Redis客户端可视化显示GOB对象

1、背景

在Go程序中，经常使用一种GOB的对象序列化编码格式，这样格式对于Go语言来说编解码速度都非常快，所以在存储对象的时候进程会用到GOB这种格式，但是因为GOB本身是一种二进制编码的格式，对可视化非常不友好，在Redis客户端中一看全是二进制，难以阅读，在定位问题时也很难直接从数据库数据中看出一些字段值是多少，问题解决办法有很多，比如自己写一个Go程序的通过redis client读数据然后gob反序列化再json格式化再展示，这当然也是可行的，就是自己写的客户端可能在实用性上还有一定的局限性，对此，我们可以基于现有的redis gui工具，去扩展GOB对象展示的能力。

2、选型

这里我选了一款开源的比较热门的redis桌面版工具，redis desktop manager

<https://github.com/uglide/RedisDesktopManager>

这个工具是基于QT框架写的，通过QT来编译，官网未编译版本必须付费才给下载，所以我下了个开源版的自己编译来使用，因为该软件支持value的自定义格式化显示，所以非常方便我们进行二次定制开发

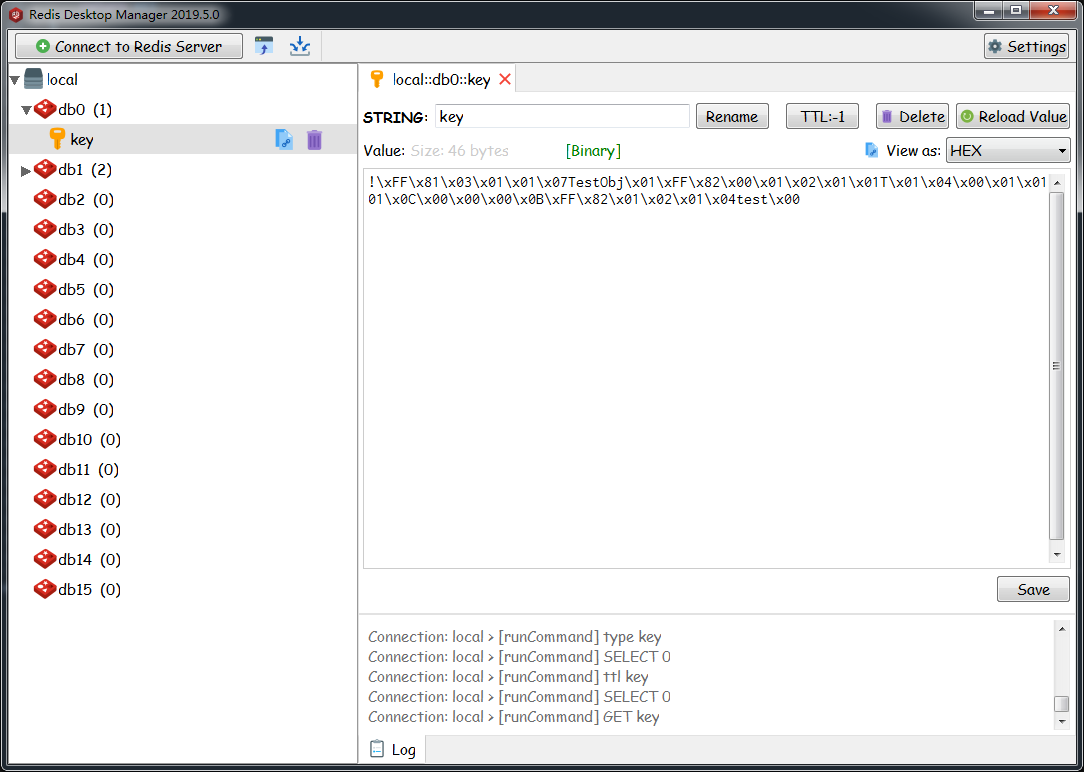
3、准备

使用前请安装好Python3 64位，Go 64位，mingw64，以及gcc工具链

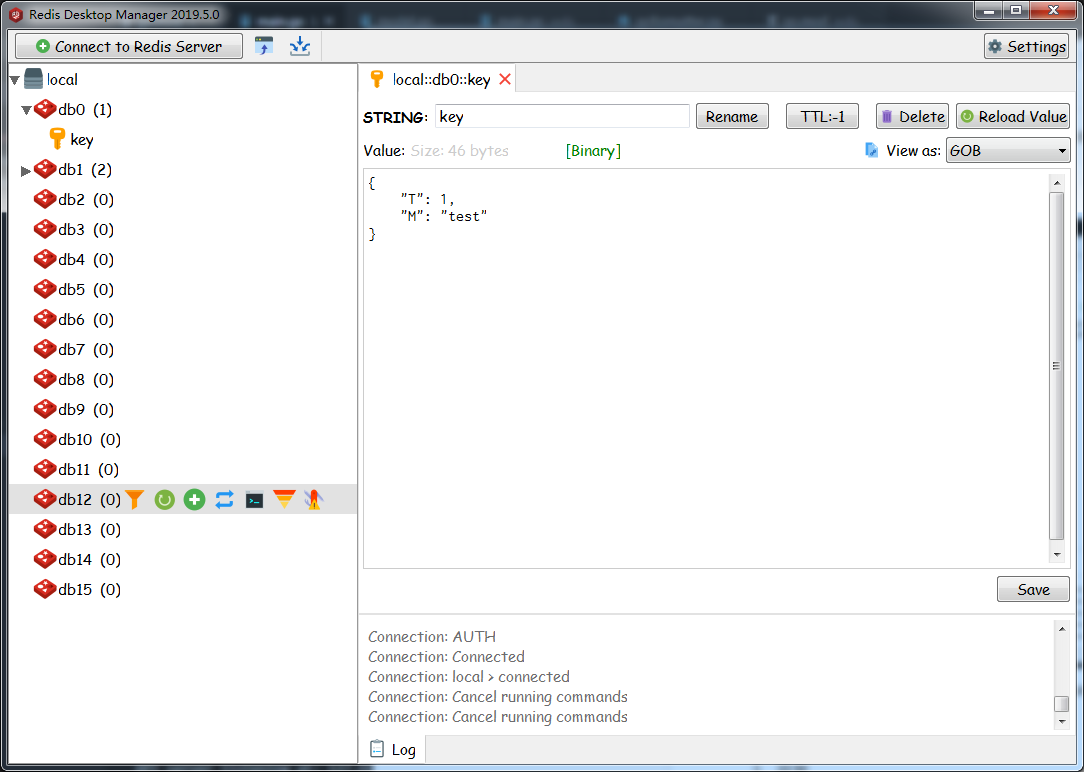
pip install -r requirements.txt安装好python依赖

3、教程

先介绍成果了



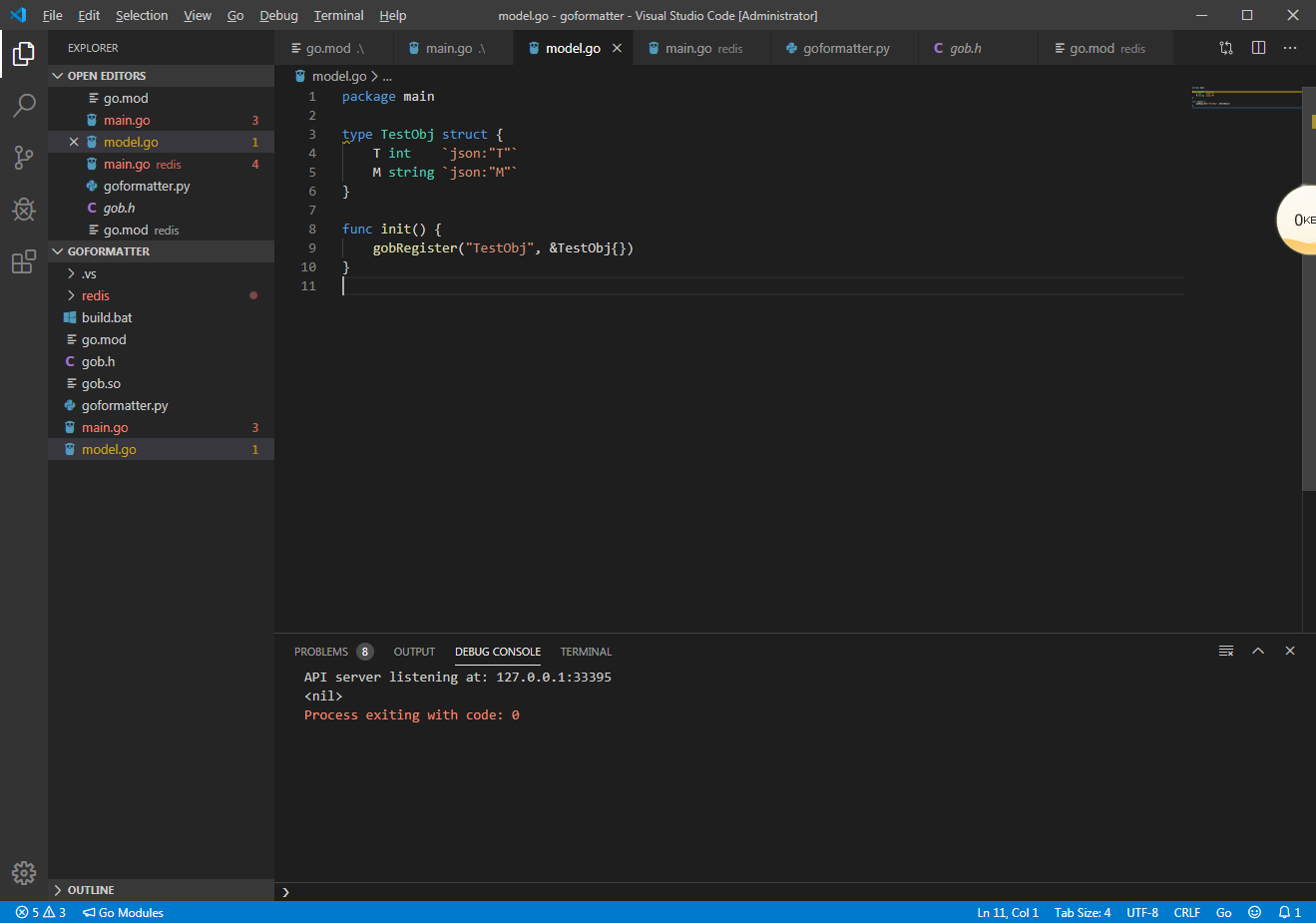
在选择一个key的时候，右边会显示对应的value，因为是二进制的编码格式，所以看起来就是一堆乱码



然后view as的时候选择GOB，就可以之间显示成json的内容了，是不是很方便很直观，当然这个也是需要二次开发的，因为GOB不像json格式一样不需要知道原始的结构定义。

4、定制结构

下载goformatter压缩包，解压，在model.go文件下或者在文件夹下新建一个xxx.go文件，把你的结构体定义以及json标签都写好，调用gob注册一下即可，但是还是会存在一些问题，对于非interface{}类型进行encode，解码时不能直接通过interface{}类型来解，需要明确的结构体对象类型，interface{}在encode时会大一些，所以可能很多人还是会选择按结构体来存，这时候只能通过去猜，一个一个结构体来试一下，确认对象是否能解，然后展示，问题是只要字段名称能对上一个，Gob也认为解正确了，所以这种碰撞解法并不会很好用，如果不嫌麻烦的话还是建议多加一个解码标签



gobRegister注册一个对象，前面的名字用于解码时关联，后面就是一个对象

如果是通用的Gob解码这一步就结束了，编译

go build -buildmode=c-shared -o gob.so (在mingw64环境下进行)

替换RDM工具的formatters目录中的gob.so即可

5、进一步定制

上面比较简单的解码方式可能不能满足要求，下面就把那个对象名称用上，使用自定义解码，同样在formatters目录下，找到\_\_init\_\_.py文件



新增一个标签，OK，自定义解码就完成了，重启RDM即可

6、开发

上面的步骤其实已经足够Gob解码使用了，下面就是一些开发的记录，选择性忽略

RDM的formatters是通过C调Python实现自定义格式化的脚步，接口也很简单就两个，encode和decode，注册个类型，就可以在界面上选择和使用了

然后上面的效果就是C调Python，Python调C，C调Go，实现的，整个流程就是要了解一些跨语言调用的一些接口，C作为桥梁，还是挺麻烦的，尤其注意字符串的内存释放，不然查着查着内存就满了

我编译的都是64位的，包括Python也是用的64位版本，用的是Python3，所以注意Go也要编译64位版本的so。如果是单纯就想用编译好的RDM的，可以把formatters目录下的bk恢复回来就行了，不恢复也不影响使用