

# Bug修改记录

## 1.19366: LHttpRequest的默认header(LHttpRequest::UserAgentHeader)返回非空

- 解决: 已修改, 删除了 LHttpRequest 中数据结构体请求头的默认值

```
void LHttpRequest::LHttpDataStruct::init() {  
    m_rawHeaders.insert("Accept-Language", "zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6");  
    m_rawHeaders.insert("Connection", "keep-alive");  
}
```

- 测试: 在 LHttpRequest 中的 test7(), 结果全部为空

```
lark5@DavidingPlus:~/Lark5/larksdk/build$ ./snippet/HttpRequestTest/HttpRequestTest  
(  
(  
(  
(  
(  
(  
(  
(  
lark5@DavidingPlus:~/Lark5/larksdk/build$ |
```

## 2.关于post请求头, 也就是剩下的4个bug

- 分析: 首先有一点是我自己的错误, 我在做复用的时候忘了把 data 段给加到 send\_message 里面去了, 导致数据没发出去, 已修改

```
// 如果携带数据添加数据进去  
if (!data.isEmpty()) {  
    // Content-Length请求头在前面处理过了  
    send_message.append("\r\n");  
    send_message.append(data.toString());  
}
```

- 然后谈谈 post 和 get 在请求头层面的区别, 主要区别就在于 post 请求可以向服务器提交一些数据, 所以在请求头里面需要多出一些默认的请求头, 比较重要的就是测试提到的 Content-Length 和 Content-Type, 用于指明数据的长度和具体是什么样的数据格式, 当然这里面我们需要注意到 Content-Length 可以被用户设置, 所以我们需要做处理, 保证这个 size 在请求头当中是正确的

```
// 从request中的rawHeaders中读取，并且设置到请求报文中
for (auto &item : request.m_pData.m_rawHeaders) {
    // 关于Content-Length做一个特判，为0则不设置进去，因为没有数据
    if ("Content-Length" == item.key().toStdString() and 0 == item.value().toInt())
        continue;

    send_message.append(item.key().toStdString());
    send_message.append(": ");
    send_message.append(item.value().toStdString());
    send_message.append("\r\n");
}
}
```

其次，Content-Type这个东西，我查询了资料，post请求就使用默认的application/x-www-form-urlencoded，put请求没找到，就不设置，让用户自己指定即可

```
// 处理Content-Type的默认值，post有，put请求暂时没找到
if (3 == op and request.header(LHttpRequest::ContentTypeHeader).isEmpty())
    request.setHeader(LHttpRequest::ContentTypeHeader, "application/
x-www-form-urlencoded");
```

- 关于bug

以下的测试都在HttpControlTestr的testPost2()中

- 19371: LHttpRequest发送post请求并携带数据 请求报文中没有Content-Length头信息

做了两个测试，一个是长度不对，一个是直接没有数据，检验我们内部对用户错误设置的修正

```
lr.setHeader(LHttpRequest::ContentLengthHeader, "12"); // 12和下面数据不相等，8是下面数据的长度
// lr.setHeader(LHttpRequest::ContentLengthHeader, "8");
lr.setHeader(LHttpRequest::ContentTypeHeader, ""); // 设置为空

LHttpRequest lc;
LHttpRequestReply lry = lc.post(lr, "&nothing"); // 1.数据和用户设置的不相等，请求头中会进行修正
// LHttpRequestReply lry = lc.post(lr, ""); // 2.直接不要数据，请求头中不会包含
```

第一个

```
-----
POST /post HTTP/1.1
Host: 192.168.1.211
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: keep-alive
Content-Length: 8
Content-Type: application/x-www-form-urlencoded

&nothing
-----
```

第二个

```
-----
POST /post HTTP/1.1
Host: 192.168.1.211
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded

-----
```

- 19370: LHttpRequest发送post请求并携带数据 请求报文中没有数据信息

刚才第一个例子，携带了数据

- 19368: LHttpRequest设置的ContentLengthHeader post请求发送的数据长度一样 程序阻塞

我设置为12或者8，不管哪一个，最后请求头中的都是12，都不会阻塞

```
lr.setHeader(LHttpRequest::ContentLengthHeader, "12"); // 12和下面数据不相等，8是下面数据的长度
// lr.setHeader(LHttpRequest::ContentLengthHeader, "8");
lr.setHeader(LHttpRequest::ContentTypeHeader, ""); // 设置为空
```

结果：

```
-----
POST /post HTTP/1.1
Host: 192.168.1.211
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: keep-alive
Content-Length: 8
Content-Type: application/x-www-form-urlencoded

&nothing
-----
```

- 19367: LHttpRequest未设置ContentTypeHeader post请求头中没有默认的Content-Type信息

上面的测试可以看出，有的