 登录系统用户名

password: 登录系统密码

返回值：

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

sessionID: 生成的会话 ID, 其他函数调用时需要该参数

返回例：

```
{
  "ErrorCode": 0,
  "ErrorMessage": "(E0)",
  "sessionID": "468f4f02-c533-4989-9e47-002acb561e1b"
}
```

登出系统

<http://localhost:6003/DAQTest/Logout?sessionID=7bd8c6f1-5da1-4f31-933f-025277e00fa8>

参数:

sessionID: 会话 ID (Login 时返回)

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

返回例:

```
{  
  "ErrorCode": 0,  
  "ErrorMessage": "(E0)"  
}
```

维持心跳

<http://localhost:6003/DAQTest/KeepAlive?sessionID=7bd8c6f1-5da1-4f31-933f-025277e00fa8>

参数:

sessionID: 会话 ID

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

返回例:

```
{  
  "ErrorCode": 0,  
  "ErrorMessage": "(E0)"  
}
```

获取运行中测试列表

<http://localhost:6003/DAQTest/GetRunningTest?sessionID=7bd8c6f1-5da1-4f31-933f-025277e00fa8>

参数:

sessionID: 会话 ID

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

Tests: 获取的测试条目数组, 每个条目包括下面内容

testID: 测试 ID, 用于调用操作测试的函数

testName: 测试名

createTime: 测试创建时间字符串

返回例:

```
{
  "ErrorCode": 0,
  "ErrorMessage": "(E0)",
  "Tests": [
    {
      "testID": "4842656941175208427",
      "testName": "温升试验_003",
      "createTime": "2023-08-27 09:24:31"
    },
    {
      "testID": "5269691028443731747",
      "testName": "DataExport",
      "createTime": "2023-08-27 10:32:21"
    }
  ]
}
```


打开测试

<http://localhost:6003/DAQTest/OpenTest?sessionID=13899e0e-db6d-4b99-af97-f27386b54882&testID=4935486268784259400>

参数：

sessionID: 会话 ID

testID: 测试 ID

返回值：

错误信息：ErrorCode 错误码，ErrorMessage 错误详细信息

Test: 测试信息，包括下面内容

Name: 测试名字

Interval: 测试采集周期

ChannelSettingList: 测试通道信息列表，每条记录包括下面内容

No: 通道编号

Name: 通道名字

Unit: 通道单位

Max: 通道上限

Min: 通道下限

DeciPos: 通道小数点

注意：

OpenTest 成功后需要调用 CloseTest 关闭测试

返回例:

```
{
  "ErrorCode": 0,
  "ErrorMessage": "(E0)",
  "Test": {
    "Name": "DataExport",
    "Interval": 1,
    "ChannelSettingList": [
      {
        "No": 1,
        "Name": "Simulator-CH1",
        "Unit": "",
        "Max": 100,
        "Min": 0,
        "DeciPos": 3
      },
      {
        "No": 2,
        "Name": "Simulator-CH2",
        "Unit": "",
        "Max": 100,
        "Min": 0,
        "DeciPos": 3
      }
    ]
  }
}
```

关闭测试

<http://localhost:6003/DAQTest/CloseTest?sessionID=7bd8c6f1-5da1-4f31-933f-025277e00fa8&testID=5512840618815756261>

参数:

sessionID: 会话 ID

testID: 测试 ID

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

返回例:

```
{  
  "ErrorCode": 0,  
  "ErrorMessage": "(E0)"  
}
```

获取测试状态

<http://localhost:6003/DAQTest/GetTestStatus?sessionID=13899e0e-db6d-4b99-af97-f27386b54882&testID=4935486268784259400>

参数:

sessionID: 会话 ID

testID: 测试 ID

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

status: 测试状态 (Created: 创建未执行, Running: 运行中, Finished: 测试结

束)

返回例:

```
{  
  "ErrorCode": 0,  
  "ErrorMessage": "(E0)",  
  "status": "Running"  
}
```

获取测试最新数据

<http://localhost:6003/DAQTest/GetTestNewestData?sessionID=13899e0e-db6d-4b99-af97-f27386b54882&testID=4935486268784259400>

参数:

sessionID: 会话 ID

testID: 测试 ID

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

TestData: 最新数据, 包括下面内容

time: 数据时间

no: 数据编号

data: 各通道数据, 通道数据包括下面内容

channelNo: 通道编号

value: 通道值

status: 通道数据状态, 0 为正常, 其他为异常

alarm: 通道报警状态, bit0: 第一级报警, bit1: 第二级报警

返回例:

```
{
  "ErrorCode": 0,
  "ErrorMessage": "(E0)",
  "TestData": {
    "time": "2023-08-27T11:54:19",
    "no": 4878,
    "data": [
      {
        "channelNo": 1,
```

```
    "value": 73.70116424560547,  
    "status": 0,  
    "alarm": 0  
  },  
  {  
    "channelNo": 2,  
    "value": 99.8520736694336,  
    "status": 0,  
    "alarm": 0  
  }  
]  
}  
}
```

获取测试缓存数据范围

<http://localhost:6003/DAQTest/GetTestDataRange?sessionID=b796f565-507d-4f45-8b54-213c19299659&testID=5068677805055829947>

参数:

sessionID: 会话 ID

testID: 测试 ID

返回值:

错误信息: ErrorCode 错误码, ErrorMessage 错误详细信息

start: 数据开始编号

end: 数据结束编号

返回例:

```
{  
  "ErrorCode": 0,  
  "ErrorMessage": "(E0)",  
  "start": 4100,  
  "end": 5040  
}
```

获取测试缓存数据

<http://localhost:6003/DAQTest/GetTestData?sessionID=b796f565-507d-4f45-8b54-213c19299659&testID=5068677805055829947&startNo=2618&endNo=2619>

参数：

sessionID：会话 ID

testID：测试 ID

startNo：获取的数据的开始编号

endNo：获取的数据的结束编号，-1 表示获取到最新数据

返回值：

错误信息：ErrorCode 错误码，ErrorMessage 错误详细信息

TestData：多个测试点的数据，每个测试点的数据格式跟“获取测试最新数据”调用相同

注意：

一次最多返回 500 个数据编号的数据

返回例：

```
{
  "ErrorCode": 0,
  "ErrorMessage": "(E0)",
  "TestData": [
    {
      "time": "2023-08-27T11:58:58",
      "no": 5157,
      "data": [
        {
          "channelNo": 1,
          "value": 13.85418701171875,

```



```
        "status": 0,  
        "alarm": 0  
    },  
    {  
        "channelNo": 2,  
        "value": 1.4002183675765991,  
        "status": 0,  
        "alarm": 0  
    }  
]  
},  
{  
    "time": "2023-08-27T11:58:59",  
    "no": 5158,  
    "data": [  
        {  
            "channelNo": 1,  
            "value": 12.670540809631348,  
            "status": 0,  
            "alarm": 0  
        },  
        {  
            "channelNo": 2,  
            "value": 1.8398913145065308,  
            "status": 0,  
            "alarm": 0  
        }  
    ]  
}  
]  
}
```