Mars 数据库快速入门

Mars 数据库整个分成开发、运行2个部分，两者之间相互独立。数据库的开发用于管理数据库支持的变量、安全信息、接口信息等；数据库的运行负责数据的采集、存储、对外提供实时、历史数据服务。

数据库开发支持分布式、多客户端、多数据库开发；开发客户端目前仅支持在Windows下运行的桌面客户端；服务器则是跨平台的，对外提供GRPC、Web API接口，方便第三方进行扩展接入。

数据库运行支持跨平台、多种类型数据访问接口。支持有Web API、GRPC、自定义私有协议接口、OPC UA。数据库运行时采用数据库核心服务和API接口独立分开的多进程设计，API 服务和数据库核心功能可以在一台机器上也可以分别在不同的机器上；同时运行有多中类型的API接口、同一个种多个API接口同时运行，以支持多种、多个类型的客户端同时访问。

数据库运行时不支持直接添加、删除、修改变量，但支持数据库的热启动（即在不退出重启的情况下，可以动态加载生效数据库库开发时新增、修改的变量，不支持删除的变量动态生效）。

历史数据支持定时导出功能，由于采用独立的文件存储设计，历史数据的导入，只要将相应的文件拷贝到历史存储目录即可。

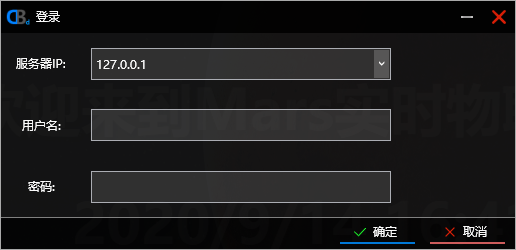
数据库支持双机冗余热备功能，历史数据路径存储到第三方的物理磁盘上，实时数据支持实时数据同步。

# 数据库开发

数据库的开发主要包括开发服务（InDBStudioServer）和开发客户端(InDBStudio)2部分。我们可以通过开发客户端进行界面画的开发配置、也可以直接通过InDBStudioServer提供的命令行，进行快速的创建。

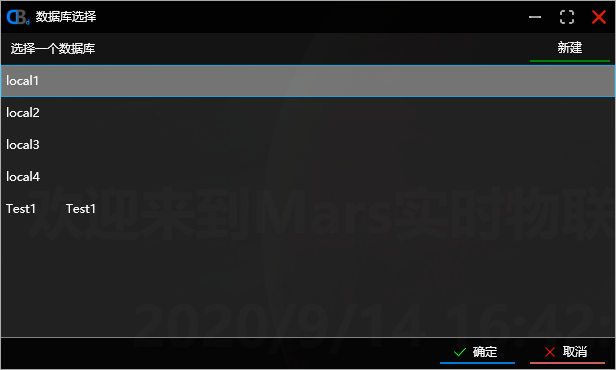
## 通过DBInStudio 管理数据库

打开DBInStudio 程序后，单击工具栏上的“登录”菜单，登录到指定的服务器上，如下图：



对于本机开发，则服务器IP选择默认的127.0.0.1 即可。用户名、密码默认都是Admin(注意大小写)。如果采用本机开发，在单击“确定”后，会自动启动DBInStudioServer 进行，如果发现没有启动的话。

登录成功后，则进入数据库新建、选择界面，如下图，可以选择开发一个已有的数据库或者新建一个。



上图中，单击右上角的“新建”按钮，则可以新建一个数据库。

选择、新建完数据库后，则进行数据库开发的主题界面了，如下图。



左侧功能树，包括以数据名称命名的节点和服务器安全节点2部分。数据库节点用于管理配置该数据库所有的数据，服务器安全节点用于配置开发服务器的账户安全信息。

### 变量的开发

在左侧的功能树种，选择“变量”节点则可以开始变量的开发了。变量支持分组，选择“变量”后，可以通过功能树下侧的“添加”按钮，添加子组。选择一个子组或根组“变量”节点后，右侧工作区则用于改组变量的管理。如下图：



支持多类型变量的筛选、编辑模式的切换、变量的导入、导出、功能快捷键等功能。

编辑模式：逐行模式、逐列模式。逐行模式，选择单位是整行，复制、粘贴也都是整行处理。逐列模式，选择单位为一个单元格，可以对某几个单元格进行复制、粘贴。

转换：用于在驱动将数值写入到数据库时，对数据所进行的数值变换，目前支持y=kx+b 线性变换。

变量记录类型：定时记录、值变量记录、驱动记录。定时、值改变记录的精度均为1S，频率超出部分将会被忽略。驱动记录：历史记录的发起者不在是数据库自身，而是有驱动什么时候采集到数据，改类型可以实现历史数据的断点续传功能。在网络不好状态下，历史数据先有外部存储起来，等到网络恢复正常后将存储起来的历史数据补录到数据库中。

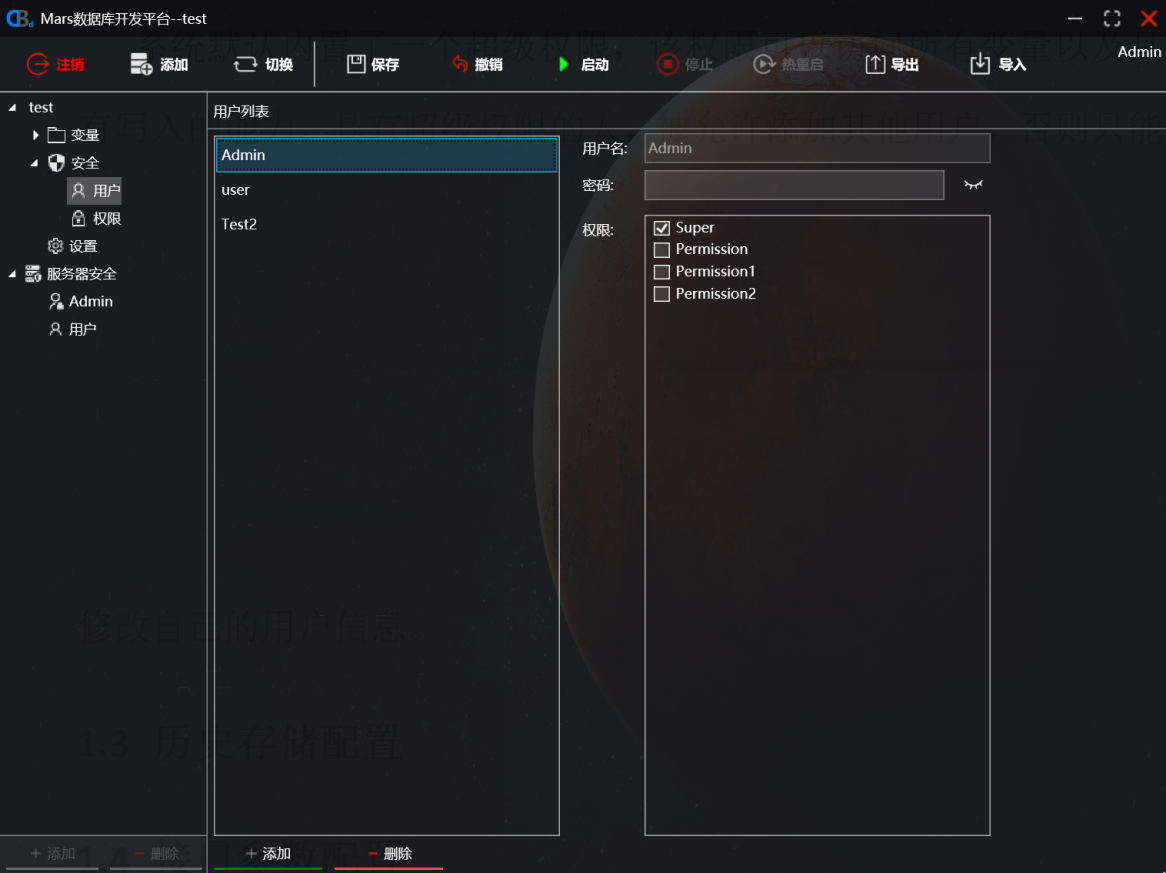
变量压缩类型：支持无压缩、无损压缩、死区压缩、斜率压缩4种。无压缩在执行最快，但是占用磁盘较大；死区、斜率压缩是有损压缩，会导致数据损失，但是压缩效率高；无损压缩则介于两者之间，压缩率不高（Double型变量能够达到6:1的压缩率），但是能够保证数据的准确性。

驱动：目前支持Sim，Spider 2种驱动，Sim 是一个模拟驱动，可以模拟Sin、Cos、Step、Square、DateTime等类型的数据；其中Step 是整数递增变换（0-360），Square 是每个1分钟反转一次，DateTime 取当前时间。Spider 驱动是配合Mars 数据库的一个专用的采集程序，详情参考Spider项目。Sim驱动需要寄存器配合来指定数据源，Spider 则是通过在外部的Spider 采集程序中指定。

### 安全配置

Mars 数据库安全分为用户和权限2部分。用户可以自己定义权限，权限的定义需要指定该权限能够访问的变量组以及是否具有写入的能力。在添加用户时，可以为用户分配相应的权限，从而使得该用户具有访问某些变量的能力。

系统默认内置了一个超级权限，该权限可以访问所有变量以及具有写入的能力。



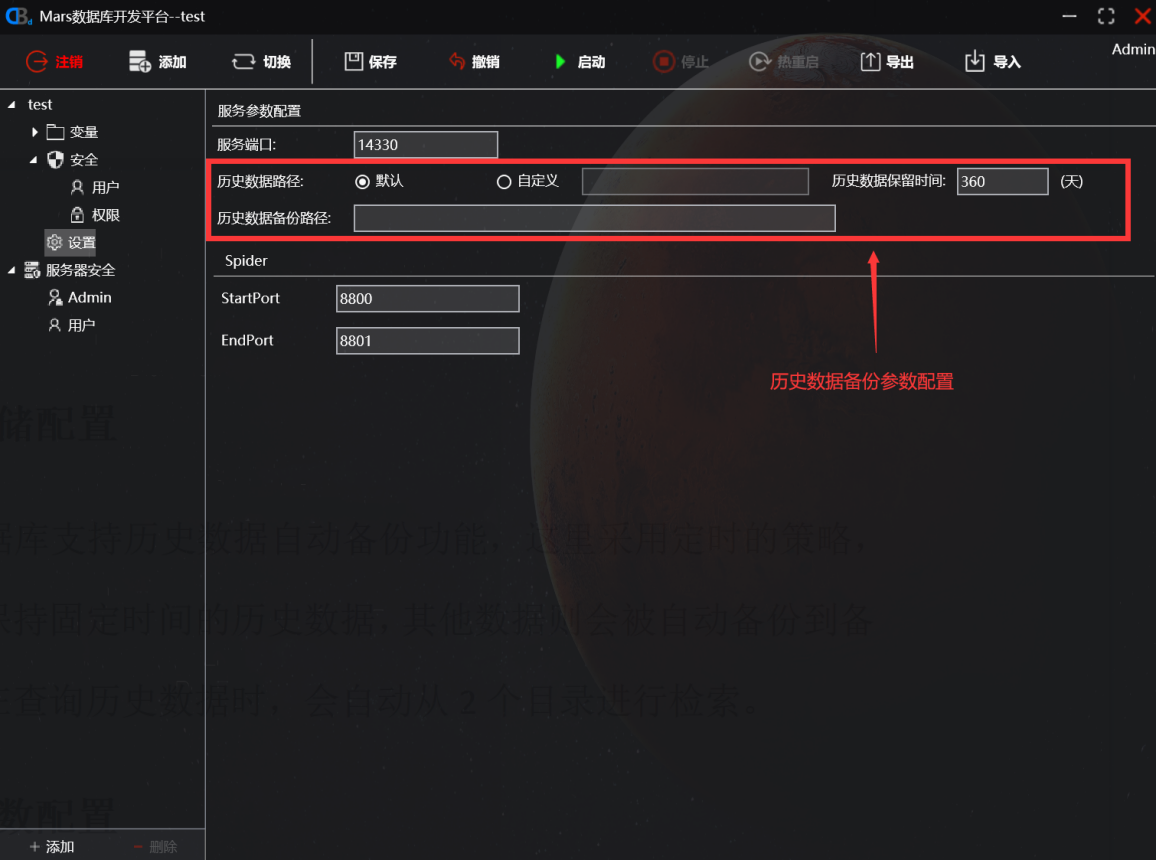
数据库用户管理



数据库权限配置

### 历史存储配置

Mar 数据库支持历史数据自动备份功能，这里采用定时的策略，即本地始终保持固定时间的历史数据，其他数据则会被自动备份到备份目录里。在查询历史数据时，会自动从2个目录进行检索。如下图：



历史数据默认存储路径为，数据库配置目录下的HisData 目录。

### 接口参数配置

这里主要配置数据面向API Proxy 提供的网络服务端口，具体API Proxy 是什么，则可以参考数据运行章节的“实时数据服务”部分。

同时提供数据库驱动所对外提供的数据服务接口，这里提供了Spider 驱动的参数配置。Spider 是专门和Mars 数据库搭配数据采集程序，具体可以参考 [Spider](https://gitee.com/chongdaoyang/Spider) 项目。具体如下图：



### 切换数据库

通过工具栏菜单中的“切换”菜单，可以重新选择或新建数据库，进行编辑。

### 数据库运行管理

当数据库的运行和开发服务器在同一台机器上时，可以通过工具栏上的“启动”、“停止”、“热启动”对运行进行控制。热启动的意思是，在不退出的情况下，加载使能增加、修改部分，删除部分必须通过冷启动方式生效。

### 数据库导入、导出

通过工具栏上的“导入”、“导出”可以对整个数据库进行导入、导出操作。在数据库导入时，有“替换”、“覆盖”，“追加”三种模式选择。

替换：删除当前数据库中的所有变量，通过导入文件的重新创建变量。

覆盖：导入变量时，发现已经存在的则以及导入文件中的配置为准，全新的则直接导入。

追加：导入变量时，如果发现变量已经存在，则忽略以数据库中的为准，如果没有则直接添加。

## 通过DBInStudioServer 快速管理数据库

DBInStudioServer 自身也提供了指令函数，通过这些指令我们可以直接在控制台中快速创建数据库。

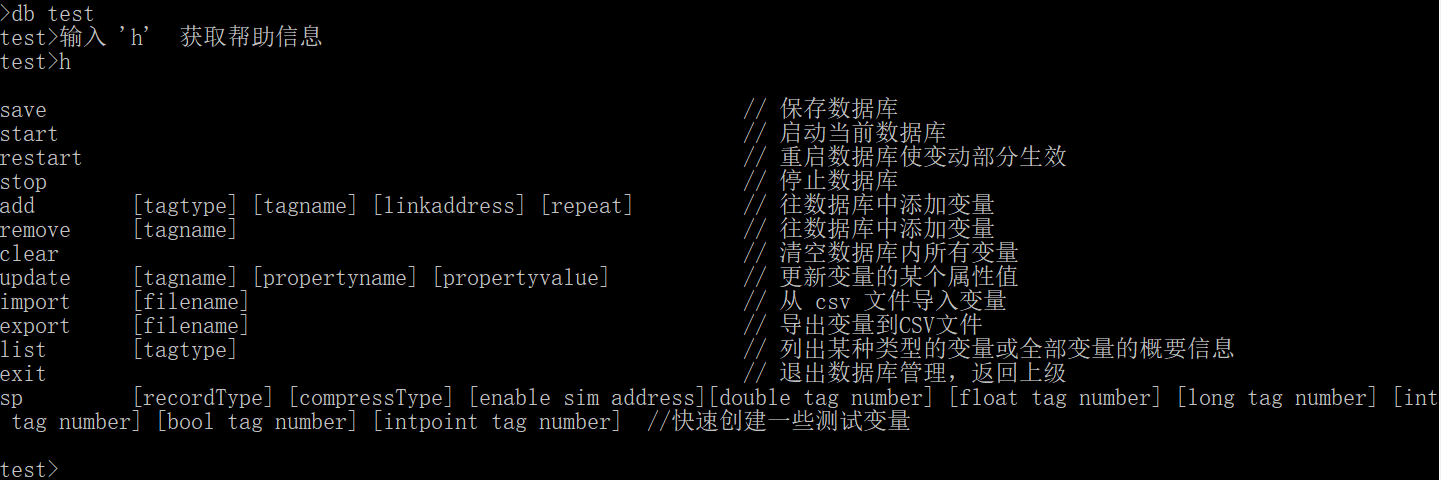
1. 数据库相关指令

db [databaseName] 新建、编辑一个数据库

list 枚举以及存在的数据库

1. 变量相关指令

通过db 指令进入一个数据编辑状态之后，就可以使用变量相关指令了。如下图：



其中SP 指令我们可以快速批量创建用于测试目的的数据库变量，方便我们在做数据库性能测试时使用。

recordType 取值: 0--定时 1--值改变 2--驱动

CompressType 取值: 0--无压缩 1--无损压缩 2--死区压缩 3--斜率压缩

Enable sim address 取值: 0 -- Sim 驱动 1--Spider 驱动

# 数据库运行

将开发好的数据库连同Mars 数据库程序集一起拷贝的要运行的目标机器上，即完成了数据库有开发到运行的部署。数据库的存放路径始终在Data 子目录里，不可更改。每个数据库以目录（目录以数据库的名称命名）的形式存在，改数据库所有文件在改子目录里。

## 启动数据库

要运行数据库，首先打开DBInRUN 进行，然后在命令行里输入：

Start XXX 命令以启动XXX数据库。如下图所示：



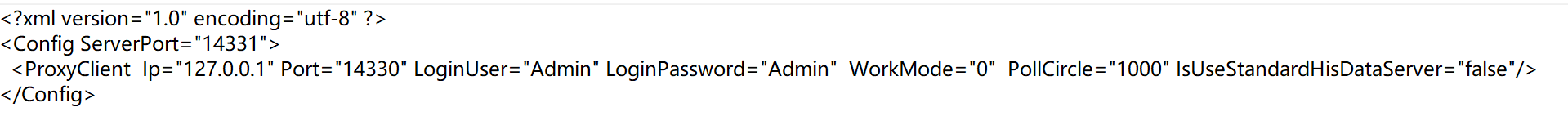
DbInRun 同时也支持 数据库名称 以参数的形式，在启动DbInRun的时候直接传递进来，从而快捷启动数据库。

具体如下，可以在CMD 指令输入: DbInRun test 达到启动test 数据库的目的；我们将DbInRun test 写入到一个bat 文件中，直接双击这个bat 文件可以达到直接运行test 数据的目的。也可以做一个DbInRun 的快捷方式，在快捷方式中指定test 参数，运行的时候直接双击快捷方式。

## 实时、历史数据服务

Mars 数据库通过各种API Proxy提供各种API接口，目前提供GRPC、WebAPI、私有协议、OPC UA接口。API Proxy 和数据库核心见通过网络进行高速通信，然后将实时数据缓存下来之后，再提供实时数据服务；历史数据直接将请求转接给数据库核心，进行历史数据查询，也可以转接给专门的历史数据服务上进行查询。

API Proxy 和数据库核心进行网络通信的参数配置在Config目录下，以API Proxy 名称命名的扩展名为.cfg 的文件中，例如:WebAPI 的配置存放在Config/DbWebApi.cfg 文件中。配置文件格式如下图：



其中ServerPort 表示的是本API Proxy 对外提供服务的接口；ProxyClient 中配置的是和数据库核心进行网络通信时，使用的通信参数。

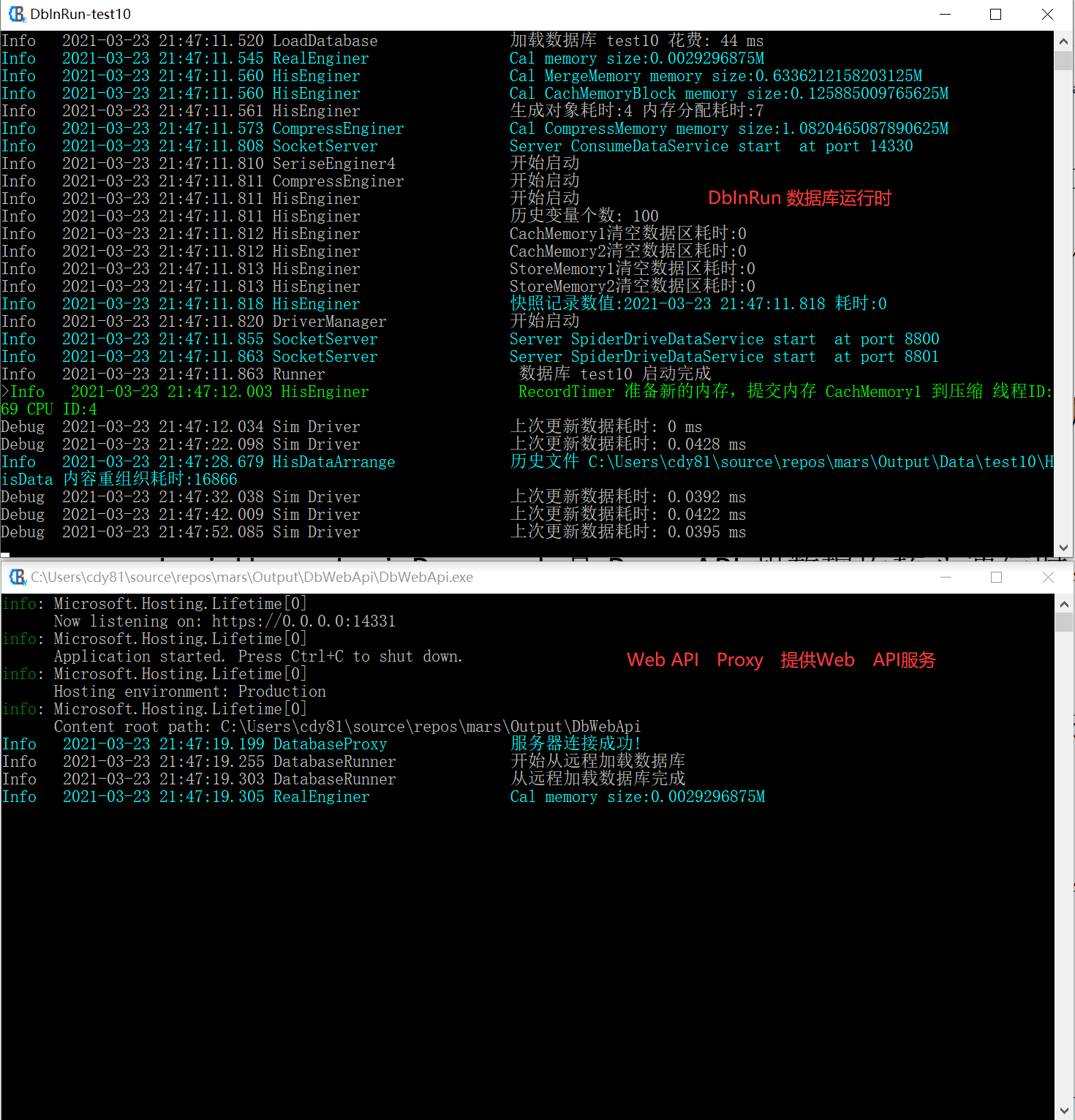
IP 表示Mars 核心所运行机器所在电脑的IP地址，Port 表示服务的端口。

LoginUser，LoginPassword 是Proxy API和数据库核心通信时，登录所用的账户信息，这里建议使用具有较高权限的账户。

WorkMode 表示双方交互的方式，0：Proxy API 定时获取数据，定时的周期区Poll Circle的值，1：表示数据库核心主动推送。

IsUseStandardHisDataServer 表示在进行历史数据查询转发时，是发给数据库核心，还是专门的历史数据服务。

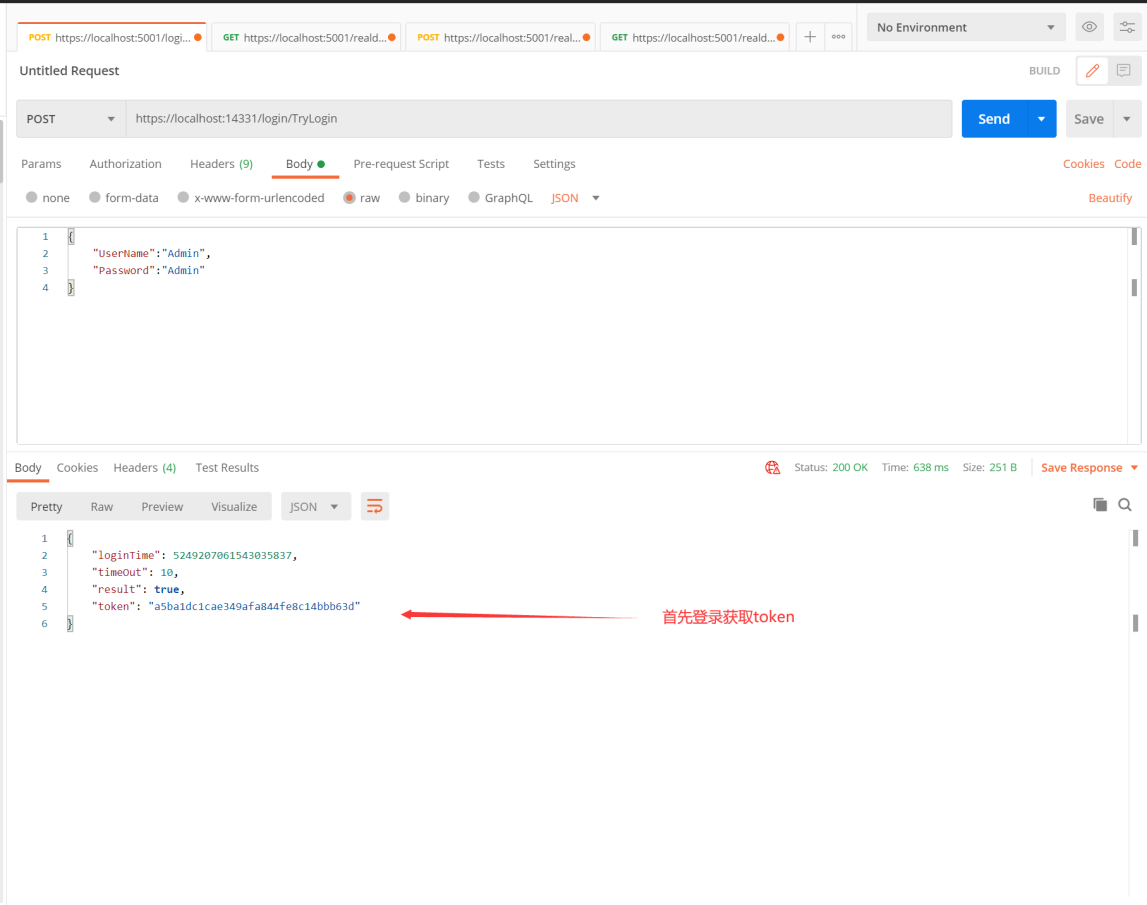
在配置完成上述参数后，直接启动相应的Proxy API接口程序即可。如下图，DbInRun 和DBWebAPI 联合一起，实现了实时、历史数据的Web API　服务。DBInRun　和DBGRPCAPI　一起提供GRPC　API　数据服务。



下面通过Postman来进行Web API测试。

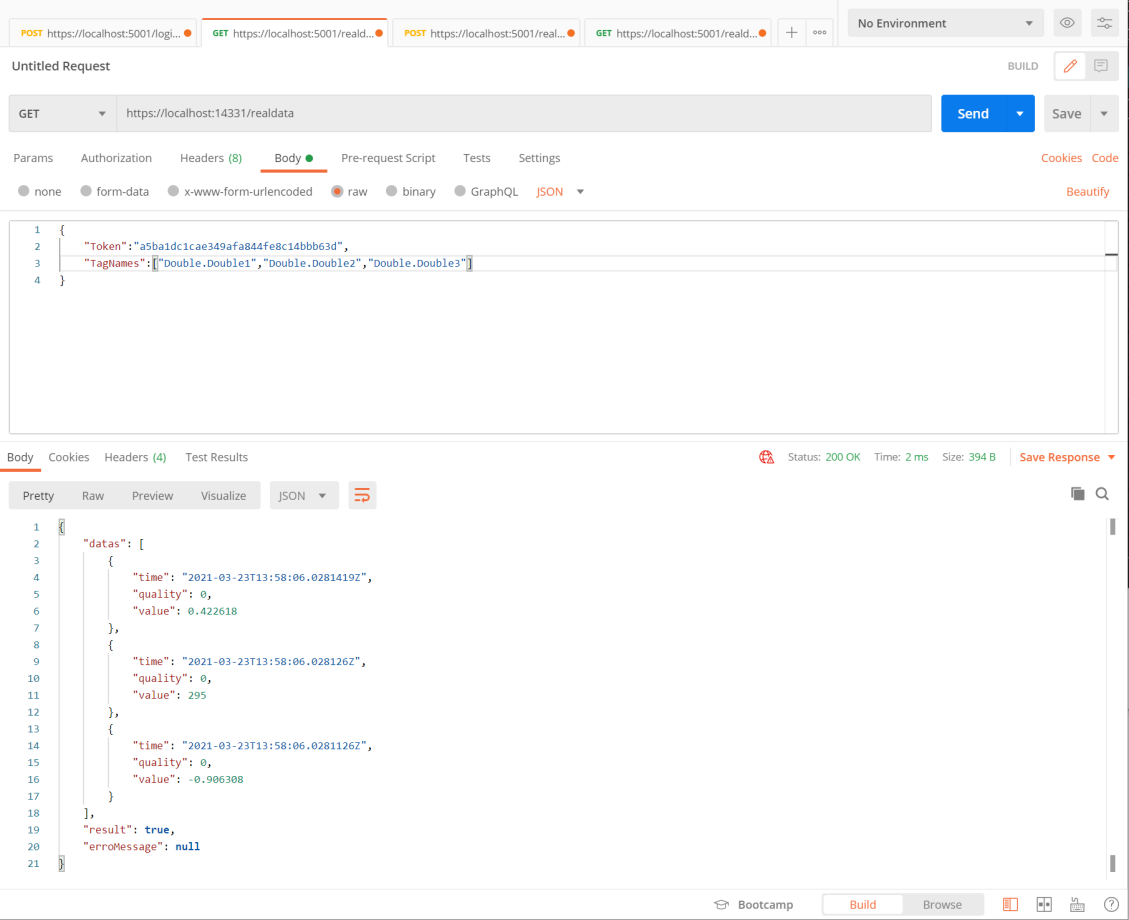
1. 首先登录，请求的地址为：

<Https://localhost:14331/login/TryLogin> 内容格式参考下图，

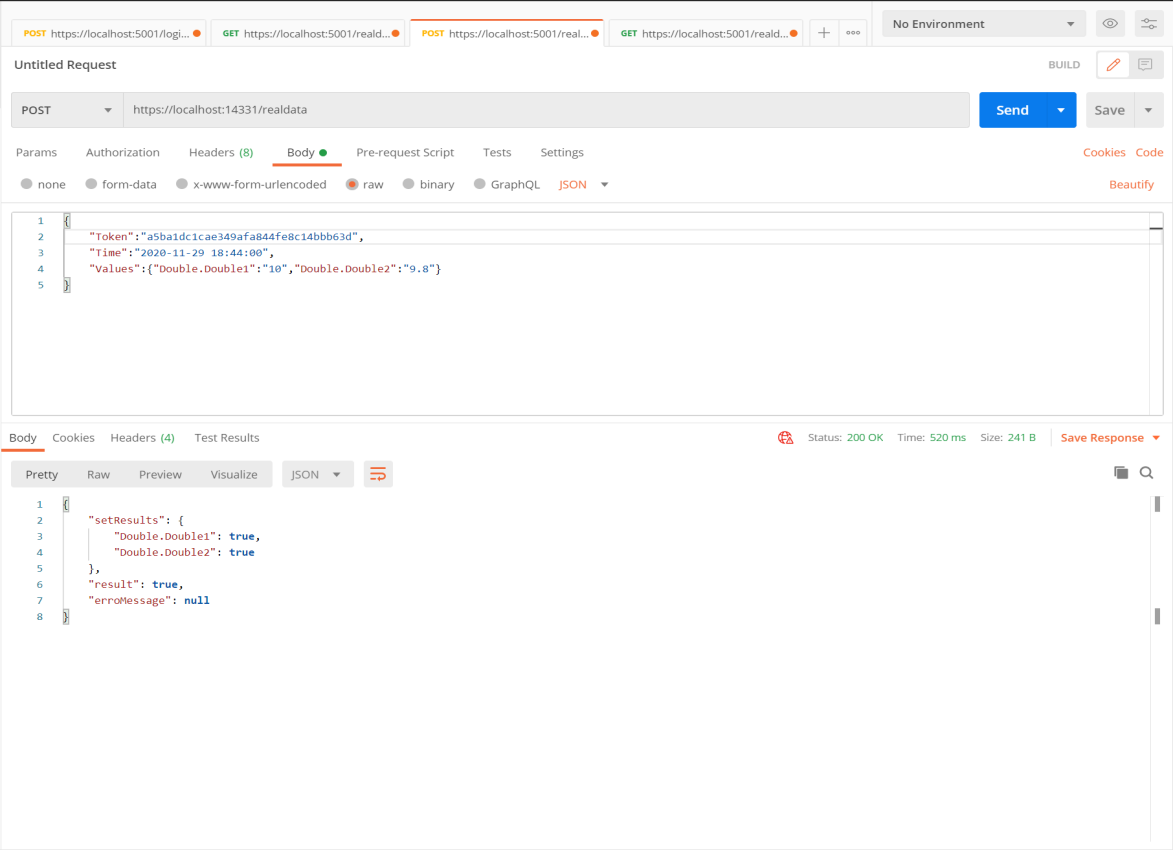


1. 获取实时数据

请求地址：https://localhost:14331/realdata

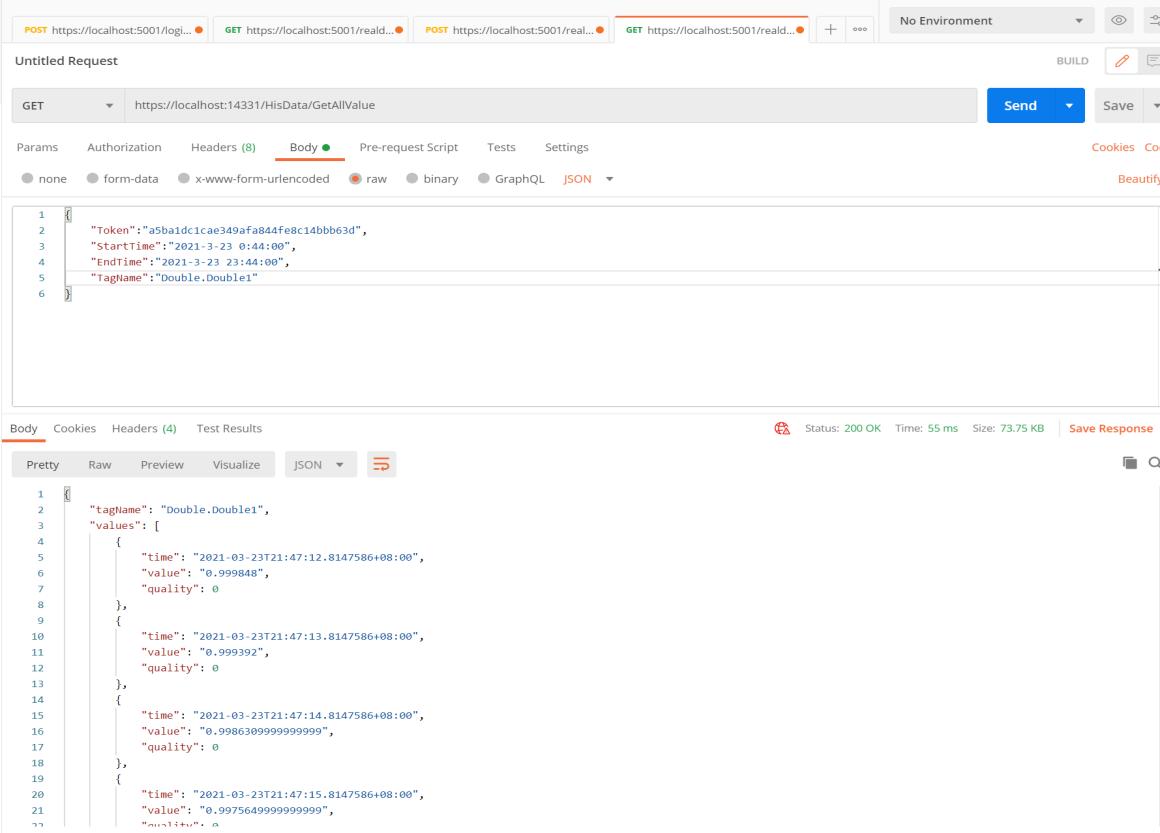


1. 设置实时数据

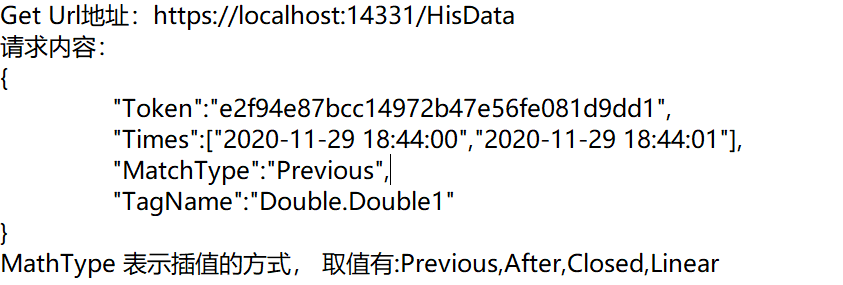


1. 获取历史数据

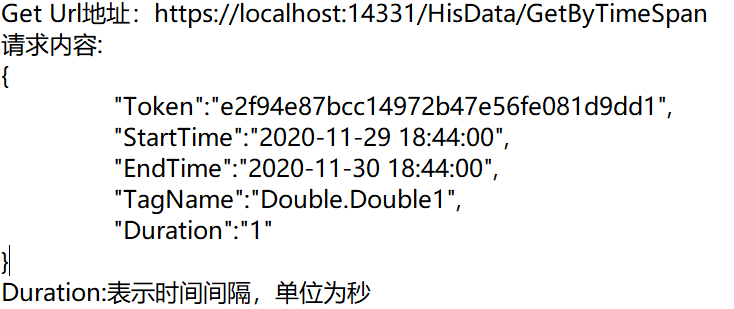
获取某个时间段，所记录的所有历史值。



1. 获取某个时间点的数据

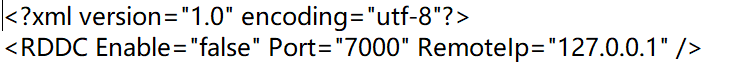


1. 以固定间隔读取某个时间段内历史值



## 双机冗余备份

在两台机器上准备相同的数据库，然后分别修改数据库所在目录下的RDDC.cfg 配置文件，格式如下：



Enable ：是否使能冗余

Port：双方网络通信端口

RemoteIp ： 对端所在机器的IP地址

注意：这里相互冗余的两台机器，只做了实时数据的数据同步，历史数据通过将数据存储到第三方存储空间上，实现数据同步。

# 数据分析工具

HisDataTools 是一个用户查看、分析历史数据的工具，支持以表格和图标两种方式进行数据查询。如下图：

