

# 开发框架介绍

海报墙系统由移动端和web端两部分组成。由于web端仅需实现登陆和上传海报两个功能，故我们没有使用开发框架。在移动端，我们使用了Android的开发框架。

## 【Android开发框架】

### 一、应用功能开发

Android应用程序的工程文件包含如下几部分：

- Java源代码部分(包含Activity)，都在src目录当中；
- R.java文件，这个文件是自动生成与维护的，开发者不需要修改，提供了Android对的资源全局索引；
- Android Library，这个是应用运行的Android库；
- assets目录，这个目录里面主要用与放置多媒体等一些文件；
- res目录，放置的是资源文件，其中的drawable包含的是图片文件，layout里面包含的是布局文件，values目录里面主要包含的是字符串(strings.xml)、颜色（colors.xml）以及数组（arrays.xml）资源；
- AndroidManifest.xml，是整个应用的配置文件，在这个文件中，需要声明所有用到的Activity、Service、Receiver等。

应用功能程序主要由四部分组成，即Activity、Broadcast Intent Receiver、Service、Content Provider。

(1) Activity：使用最频繁和基本的模块。每个Activity就是一个手机页面。当Activity运行结束后，系统将返回值。Android系统会自动记录所有跳转页面记录记录，并将以前的Activity压入系统堆栈，供用户使用。

(2) Broadcast Intent Receiver：Intent提供了不同Activity之间的跳转机制。我们可以使用Intent的实例，实现Activity之间的跳转。方式例举如下：

```
Intent in = new Intent(Activity A, Activity B);
startActivity(in);
```

(3) Service：用户可以startService(Intent service)启动一个Service，也可通过Context.bindService来绑定一个Service。

(4) Content Provider：提供了应用程序之间数据交换的机制。由于Android应用程序内部的数据都是私有的，故通过使用ContentProvider的抽象接口，程序才可以将自己的数据暴露给外部，供其使用。同时，通过Content Provider的机制，隐蔽了具体的数据存储实现，是一种良好的封装。标准的ContentProvider提供了基本的CRUD（Create,Read,Update,Delete）的接口，并且实现了权限机制，保护了数据交互的安全性。

## 二、数据存储

在Android，可供选择的存储方式共有五种，分别为：

(1) SharedPreferences：Android提供的一种配置文件读写方式。

(2) 文件存储：通过openFileInput、openFileOutput等系统提供的API来进行数据的读写访问。

(3) SQLite数据库存储方式：通过继承SQLiteOpenHelper类，通过此类的应用程序级别的实例来进行数据库操作。

(4) 内容提供者方式：通过调用其他应用程序的数据接口来实现数据的读写访问

(5) 网络方式：通过网络访问服务接口实现数据的读写服务。我们采用这种方式来向服务器请求海报图片。

## 三、网络访问

主要是Http访问技术的封装，通过：

- java.NET.\*
- android.net.\*

这两个包所含的类提供的接口进行web服务访问。

## 【框架结构】

分为表示层、业务层、数据层和通信层四部分