

# 现代操作系统应用开发实验报告

姓名：余漫霖

学号：16340280

实验名称：网络访问

## 一、参考资料

<http://rapidjson.org/zh-cn/index.html>

## 二、实验步骤

1. 输入用户名和密码进行注册

在 LoginRegisterScene.cpp 中：

发送登录请求：

```
97 void LoginRegisterScene::registerButtonCallback(Ref * pSender) {
98     // Your code here
99     HttpRequest* request = new HttpRequest();
100     request->setUrl("http://127.0.0.1:8000/users");
101     request->setRequestType(HttpRequest::Type::POST);
102     request->setResponseCallback(CC_CALLBACK_2(LoginRegisterScene::onHttpRequestCompletedRegister, this));
103
104     std::string username = usernameInput->getStringValue();
105     std::string password = passwordInput->getStringValue();
106     std::string postData = "{\"username\":\"" + username + "\", \"password\":\"" + password + "\"}";
107     request->setRequestData(postData.c_str(), postData.length());
108     request->setTag("POST test");
109
110     cocos2d::network::HttpClient::getInstance()->send(request);
111     request->release();
112 }
```

登录请求后的信息回显：

```

133 void LoginRegisterScene::onHttpRequestCompletedRegister(HttpClient *sender, HttpResponse *response) {
134     if (!response) {
135         return;
136     }
137     if (!response->isSucceed()) {
138         messageBox->setString(response->getErrorBuffer());
139         return;
140     }
141     auto responseData = response->getResponseData();
142     rapidjson::Document doc;
143     doc.Parse(responseData->data(), responseData->size());
144     if (doc["status"] == true) {
145         messageBox->setString("Register OK");
146     }
147     else {
148         messageBox->setString(std::string("Register Failed\n") + doc["msg"].GetString());
149     }
150 }

```

## 2. 输入用户名和密码进行登录

在 LoginRegisterScene.cpp 中:

发送登录请求:

```

79 void LoginRegisterScene::loginButtonCallback(cocos2d::Ref * pSender) {
80     // Your code here
81     HttpRequest* request = new HttpRequest();
82     request->setUrl("http://127.0.0.1:8000/auth");
83     request->setRequestType(HttpRequest::Type::POST);
84     request->setResponseCallback(CC_CALLBACK_2(LoginRegisterScene::onHttpRequestCompletedLogin, this));
85
86     std::string username = usernameInput->getStringValue();
87     std::string password = passwordInput->getStringValue();
88     std::string postData = "{\"username\":\"" + username + "\", \"password\":\"" + password + "\"}";
89     request->setRequestData(postData.c_str(), postData.length());
90     request->setTag("POST test");
91
92     cocos2d::network::HttpClient::getInstance()->enableCookies(NULL);
93     cocos2d::network::HttpClient::getInstance()->send(request);
94     request->release();
95 }

```

登陆请求后的信息回显:

```

114 void LoginRegisterScene::onHttpRequestCompletedLogin(HttpClient *sender, HttpResponse *response) {
115     if (!response) {
116         return;
117     }
118     if (!response->isSucceed()) {
119         messageBox->setString(response->getErrorBuffer());
120         return;
121     }
122     auto responseData = response->getResponseData();
123     rapidjson::Document doc;
124     doc.Parse(responseData->data(), responseData->size());
125     if (doc["status"] == true) {
126         messageBox->setString("Login OK");
127     }
128     else {
129         messageBox->setString(std::string("Login Failed\n") + doc["msg"].GetString());
130     }
131 }

```

## 3. 获取用户信息

在 UsersInfoScene.cpp 中:

发送获取用户信息的请求：

```
58 void UsersInfoScene::getUserButtonCallback(Ref * pSender) {
59     // Your code here
60     HttpRequest* request = new HttpRequest();
61     request->setUrl("http://127.0.0.1:8000/users?limit" + limitInput->getString());
62     request->setRequestType(HttpRequest::Type::GET);
63     request->setResponseCallback(CC_CALLBACK_2(UsersInfoScene::onHttpRequestCompleted, this));
64
65     cocos2d::network::HttpClient::getInstance()->send(request);
66     request->release();
67 }
```

发送请求后的信息回显：

```
70 void UsersInfoScene::onHttpRequestCompleted(HttpClient *sender, HttpResponse *response) {
71     if (!response) {
72         return;
73     }
74     if (!response->isSucceed()) {
75         messageBox->setString(response->getErrorBuffer());
76         return;
77     }
78     auto responseData = response->getResponseData();
79     rapidjson::Document doc;
80     doc.Parse(responseData->data(), responseData->size());
```

```
81     if (doc["status"] == true) {
82         std::string info;
83         for (int i = 0; i < doc["data"].Size(); i++) {
84             info += "Username: ";
85             info += doc["data"][i]["username"].GetString();
86             info += "\nDeck: \n";
87             int s = doc["data"][i]["deck"].Size();
88             auto &deck = doc["data"][i]["deck"];
89             for (int j = 0; j < s; j++) {
90                 auto group = deck[j].GetObjectW();
91                 for (auto &g : group) {
92                     info += " ";
93                     info += g.name.GetString();
94                     info += ": ";
95                     int value = g.value.GetInt();
96                     info += intToStr(value);
97                     info += "\n";
98                 }
99                 info += " ---\n";
100             }
101             info += "---\n";
102         }
103         messageBox->setString(info);
104     }
105     else {
106         messageBox->setString(std::string("Info Failed\n") + doc["msg"].GetString());
107     }
108 }
```

#### 4. 修改个人信息

在 ModifyUserScene.cpp 中：

发送修改个人信息的请求：

```

57 void ModifyUserScene::putDeckButtonCallback(Ref * pSender) {
58     // Your code here
59     HttpRequest* request = new HttpRequest();
60     request->setUrl("http://127.0.0.1:8000/users");
61     request->setRequestType(HttpRequest::Type::PUT);
62     request->setResponseCallback(CC_CALLBACK_2(ModifyUserScene::onHttpRequestCompleted, this));
63
64     std::string deck = deckInput->getString();
65     std::string putData = "{\"deck\": " + deck + "}";
66     request->setRequestData(putData.c_str(), putData.length());
67
68     cocos2d::network::HttpClient::getInstance()->send(request);
69     request->release();
70 }

```

发送请求后的信息回显：

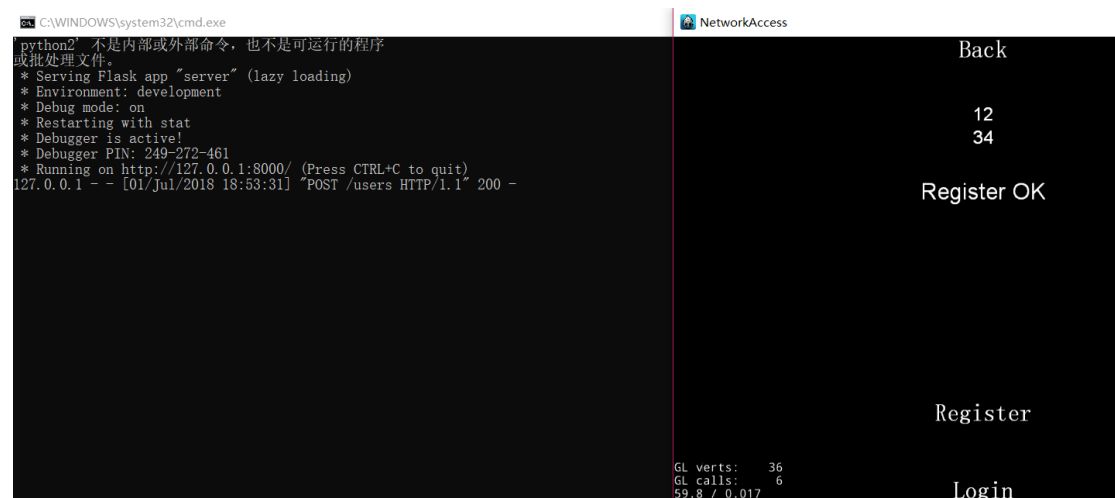
```

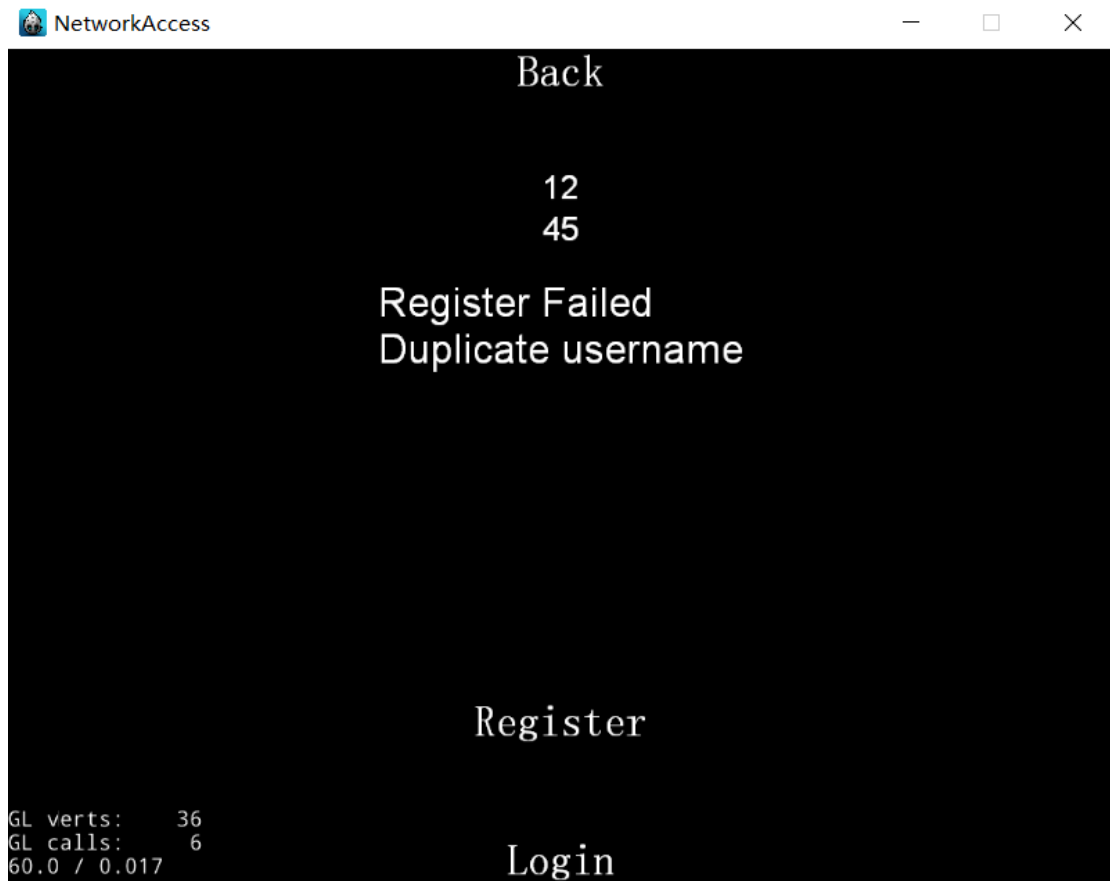
72 void ModifyUserScene::onHttpRequestCompleted(HttpClient *sender, HttpResponse *response) {
73     if (!response) {
74         return;
75     }
76     if (!response->isSucceed()) {
77         messageBox->setString(response->getErrorBuffer());
78         return;
79     }
80     auto responseData = response->getResponseData();
81     rapidjson::Document doc;
82     doc.Parse(responseData->data(), responseData->size());
83     if (doc["status"] == true) {
84         messageBox->setString("PUT OK");
85     }
86     else {
87         messageBox->setString(std::string("PUT Failed\n") + doc["msg"].GetString());
88     }
89 }

```

### 三、关键步骤截图

#### 1. 注册：





2. 登录:

Back

12

56

Login Failed  
Wrong username or password

Register

GL verts: 36  
GL calls: 6  
60.0 / 0.017

Login

Back

12

34

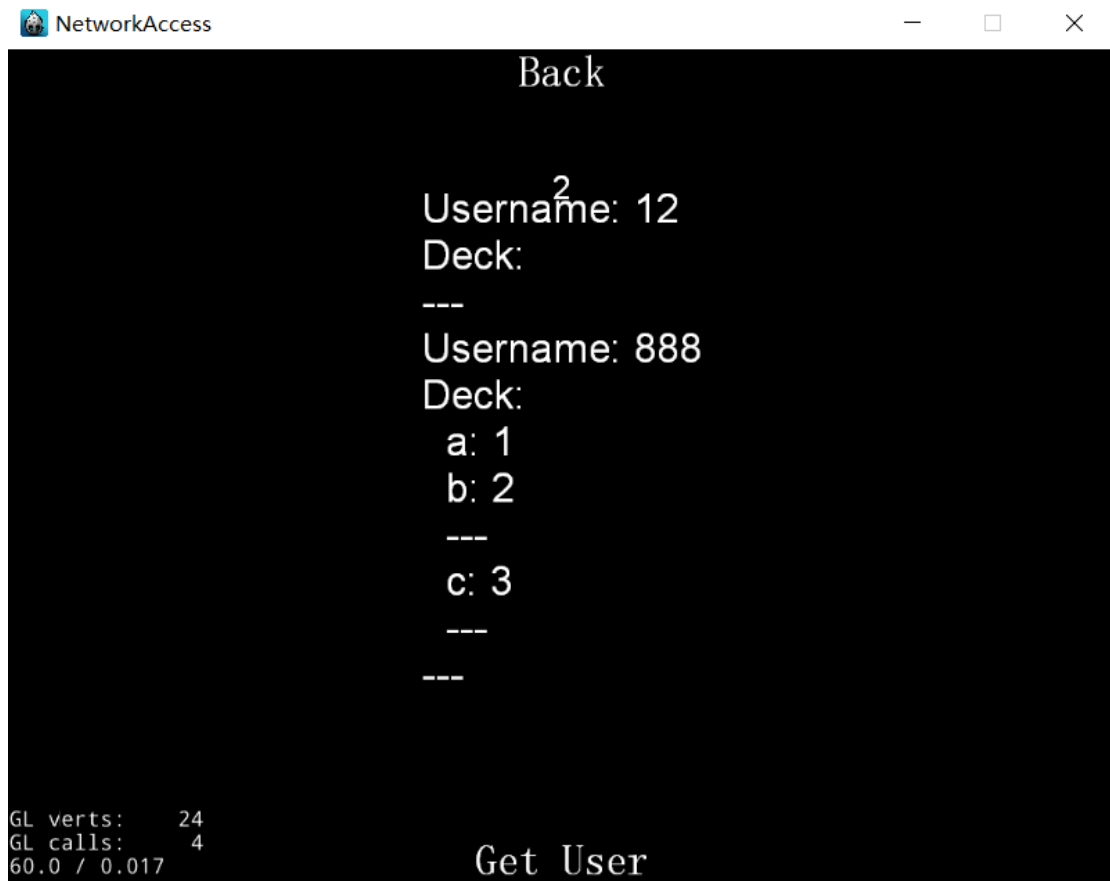
Login OK

Register

GL verts: 36  
GL calls: 6  
60.0 / 0.017

Login

### 3. 获取用户信息:



### 4. 修改个人信息:

Back

[{"www":2333}]

PUT OK

GL verts: 24  
GL calls: 4  
60.0 / 0.017

Post Deck

Back

Username: 12

Deck: 2

www: 2333

---

Username: 888

Deck:

a: 1

b: 2

---

c: 3

---

---

GL verts: 24  
GL calls: 4  
60.0 / 0.017

Get User



## 四、亮点与改进

在 LoginRegisterScene 中，在发送登录请求时，使用了

```
HttpClient::getInstance()->enableCookies(NULL);
```

这样可以启用 Cookie 机制，在客户端保持某些状态，实现客户端与服务端之间的会话跟踪。

```
79 void LoginRegisterScene::loginButtonCallback(cocos2d::Ref * pSender) {
80     // Your code here
81     HttpRequest* request = new HttpRequest();
82     request->setUrl("http://127.0.0.1:8000/auth");
83     request->setRequestType(HttpRequest::Type::POST);
84     request->setResponseCallback(CC_CALLBACK_2(LoginRegisterScene::onHttpRequestCompletedLogin, this));
85
86     std::string username = usernameInput->getStringValue();
87     std::string password = passwordInput->getStringValue();
88     std::string postData = "{\"username\":\"" + username + "\",\"password\":\"" + password + "\"}";
89     request->setRequestData(postData.c_str(), postData.length());
90     request->setTag("POST test");
91
92     cocos2d::network::HttpClient::getInstance()->enableCookies(NULL);
93     cocos2d::network::HttpClient::getInstance()->send(request);
94     request->release();
95 }
```

## 五、遇到的问题

"void cocos2d::network::HttpRequest::setResponseCallback(const cocos2d::network::ccHttpRequestCallback &): 无法将参数 1 从"std::Binder<std::\_Unforced,void (\_\_thiscall LoginRegisterScene::\*)(void),LoginRegisterScene \*,const std::\_Ph<1> &,const std::\_Ph<2> &>"转换为"const cocos2d::network::ccHttpRequestCallback &"

"void LoginRegisterScene::onHttpRequestCompletedLogin (cocos2d::network::HttpClient \*,cocos2d::network::HttpResponse \*)": "LoginRegisterScene"中没有找到重载的成员函数

遇到了找不到函数的问题。

解决：在 LoginRegisterScene.h 头文件里添加了 `using namespace cocos2d::network;`，这样

才能解析新添加的，有 HttpClient 和 HttpResponse 类型的

```
onHttpRequestCompletedLogin(HttpClient *sender, HttpResponse *response)
```

函数。其他头文件也是类似的处理。

## 六、思考与总结

1. 本次实验，主要使用了 `HttpRequest` 和 `HttpClient` 类来实现网络访问请求：构建 `HttpRequest` 实例，设置连接方法的类型、待连接的地址、回调函数，将该请求实例添加到 `HttpClient` 的当前实例的任务队列，之后释放连接。
2. 加深了对 json 格式的了解，学到了如何使用 `rapidjson::Document` 在 C++ 中解析 json 格式。
3. 值得一提的是，添加新函数时遇到了因为没有引入命名空间而导致无法生成的问题，这也体现了打好 C++ 基础的重要性。