

## Storage 类接口补充说明

### 1.Storage 类中 writeToFile 和 readFromFile 操作的存储数据的两个文件的数据格式有什么要求呢？

数据文件保存在工程目录的“data”文件夹下；另外，规定两个文件的数据格式要求如下：

- a) 每个项都必须加双引号；
- b) 一条完整的记录中间不会出现换行符，但末尾需加上一个换行符，即一行一条记录；
- c) 输入的项的值不会包含有双引号，“&”字符和逗号；
- d) 数据文件列的顺序是固定的。（不能自己改动否则会机器测试不通过）
- e) user.csv 文件中每条记录的格式是

"<username>","<password>","<email>","<phone>"

一个用户名为 name,密码为 123，邮箱是 123@123.com,手机是 123456 的记

录在数据文件中的记录如下

"name","123","123@123.com","123456"

- f) meeting.csv 文件中每条记录的格式是

"<sponsor>","<participators>","<start Date>","<end Date>","<title>"

一个发起者为 A,参与者有 B 和 C，邮箱是开始时间是 2016-08-01/00:00,结束时间是 2016-08-01/12:00，标题为 meeting 的记录在数据文件中的记录如下

"A","B&C","2016-08-01/00:00","2016-08-01/12:00","meeting"

### 2.下面这个函数接口的 filter 和 switcher 是什么？其他的接口中的又是什么？

```
int updateUser(std::function<bool(const User&)> filter,
               std::function<void(User&)> switcher);
```

回答：

filter 和 switcher 是 c++11 里引入的，所以编译的时候需要加上参数-std=c++0x 才能进行编译；从接口可以看出，filter 和 switcher 也就函数 updateUser 的参数而已，也是 function 模板类的一个实例（可以把 function 看成 vector 来理解）；

就 filter 来说，function<bool(const User&>指的是：filter 是一个返回值为 bool、参数为 const User&的函数；而 switcher 就是：返回值为 void、参数为 const User&的函数；至于这两个函数到底实现什么样的功能是由调用 updateUser 这个函数时传给 updateUser 的参数决定的；现在你可以把 filter 看成是一个过滤函数，当 filter 返回 true 时就对这个 User 进行 switcher 函数操作（注意，阶段二的时候会使用 lambda 表达式编写 filter 和 switcher 函数，到时就会理解）；

在其他函数接口里面的 filter 也和上面讲的 filter 相似，比如 function<bool(const Meeting&>;就是说：传进一个 Meeting 类给 filter 函数，返回 true 或者 false；根据返回的 true 或者 false 来继续后面的操作；至于需不需要继续后面的操作，以及 filter 什么时候返回 true 或者 false 都是由你自己决定的；

比如在 int deleteMeeting(std::function<bool(const Meeting&> filter);这个接口里规定：

当返回 filter 为 true 时，删除掉 meeting；那么在进行阶段二里调用 deleteMeeting 的函数时你传给 deleteMeeting 的 filter 函数就是把 Meeting 对象实例当作参数，并判断这个 Meeting 实例是否需要删除，如果需要删除的就返回 true，否则返回 false；

#### 4.关于 sync 这个接口，到底是用来干什么的？

Sync，意思是同步，同时；具体就是提供一个立即把保存在内存里的数据写入到文件，防止文件数据的不一致（为什么？所有访问数据不都是通过 Storage 来访问的吗？怎么会不一致？答：那些文件也是存储在操作系统，也就是电脑磁盘上的，其他程序也是可以打开

那个文件来进行读的操作，为了防止其他程序读这个文件时数据的不一致，调用这个函数立即对文件进行写入来进行文件同步）；所以在这个接口函数里面进行写操作（writeToFile）的调用即可（如有其他说明会另行通知）；