

TÀI LIỆU PLUGIN-X GOOGLE PLAY SERVICE.

1. Giới thiệu.

Đây là tài liệu hướng dẫn viết plugin-x cho google play service trên nền tảng android.

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

2.1 Thư viện.

- BaseGameUtils – thư viện mô tả chức năng cần thiết của google play service.
- Google-play-service : sdk của google play service.
- libPluginProtocols.jar – thư viện protocols

2.2 Bản cài đặt .

- Eclipse.
- Android-sdk.
- Android-ndk.
- Ant-1.9.3
- Cywin.

3. Hướng dẫn cài đặt và cách viết plugin.

3.1 Tạo app trên trang developer.

Để app của bạn test được, đầu tiên bạn cần phải cấu hình 1 app trên trang developer của google .

https://developers.google.com/games/services/android/quickstart#step_2_set_up_the_game_in_the_dev_console

Sau khi cài đặt thành công được app, chúng ta sẽ có được:

- APP_ID : đây là id của của project.
- Leaderboard ID : ID của bảng xếp hạng.
- Achievement ID : ID của bảng thành tựu .

Và chúng ta cần có các request của leaderboard và request achievement , id của 2 cái này thì do chúng ta tự đặt , có thể bất cứ 2 số nào đó đều được chấp nhận.

Vậy các id cần có cho 1 app của bạn khi sử dụng google play service bao gồm:

- App id.
- Leaderboard ID.
- Achievement ID.
- Request Achivement ID.
- Request Leaderboard ID.

3.2 Hướng dẫn viết các chức năng của google play service.

Các phương thức chức năng của google play service, mình đã viết ở trong class GameCenter.java ở trong thư mục BaseGameUtils.

Để viết được class này, mình có sử dụng 2 class GameHelper.java và GameHelperUtils.java, đây là 2 class hỗ trợ cho việc viết phần class GameCenter.java.

Để hiểu kỹ về nó, bạn vào trong source kèm theo để tìm hiểu.

Dưới đây mình sẽ mô tả về class GameCenter.java. Class này gồm 5 phương thức chính tương ứng với 8 chức năng chính của google play service.

- public static GameHelper getGameHelper().
- Public void signIn();
- Public void signOut();
- Public void isSignIn();
- public boolean showAchievements(int idAchievements);
- public void postAchievement (final String idName , int percentComplete);
- public void postAchievement (final String idName , int percentComplete);
- public void clearAllAchievements();
- public boolean showScores(final int idBoard);
- public void postScore(final String idName, final int score);

// Phương thức khởi tạo cho gamehelper .

// ở phương thức này, nếu bạn chưa khởi tạo cho google play service thì nó sẽ khởi tạo đồng thời sẽ login cho vào. Hàm này chỉ được gọi 1 lần, nếu bạn đã khởi tạo rồi mà cố khởi tạo nữa thì chương trình sẽ báo lỗi.

```
public static GameHelper getGameHelper() {
    if (mHelper == null) {
        mHelper = new GameHelper(mActivity, mRequestedClients);
        mHelper.enableDebugLog(true);
        mActivity.runOnUiThread(new Runnable() {

            @Override
            public void run() {
                mHelper.setup(new GameHelperListener() {
                    @Override
                    public void onSignInSucceeded() {
                        Log.d(TAG, " đang nhập thành công
");
                    }

                    @Override
                    public void onSignInFailed() {
                        Log.d(TAG, " đang nhập thất
bai");
                    }
                });
            }
        });
    }
}
```

```

    }
    ;
    return mHelper;
}

```

- Phương thức Sign In.

```

// Chức năng của nó là cho người dùng đăng nhập .
public void signIn() {

    mActivity.runOnUiThread(new Runnable() {

        @Override
        public void run() {
            getGameHelper().beginUserInitiatedSignIn();
        }

    });
}

```

- Phương thức SignOut.

// Chức năng của nó là hủy kết nối và đăng xuất tài khoản khỏi google play service.

```

public void signOut()
{
    if ( mHelper == null ) {
        return;
    }
    mActivity.runOnUiThread( new Runnable() {

        @Override
        public void run() {
            getGameHelper().signOut();
        }

    });
}
}

```

- Phương thức hiện thị bảng thành tựu.

// đối truyền vào là id của request achievements. Chức năng này yêu cầu người dùng phải đăng nhập .

```

public boolean showAchievements(int idAchievements){
    REQUEST_ACHIEVEMENTS = idAchievements;
    if ( !isSignedIn() ){
        return false;
    }
    mActivity.runOnUiThread( new Runnable() {

        @Override
        public void run() {
            // TODO Auto-generated method stub

```

```

        mActivity.startActivityForResult(Games.Achievements.getAchievementsIntent(getGameHelper().getApiClient()),
                                           REQUEST_ACHIEVEMENTS);
    }
});
return true;
}

```

- Phương thức post Achievement.

idName là id của AchievementID và percentComplete là phần trăm, hoàn thành của achievement đó.

```

public void postAchievement ( final String idName , int
percentComplete) {
    if ( !isSignedIn() || percentComplete < 100 ) {
        return;
    }
    mActivity.runOnUiThread( new Runnable() {

        @Override
        public void run() {

            Games.Achievements.unlock(getGameHelper().getApiClient(),
idName );

        }
    });
}

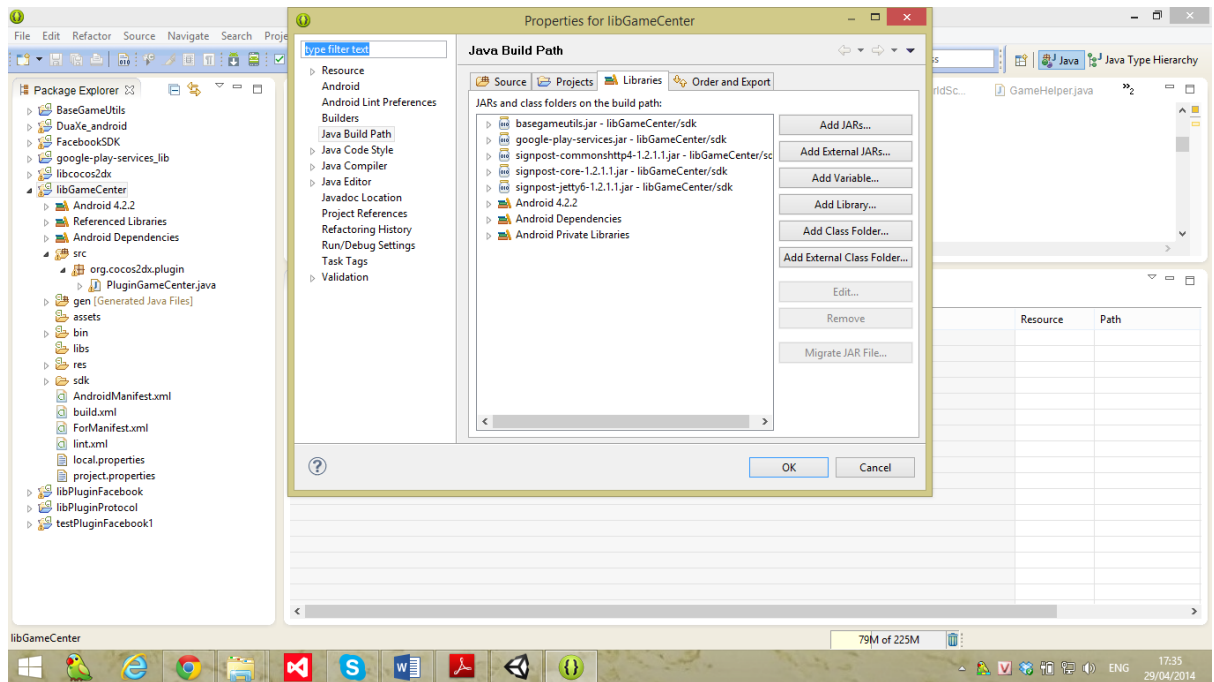
```

- Tương tự với postScore và showLeaderBoard.

Sau khi đã định nghĩa xong các phần chức năng, tiếp theo là chức ta sẽ viết được plugin. Về cơ bản plugin các chức năng mình viết rõ ở trong source.

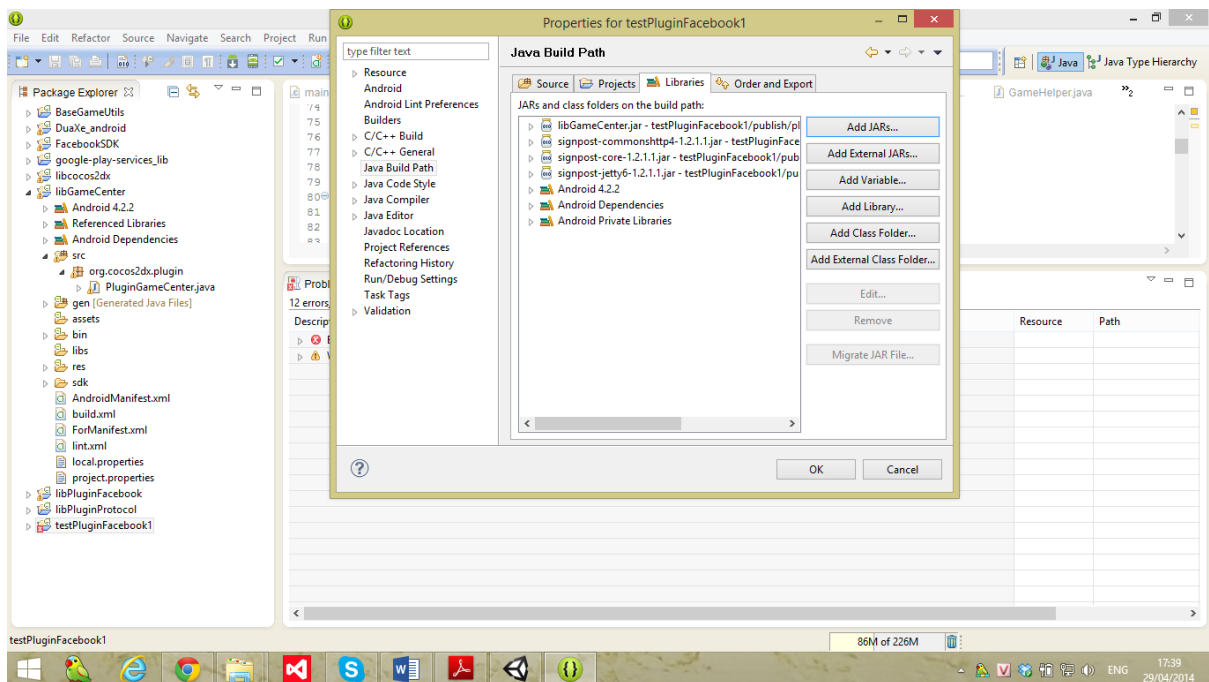
Phần cài đặt.

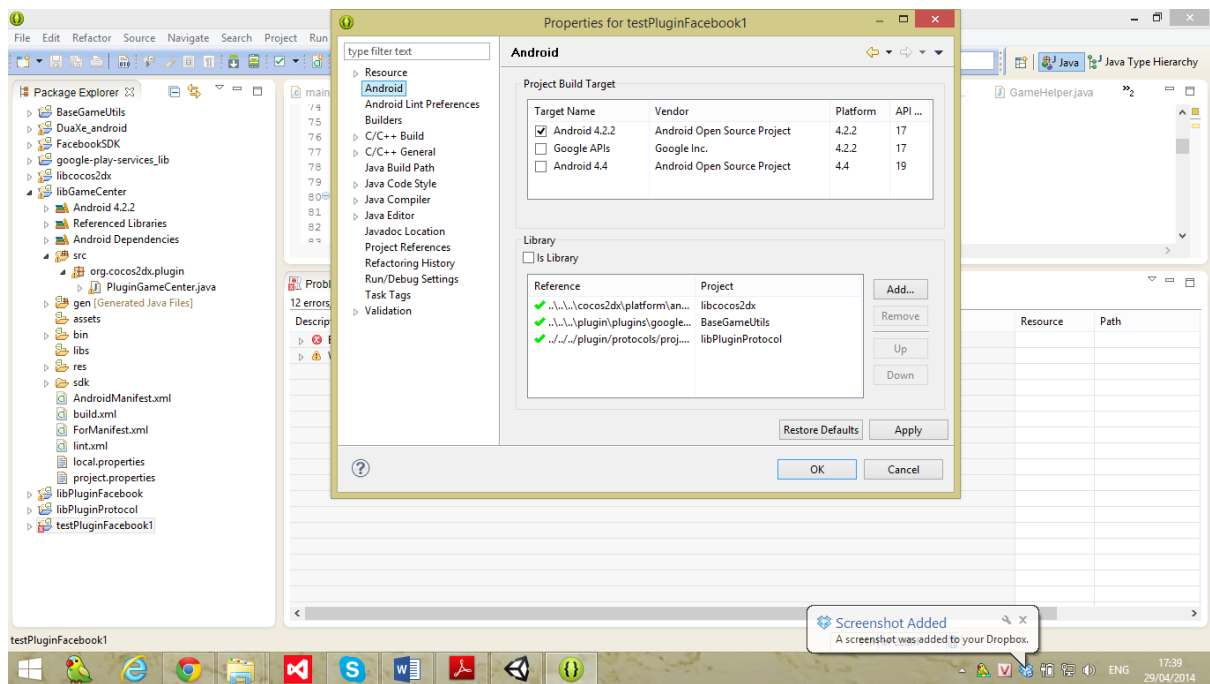
- Copy các thư viện google_play_service.jar , basegameutils.jar vào thư mục sdk của project libGameCenter và cấu hình vào project vào .



- Dịch project libGameCenter như thư viện facebook.

Sau khi thành công thì chúng ta có được thư mục publics . Sau đó chúng ta import thư viện vào plugin như của facebook vào project game của chúng ta.





Như vậy là đã hoàn thành cấu hình plugin google play service . Cách gọi và thực hiện các hàm giống như của facebook.