**客户端离线数据上传解决方案**

**引言**

在网络广泛普及的今天,各行各业的发展都对网络产生了很大的依赖性.瞬时的断网可能会对一些行业造成巨大的损失,如银行,金融等数据量较大的行业.断网或者网络传输失败都可能会使一些即时产生的关键数据丢失,导致一些问题或者纠纷产生.为了保障一些行业中数据即时产生、发送、存储的可靠性,尤其是客户端即时产生的数据能安全上传,使用一种安全的离线数据存储和自动重传机制非常重要.

**离线数据存储技术**

SQLite是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。它能够支持Windows/Linux/Unix等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java等，还有ODBC接口，同样比起Mysql、PostgreSQL这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

不像常见的客户-服务器范例，SQLite引擎不是个程序与之通信的独立进程，而是连接到程序中成为它的一个主要部分。所以主要的通信协议是在编程语言内的直接API调用。这在消耗总量、延迟时间和整体简单性上有积极的作用。整个数据库(定义、表、索引和数据本身)都在宿主主机上存储在一个单一的文件中。

CppSqLite对SQLite的API进行了一次封装，让开发人员更加方便的使用SQLite。

**自动重传**

在程序设计时使用定时器按时轮询检查SQLite数据库,如果存在数据则自动重发上传,避免传输失败后数据的丢失.

**数据上传解决方案**

存在的问题:上传时数据类型不统一,数据比较零散,对应不同数据还要给出对应的处理信息,而使用数据库时要将数据规格存储.

解决办法:数据库的设计只需给出关键字段,将一次提交的数据统一打包成一种类型作为数据字段存储(即JSON数据),使用JSON将数据和对应的方法存储,发送成功后再对数据按照JSON解析,使不同的数据可以按对应的类型存储到另一端数据库中.

数据库的设计:

字段: id: INT

Data: TEXT

UpdateTime:DATETIME

JSON格式:

{

"businesses": [

{

"logicFunction": "firstFunction",

"parms": [

{

"parm": "1"

}

]

},

{

"logicFunction": "secondFunction ",

"parms": [

{

"parm": "2"

},

{

"parm": "3"

}

]

}

]

}

businesses：业务数组，默认业务的顺序为数组的存储顺序

parms:参数数组

parm：具体参数值，必须与logicFunction对应的函数的参数一致

logicFunction:逻辑函数，用于数据解析之后，确定执行哪个函数，其参数为parms数组的值

例如，本例中先执行函数firstFunction，再执行函数secondFunction，仅有二者同时成功则表示成功，这样一来就解决了业务相关的数据重传问题。