

TÀI LIỆU PHÁT TRIỂN PLUGIN-X CHO FACEBOOK.

1. Giới thiệu

Sau khi đã thực hiện được cài đặt plugin-x của facebook theo các hàm chức năng cơ bản mà ta có thể sử dụng được thư viện protocols mà cocos2d-x đã cung cấp sẵn. Thì ở phần này sẽ nói về cách về phát triển thêm các chức năng cho facebook. Điều này khiến phải sửa đến thư viện protocols mà cocos2d-x đã cung cấp sẵn.

2. Các thư viện và bản cài đặt cần có.

2.1 Thư viện :

- Facebook sdk. Bản mới nhất là 3.8
- libFacebook.jar (thư viện facebook).
- libProtocol.jar (thư viện protocols).

2.2 Các phần mềm cần có.

- Eclipse.
- Android-Sdk.
- Android –ndk.
- Ant bản (1.93 trở lên).

3. Hướng dẫn viết.

Lưu ý : Các phần hướng dẫn về cài đặt , cấu hình plugin vẫn không thay đổi so với việc sử dụng thư viện protocols. Việc ở đây chỉ là chúng ta thêm các hàm cần thiết trong thư viện .

Ở phần này sẽ minh họa bằng việc thêm vào 1 function lắng nghe sự kiện của ở trong file ProtocolSocial.h và ProtocolSocial.cpp của thư viện libPluginProtocol.i

3.1 Bài toán đặt ra.

Giả sử ở trong phần plugin facebook , có chức năng lấy điểm của danh sách người dùng về . Phần trả về của facebook là một chuỗi json. Và mục đích là chuyển kết quả trả về chuỗi json đó từ bên java sang c++ .

Ở trong hàm class ProtocolSocial.h, chúng ta định nghĩa lại class listener cho mục đích riêng của ta .

Ở đây mình định nghĩa 1 ProtocolSocial.h cho riêng mình để dễ cho việc cài đặt plugin của mình hơn.

Đầu tiên mình sửa ở trong class SocialWapper.java mình thêm 1 biến

```
public static final int SHARERESULT_NETWORK = 4;
```

biến này mục đích là trả về kiểm tra mạng có bị lỗi hay không .

Tiếp đến định nghĩa lại class ProtocolSocial.h như sau

```

#include "PluginProtocol.h"
#include <map>
#include <string>

namespace cocos2d { namespace plugin {

typedef std::map<std::string, std::string> TSocialDeveloperInfo;
typedef std::map<std::string, std::string> TShareInfo;

typedef enum
{
    kResultSuccess = 0,
    kResultFail,
    kResultCancel,
    kResultTimeOut,
    kResultNetWork      // thêm mã code trả về network có bị lỗi hay không
} ResultCode;

// Định nghĩa lại class ResultListener , class này sẽ truyền kết quả từ
// java sang c++
class ResultListener
{
public:
    // trả lại kết quả của phương thức share
    virtual void onShareResult(ResultCode ret, const char* msg) = 0;
    // Trả lại kết quả của phương thức lấy điểm về hoặc post điểm lên .
    virtual void onScoreResult(ResultCode ret, const char* msg) = 0;
};

class ProtocolSocial : public PluginProtocol
{
public:
    ProtocolSocial();
    virtual ~ProtocolSocial();

    /**
    @brief config the social developer info
    @param devInfo This parameter is the info of developer,
           different plugin have different format
    @warning Must invoke this interface before other interfaces.
           And invoked only once.
    */
    void configDeveloperInfo(TSocialDeveloperInfo devInfo);

    /**
    @brief share information
    @param info The info of share, contains key:
           SharedText           The text need to share
           SharedImagePath      The full path of image file
    need to share (optinal)
    @warning For different plugin, the parameter should have other keys to
    share.
           Look at the manual of plugins.
    */
    void share(TShareInfo info);

    /**
    @breif set the result listener
    @param pListener The callback object for share result

```

```

@warning Must invoke this interface before share
*/
void setResultListener(ResultListener* pListener);

/**
@brief share result callback
*/
void onShareResult(ResultCode ret, const char* msg);
void onScoreResult(ResultCode ret, const char* msg);

protected:
    ResultListener* m_pListener;
};

}} // namespace cocos2d { namespace plugin {

#endif /* ----- #ifndef __CCX_PROTOCOL_SOCIAL_H__ ----- */

```

Trong file ProtocolSocial.cpp thì chỉ cần định nghĩa lại các hàm tương ứng với mình đã sửa trong file ProtocolSocial.h , xem phần source mình gửi kèm theo.

Tiếp đến , vì chúng ta định nghĩa lại class ResultListener vì vậy ở trong class FacebookManager.cpp chúng ta kế thừa lại class đó thì chúng ta phải định nghĩa lại 1 số hàm ở trong FacebookManager.cpp , xem trong source của file FacebookManager.cpp để hiểu hơn.