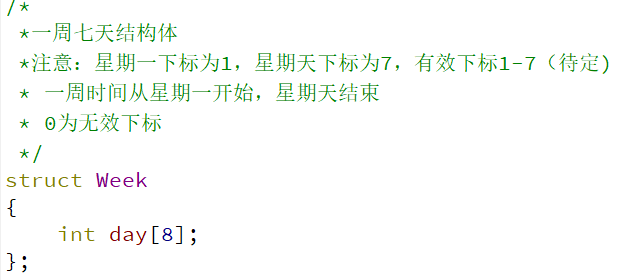
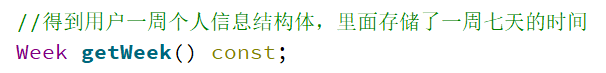
计时器后台接口文档（1.0正式版）

更新内容：

1. 用户类

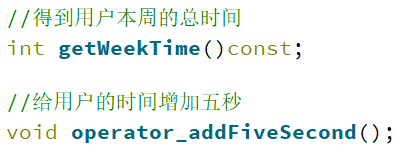
一周七天结构体





使用Week结构体的day数组分别表示每一天的秒数

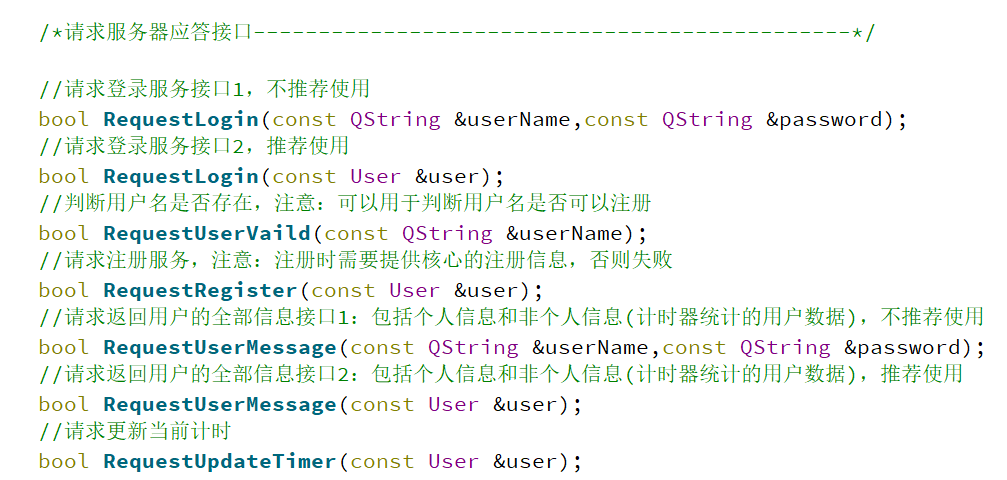
WeekTime为本周的总秒数



调用该函数将自动将该用户的今天时间增加五秒（有其他需求可以提出）

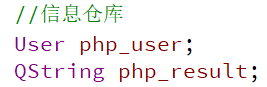
1. phpServer 类

申请服务器应答无需创建对象，直接调用静态方法getPhpServer()返回服务器指针。服务器指针只有一个，可以避免浪费空间和减少操作。



全部接口都会返回一个bool类型值，用来告诉程序员是否已经发送出了应答。

服务器的信箱

****

User 是用户信息包，result存储着的是服务器应答的结果

使用User getUser()从服务器中得到用户信息包

使用QString getResult() 从服务器中得到服务器应答结果

以上及以下方法全部需要通过服务器指针调用，通过静态方法返回服务器指针

PhpServer \* phpserver = PhpServer::getPhpServer();

在通过服务器指针可以调用所用的接口。

Phpserver->getUser();

Phpserver->getResult();

例子：

想要使用登录服务：

1. 直接调用静态方法

C:\Users\82583\AppData\Local\Temp\1517631571(1).png

这个是登录接口1，传入两个参数，一是用户名，二是用户密码

1. 登录服务只会返回一个服务器应答结果

因为应答完毕后服务器将应答异步存在服务器信息中，并发出信号

C:\Users\82583\AppData\Local\Temp\1517632000(1).png

因此需要绑定信号与槽。

在槽函数用利用int getServerStatic（）可得到服务器当前存放于仓库中的数据类型。

C:\Users\82583\AppData\Local\Temp\1517632156(1).png

以上状态中，FREE表示服务器当前处于空闲状态，仓库中为无效信息。

LoGIN 表示服务器中正在进行登录应答服务，仓库中存放的是登录信息的应答内容。

因为登录服务并不会反回用户数据包，因此此状态中用户信息包为无效数据。

setServerFree() 函数将清空服务器的仓库，释放服务器，在使用完服务器后应该将其调用，不然服务器将因为仓库已满（虚假状态）永远无法再次服务。

以上，详见代码。