Bug修改记录 11.1

1.18903：LHttpControl访问要重定向的服务器时，重定向失败

- 原因：正常的`get`请求是没问题的，只是`get`请求如果请求到的是重定向的服务器的时候，返回`3xx`的状态码，并且响应报文的字段当中会包含`Location`字段来指出新的路径
- 解决：在`get`请求当中封装响应报文`reply`的时候做一步判断，看是否遇到了重定向的问题；如果是就在内部处理了再调用一次；`get_byte`也同步更改(当然后续可能`get_byte`会被合并到`get`当中...)

```
    close(sockfd);

    // 这里需要处理一下重定向的问题
    std::string recv_message = readMessage.toStdString();

    size_t pos_location = recv_message.find("Location");
    if (std::string::npos != pos_location) {
        size_t pos_right = recv_message.find("\r\n", pos_location);
        std::string new_path = std::string(recv_message.begin() + pos_location + strlen("Locat

        // 拟一个新的request去请求
        LHttpRequest new_request = LHttpRequest(request);
        new_request.setPath(LString(new_path));

        return get(new_request);
    }

    return reply;
```

- 测试：在`HttpRequestTest`中的`test4()`中，结果如下：

```
lzx0626@DavidingPlus:~/DavidingPlus/Lark5/larksdk/build$ snippet/HttpRequestTest/HttpRequestTest
GET /get HTTP/1.1
Host: 192.168.1.211
User-Agent: Mozilla/5.0
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: keep-alive



HTTP/1.1 200 OK
Server: gunicorn/19.9.0
Date: Wed, 01 Nov 2023 01:50:06 GMT
Connection: keep-alive
Content-Type: application/json
Content-Length: 276
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true

{
  "args": {},
  "headers": {
    "Accept-Encoding": "gzip, deflate",
    "Accept-Language": "en-US,*",
    "Connection": "keep-alive",
    "Host": "192.168.1.211",
    "User-Agent": "Mozilla/5.0"
  },
  "origin": "192.168.7.20",
  "url": "http://192.168.1.211/get"
}
```

2.18872： LHttpControl的post请求地址中包含中文数据 无任务响应结果

- 原因：`LString`的`length()`方法返回多字节字符的长度不正确，导致`send()`函数发送的长度不准确，从而不正确
- 解决：将`LHttpControl`中所有关于`LString`的`length()`全部转化为`toStdString().size()`，以保安全

```
603
604     setSend(this, sendMessage);
605     int sendCount = send(sockfd, sendMessage.toStdString().c_str(), sendMessage.toStdString().size(), 0);
606     if (sendCount < 0) {
607         reply->setError(reply, LHttpReply::RemoteHostClosedError);
608     }
609
```

- 测试：`HttpControlTest`的`testPost3()`，结果如下：

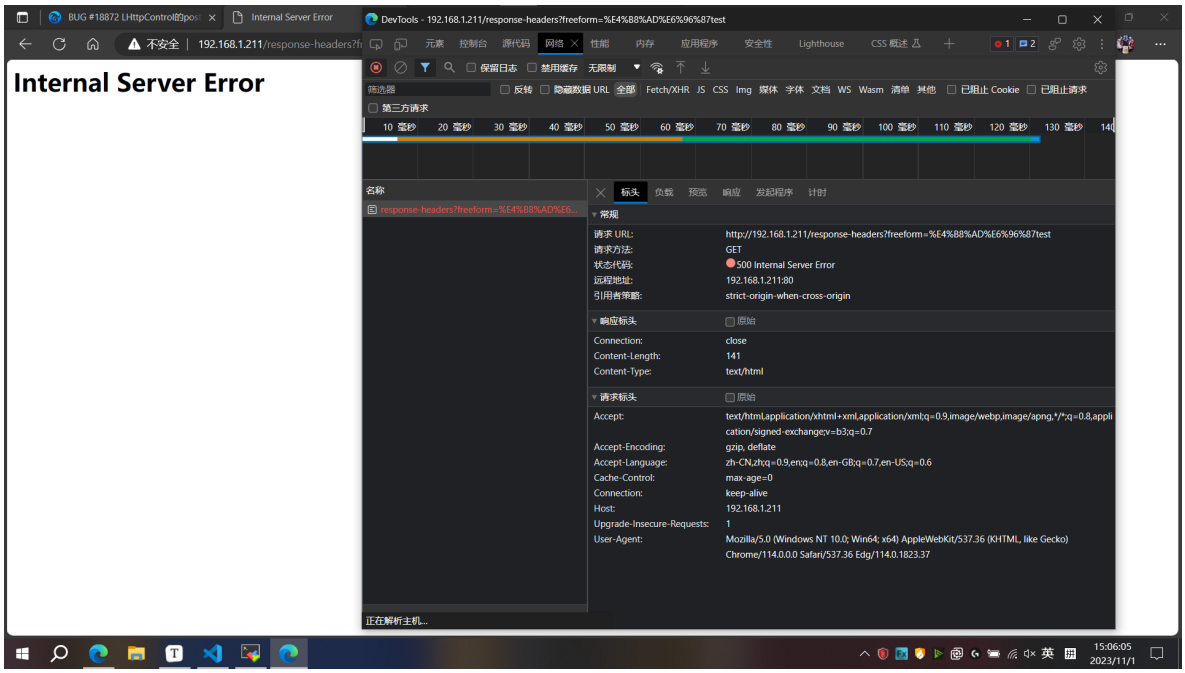
```
lzx0626@DavidingPlus:~/DavidingPlus/Lark5/larksdk/build$ snippet/HttpControlTest/HttpControlTest
500
Internal Server Error
--------------------------
POST /response-headers?freeform=中文test HTTP/1.1
Host: 192.168.1.211
User-Agent: Mozilla/5.0
Accept: */*
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 5

hello
--------------------------
HTTP/1.1 500 Internal Server Error
Connection: close
Content-Type: text/html
Content-Length: 141


--------------------------
<html>
  <head>
    <title>Internal Server Error</title>
  </head>
  <body>
    <h1><p>Internal Server Error</p></h1>

  </body>
```

返回了`500`，这也是预期之内



3.18873：LHttpControl发送put请求后 data()返回的数据有问题

- 原因：`put`请求的请求报文在写入完毕数据之后还写入了两个空行，导致格式出现问题
- 解决：已在对应位置进行修改

```
sendMessage.append("\r\n");
sendMessage.append(data);
// sendMessage.append("\r\n\r\n");
```

- 测试：`HttpControlTest`的`testPut2()`中，结果如下：

```
lzx0626@DavidingPlus:~/DavidingPlus/Lark5/larksdk/build$ snippet/HttpControlTest/HttpControlTes
t
PUT
200
OK
1
application/json
----------------------------
{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "this": "5"
  },
  "headers": {
    "Accept-Encoding": "gzip, deflate",
    "Accept-Language": "en-US,*",
    "Connection": "keep-alive",
    "Content-Length": "6",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "192.168.1.211",
    "User-Agent": "Mozilla/5.0"
  },
  "json": null,
  "origin": "192.168.7.35",
  "url": "http://192.168.1.211/put"
}

lzx0626@DavidingPlus:~/DavidingPlus/Lark5/larksdk/build$
```

4.18874：LHttpControl发送post请求后readRecv返回的数据没有响应体数据

- 原因：处理 `readBuffer` 的时候没有将响应头结束标志之后的数据正确存入
- 解决：已经在 `post` 接受数据的时候进行了正确处理

```
    while (read(sockfd, &recvChar, 1)) {
        readBuffer[cout++] = recvChar;
        if (!endOfHeaders) {
            // 检查是否读取到响应头部分的结束标志 \r\n\r\n
            if (cout >= 4 && readBuffer[cout - 4] == '\r' && readBuffer[cout - 3] == '\n' &&
                readBuffer[cout - 2] == '\r' && readBuffer[cout - 1] == '\n') {
                endOfHeaders = true;
            }
        } else {
            responseData.append(recvChar);
        }
    }
```

- 测试：`HttpControlTest` 的 `testPost3()`，和上面那个中文数据区分开，这里用的 `path` 是 `/post`，结果如下：

```
-------------------------
HTTP/1.1 200 OK
Server: gunicorn/19.9.0
Date: Wed, 01 Nov 2023 07:36:58 GMT
Connection: keep-alive
Content-Type: application/json
Content-Length: 467
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true

{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "hello": ""
  },
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "gzip, deflate",
    "Accept-Language": "en-US,*",
    "Connection": "keep-alive",
    "Content-Length": "5",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "192.168.1.211",
    "User-Agent": "Mozilla/5.0"
  },
  "json": null,
  "origin": "192.168.7.35",
  "url": "http://192.168.1.211/post"
}
```

5.18876：LHttpControl发送put/delete/get/head请求后readSend返回的请求报文格式欠佳

- 原因：的确之前封装的确实不够好，请求报文如果不携带数据段，那么请求头最后应该有一个空行；如果携带数据，那么数据后面不应该存在空行
- 解决：已按照此标准进行修改
- 测试：分别对`get`，`post`，`head`，`put`，`delete`请求进行测试，以上测试全部在`HttpControlTest`中
  - `get`，`testGet3()`

    ```
    -------------------------
    GET /basic-auth/huahua/123456 HTTP/1.1
    Host: 192.168.1.211
    Authorization: Basic aHVhaHVhOjEyMzQ1Ng==
    User-Agent: Mozilla/5.0
    Accept: */*
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
    Accept-Encoding: gzip, deflate
    Connection: keep-alive


    -------------------------
    ```

    两个空行是因为我在输出的时候有多输出了一个回车导致的，所以实际上是没问题的

    ```
    // 验证
    std::cout << "-------------------------" << std::endl
              << lry->controller()->readSend() << std::endl;

    std::cout << "-------------------------" << std::endl
              << lry->controller()->readRecv() << std::endl;
    ```

  - `post`，`testPost3()`

数据后面不存在多余空行了，就只存在我们人为进行的换行

```
--------------------------
POST /post HTTP/1.1
Host: 192.168.1.211
User-Agent: Mozilla/5.0
Accept: */*
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 5

hello
--------------------------
```

因此，后续不携带数据的输出应该类似于`get`，携带数据应该类似于`post`

- `head`，`testHead()`

  同`get`...

```
--------------------------
SendMessage:**************
HEAD /tool/params HTTP/1.1
Host: coolaf.com
User-Agent: Mozilla/5.0
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: close


--------------------------
```

- `put`，`testPut2()`

  同`post`...

```
----------------------------
PUT /put HTTP/1.1
Host: 192.168.1.211
User-Agent: Mozilla/5.0
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 6

this=5
----------------------------
```

- `delete`，`testDelete()`

  同`get`...

```
-------------------------
SendMessage:**************
DELETE /posts/1 HTTP/1.1
Host: jsonplaceholder.typicode.com
User-Agent: Mozilla/5.0
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: close


-------------------------
```

6.18902：LHttpRequest的setHeader设置的一些标头值中包含中文时 无任何响应结果

- 解决：我测试的时候能收到返回结果，可能是处理之前的细节问题的时候规范了请求报文的格式，自然就正确了吧
- 测试：`HttpRequestTest`的`test6()`，结果如下：

  标头设置为中文是不合法的，所以返回了`400 Bad Request`，响应体也说明得很清楚

```
POST
400
Bad Request

---------------------------------------
POST /post HTTP/1.1
Host: 192.168.1.211
User-Agent: Mozilla/5.0
Accept-Language: en-US,*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: eight八


---------------------------------------
---------------------------------------
HTTP/1.1 400 Bad Request
Connection: close
Content-Type: text/html
Content-Length: 168

<html>
  <head>
    <title>Bad Request</title>
  </head>
  <body>
    <h1><p>Bad Request</p></h1>
    Invalid HTTP Header: &#x27;CONTENT-LENGTH&#x27;
  </body>
</html>
```