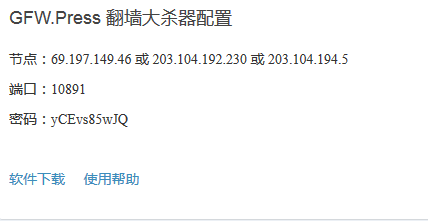
[](https://gfw.press/blog/?attachment_id=84)双击下载的安装包，按提示进行安装

安装过程通常只需点击几次 Next 按钮；

建议不要修改安装路径，如果修改，路径不要包含空格；

如果是 Windows 10 系统，请使用兼容模式进行安装；

如果安装失败，一般是安装包下载出错，请重新下载；

第一次运行，或者还没有进行过配置，会自动弹出配置窗口：

节点地址：填右图的节点中其中的一个；（节点地址可能会变，如果翻不了墙，去官网看一下节点地址是否改变）

节点端口：填右图的端口；

连接密码：填右图的密码；

本地端口就是代理服务器端口，默认是3128，通常不需要修改；

填写配置信息完成后，点击“确定”按钮，大杀器保存配置，开始运行，窗口缩小为右下角的一个绿色方块图标；如需打开配置窗口，双击图标即可；

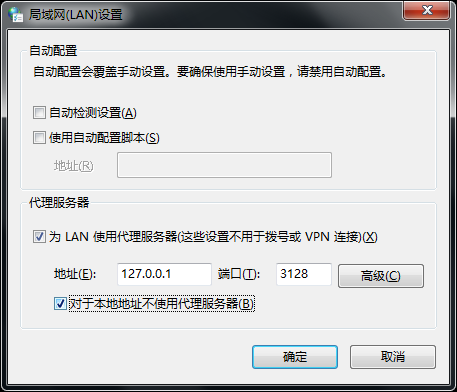
设置代理服务器

设置了代理服务器，浏览器就可以通过大杀器翻墙上网了；

点击Windows的“开始菜单》控制面板》Internet选项》连接》局域网设置”后，将出现配置窗口，按下图设置即可，端口对应就是大杀器配置的“本地端口”

地址： 127.0.0.1

端口：3128

[](https://gfw.press/blog/?attachment_id=85)

配置完成，点击“确定”按钮，然后打开IE、Chrome等浏览器，就可以翻墙访问谷歌之类被屏蔽的网站了。

注意：需要先翻墙

官网地址：https://gfw.press//

账号：[576804093@163.com](mailto:576804093@163.com)

密码：3060371LOVE

**导入上拉加载更多、下拉刷新控件（以WebView为例）：**

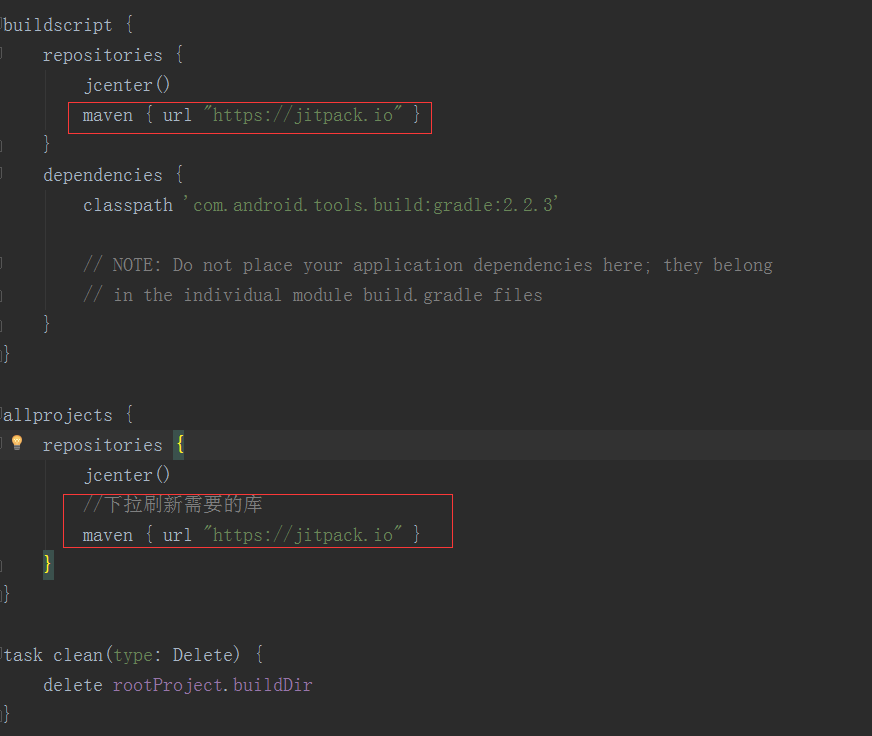
1. **在Module:app的build.gradle文件中的dependencies加入**

**compile 'com.github.Aspsine:SwipeToLoadLayout:1.0.3' 如**



1. **在Project:(项目名)的build.gradle文件中的repositories加入**

**maven { url "https://jitpack.io" } 如**



1. **在布局文件中引用刷新布局、头部和底部：**

<com.aspsine.swipetoloadlayout.SwipeToLoadLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/swipeToLoadLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1">

<com.example.administrator.signaturetest.PullToRefreshAndLoad.RefreshHeaderView

android:id="@id/swipe\_refresh\_header"

android:layout\_width="match\_parent"

android:gravity="center"

android:layout\_height="50dp" />

<WebView

android:id="@id/swipe\_target"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"/>

<com.example.administrator.signaturetest.PullToRefreshAndLoad.LoadMoreFooterView

android:id="@id/swipe\_load\_more\_footer"

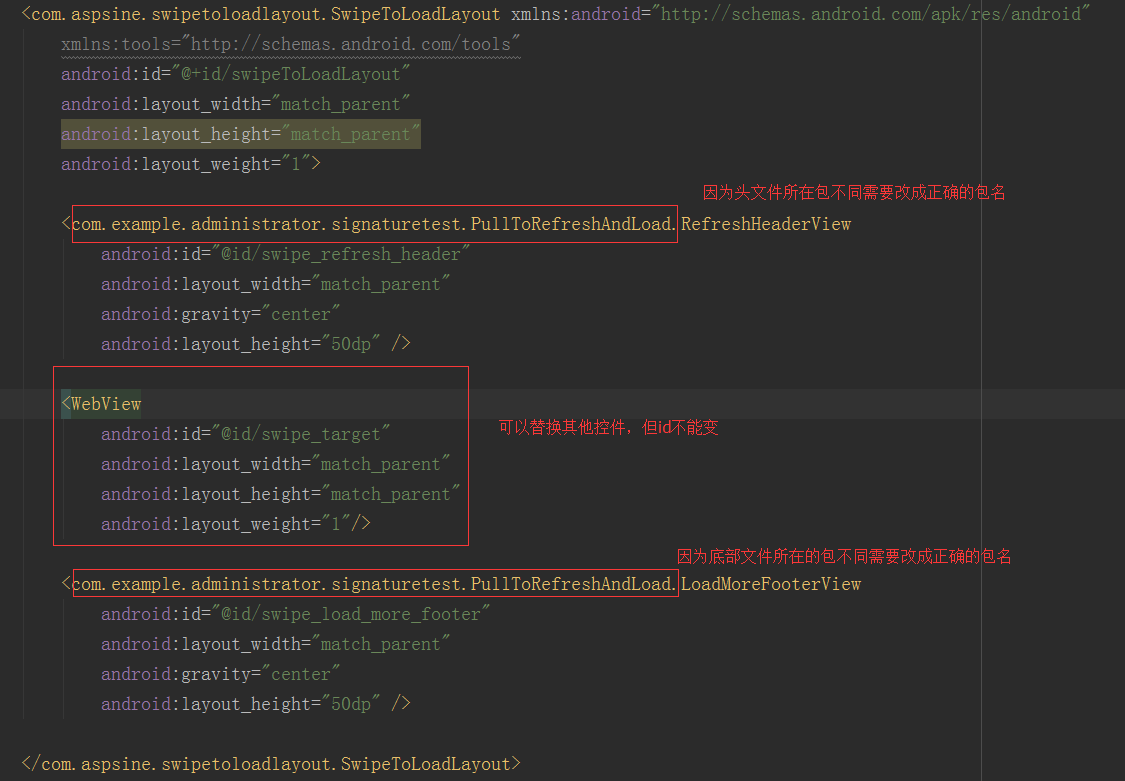
android:layout\_width="match\_parent"

android:gravity="center"

android:layout\_height="50dp" />

</com.aspsine.swipetoloadlayout.SwipeToLoadLayout>

其中四个控件ID不能改为其他



**4.创建头文件和底部文件：**

/\*\*

\* Created by Administrator on 2017/7/9.

\* 刷新头部

\*/

public class RefreshHeaderView extends TextView implements SwipeRefreshTrigger, SwipeTrigger {

public RefreshHeaderView(Context context) {

super(context);

}

public RefreshHeaderView(Context context, AttributeSet attrs) {

super(context, attrs);

}

@Override

public void onRefresh() {

// setText("REFRESHING");

setText("正在刷新中……");

}

@Override

public void onPrepare() {

setText("");

}

@Override

public void onMove(int yScrolled, boolean isComplete, boolean automatic) {

if (!isComplete) {

if (yScrolled >= getHeight()) {

// setText("RELEASE TO REFRESH");

setText("松开刷新");

} else {

// setText("SWIPE TO REFRESH");

setText("最后刷新时间: " + getLastUpdateTime());

}

} else {

// setText("REFRESH RETURNING");

setText("刷新成功");

}

}

@Override

public void onRelease() {

}

@Override

public void onComplete() {

setText("刷新成功");

}

@Override

public void onReset() {

setText("");

}

/\*\*

\* 获得系统的最新时间

\*

\* @return

\*/

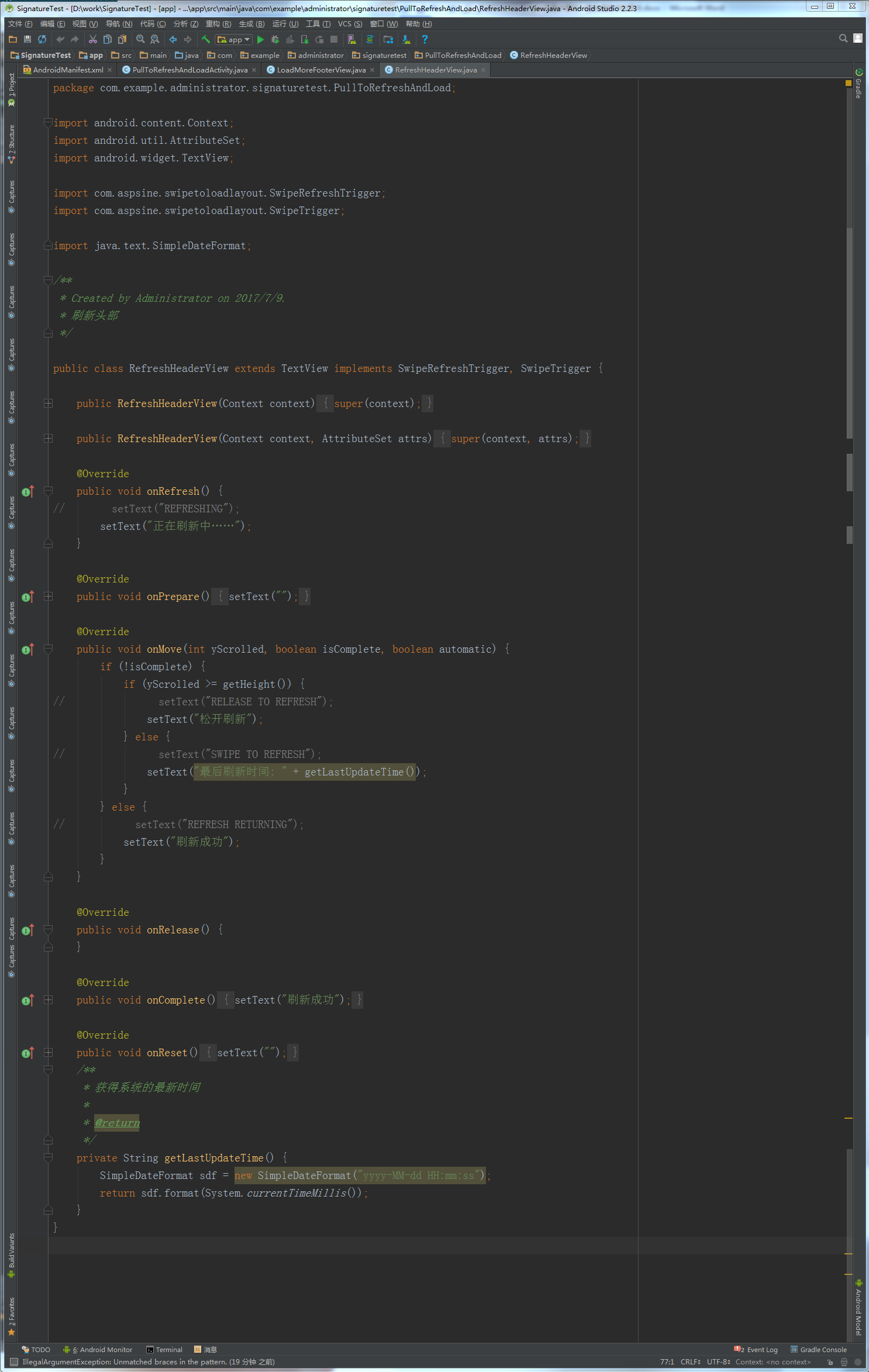
private String getLastUpdateTime() {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

return sdf.format(System.currentTimeMillis());

}

}



/\*\*

\* Created by Administrator on 2017/7/9.

\* 加载更多尾部

\*/

public class LoadMoreFooterView extends TextView implements SwipeTrigger, SwipeLoadMoreTrigger {

public LoadMoreFooterView(Context context) {

super(context);

}

public LoadMoreFooterView(Context context, AttributeSet attrs) {

super(context, attrs);

}

@Override

public void onLoadMore() {

// setText("LOADING MORE");

setText("加载更多……");

}

@Override

public void onPrepare() {

setText("");

}

@Override

public void onMove(int yScrolled, boolean isComplete, boolean automatic) {

if (!isComplete) {

if (yScrolled <= -getHeight()) {

// setText("RELEASE TO LOAD MORE");

setText("松开加载更多");

} else {

// setText("SWIPE TO LOAD MORE");

}

} else {

// setText("LOAD MORE RETURNING");

}

}

@Override

public void onRelease() {

// setText("LOADING MORE");

setText("加载更多……");

}

@Override

public void onComplete() {

// setText("COMPLETE");

}

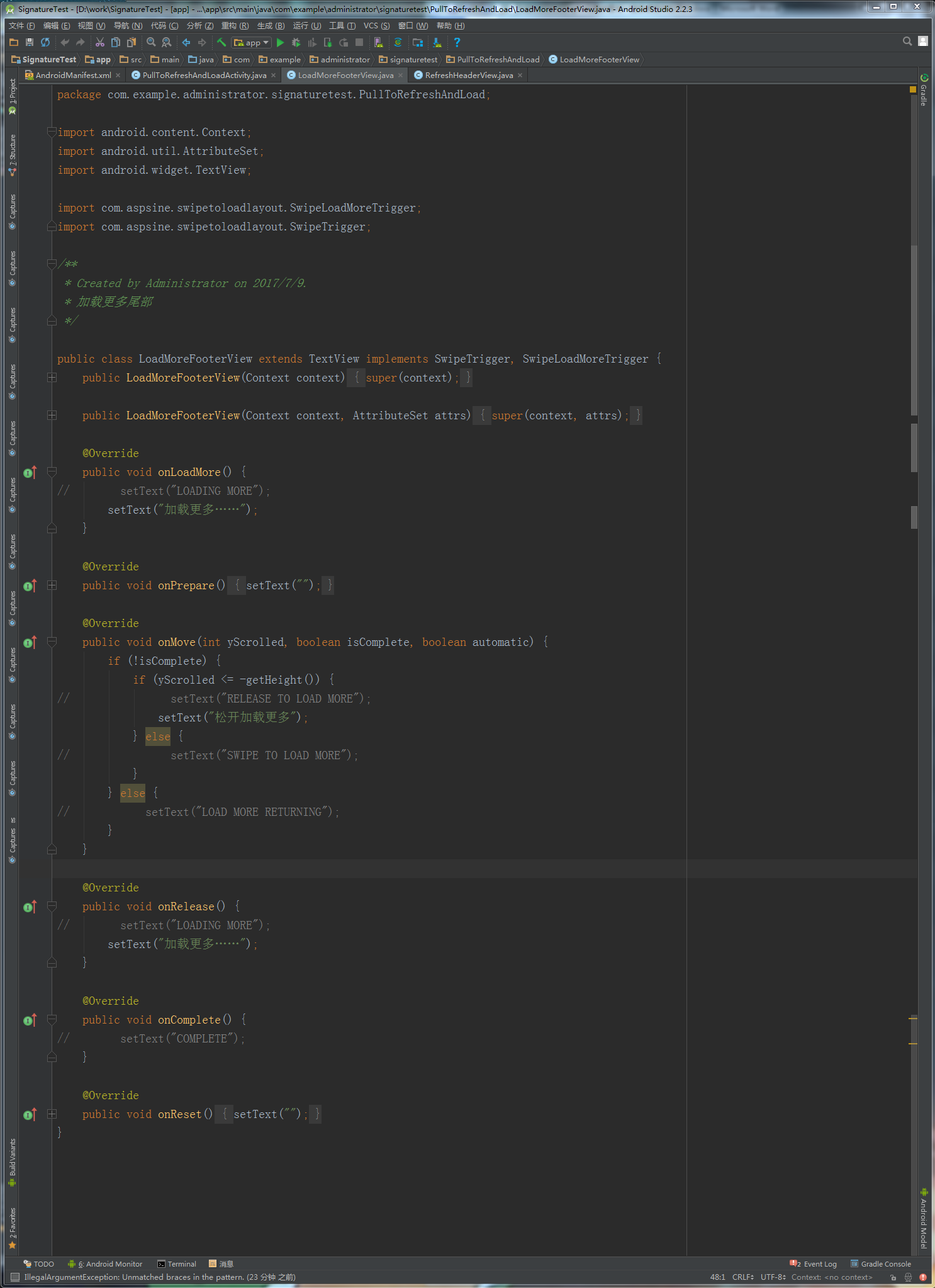
@Override

public void onReset() {

setText("");

}

}



**5.在Activity中引用:**

public class PullToRefreshAndLoadActivity extends AppCompatActivity implements OnRefreshListener, OnLoadMoreListener {

SwipeToLoadLayout swipeToLoadLayout;

WebView webView;

Handler handler;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

//取消标题栏

supportRequestWindowFeature(Window.FEATURE\_NO\_TITLE);

//设置布局

setContentView(R.layout.pull\_to\_refresh\_and\_load);

//获得刷新布局控件

swipeToLoadLayout = (SwipeToLoadLayout) findViewById(R.id.swipeToLoadLayout);

webView= (WebView) findViewById(R.id.swipe\_target);

//设置刷新监听

swipeToLoadLayout.setOnRefreshListener(this);

//设置加载更多监听

swipeToLoadLayout.setOnLoadMoreListener(this);

webView.setWebViewClient(new WebViewClient() {

@Override

public boolean shouldOverrideUrlLoading(WebView view, String url) {

view.loadUrl(url);

return true;

}

});

//WebChromeClient是辅助WebView处理Javascript的对话框，网站图标，网站title，加载进度等 :

webView.setWebChromeClient(new WebChromeClient() {

//处理alert弹出框

@Override

public boolean onJsAlert(WebView view, String url, String message, JsResult result) {

Toast.makeText(getApplicationContext(), message, Toast.LENGTH\_SHORT).show();

result.cancel();

return true;

}

});

webView.getSettings().setJavaScriptEnabled(true); //设置WebView属性,运行执行js脚本

//设置自适应屏幕，两者合用

webView.getSettings().setUseWideViewPort(true);//将图片调整到适合webview的大小

webView.getSettings().setLoadWithOverviewMode(true);// 缩放至屏幕的大小

webView.loadUrl("http://www.baidu.com");

// autoRefresh();自动刷新方法

//给Handler赋值

handler=new Handler();

}

//下拉刷新

@Override

public void onRefresh() {

swipeToLoadLayout.postDelayed(new Runnable() {

@Override

public void run() {

swipeToLoadLayout.setRefreshing(false);

// mAdapter.add("REFRESH:\n" + new Date());

handler.post(new Runnable(){//用handler转交UI处理操作

@Override public void run(){

//在这写操作webview的代码

//请不要在这里放耗时代码，否则会卡住UI线程

webView.loadUrl("http://www.qq.com/");

}

});

}

}, 2000);

}

//上拉加载更多

@Override

public void onLoadMore() {

swipeToLoadLayout.postDelayed(new Runnable() {

@Override

public void run() {

swipeToLoadLayout.setLoadingMore(false);

Toast.makeText(getApplicationContext(),"aa",Toast.LENGTH\_SHORT).show();

}

}, 2000);

}

//自动刷新

private void autoRefresh() {

swipeToLoadLayout.post(new Runnable() {

@Override

public void run() {

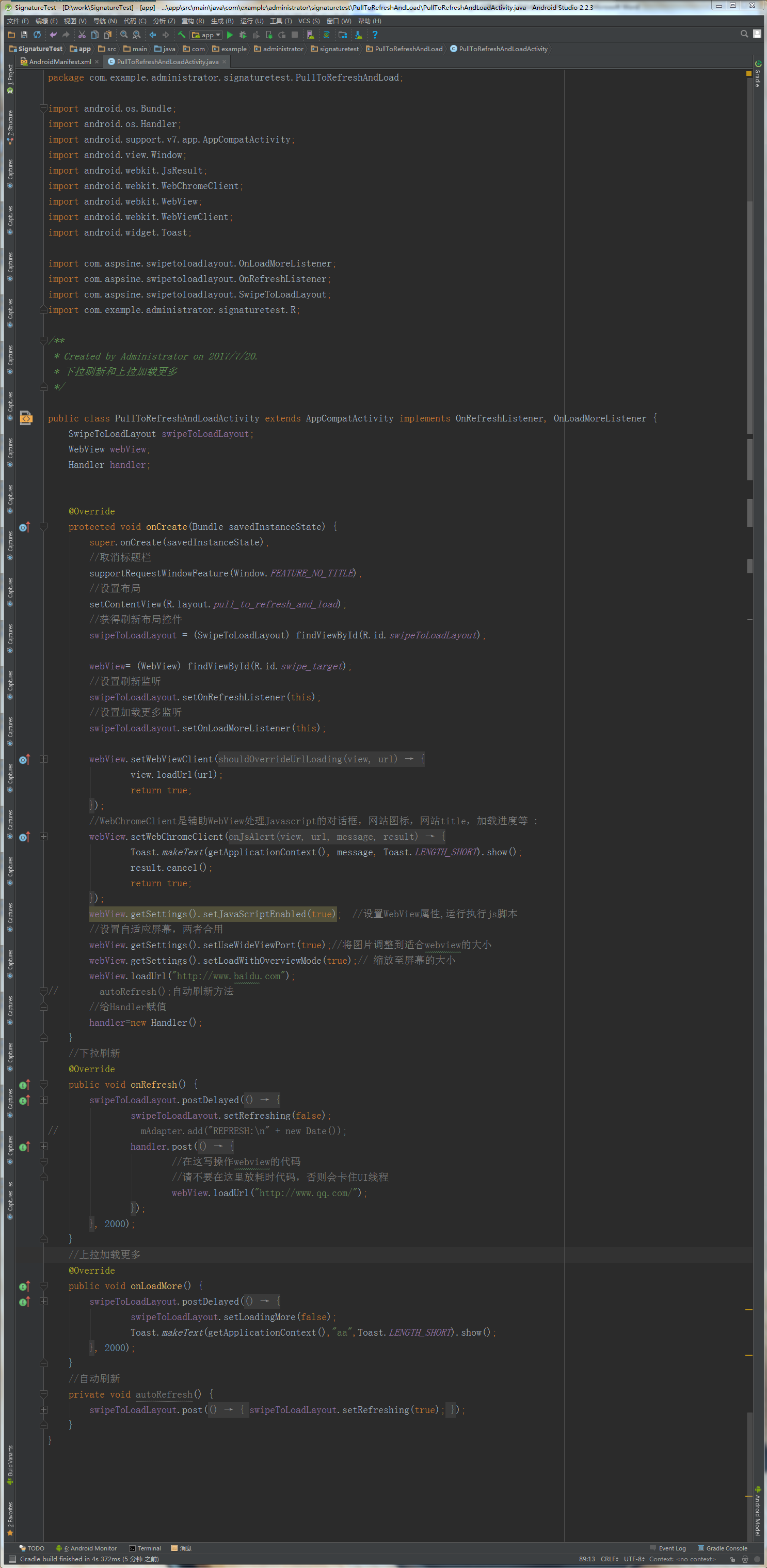
swipeToLoadLayout.setRefreshing(true);

}

});

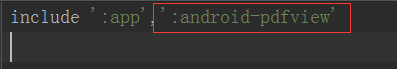
}

}



**导在线阅读PDF控件（方法一）：**

**1.在settings.gradle中导入':android-pdfview'，如：**

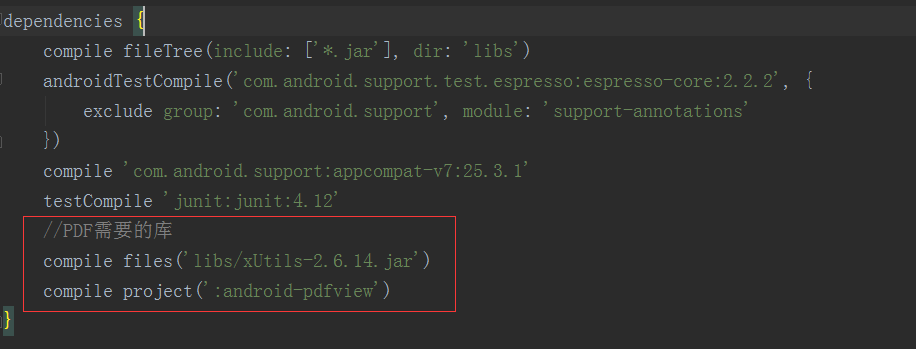


**2.往APP目录下的libs目录导入xUtils-2.6.14jar包**

**3.在build.gradle文件的dependencies中导入**

**compile files('libs/xUtils-2.6.14.jar')**

**compile project(':android-pdfview')**



**4.在需要的XML中引用：**

**<com.joanzapata.pdfview.PDFView**

**android:id="@+id/pdfview"**

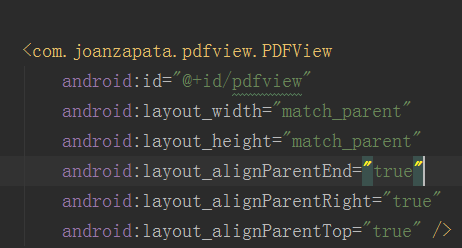
**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:layout\_alignParentEnd="true"**

**android:layout\_alignParentRight="true"**

**android:layout\_alignParentTop="true" />**



**5在需要的Activity中引用：**

public class MainActivity extends AppCompatActivity implements OnPageChangeListener {  
 public static final String *Pdf\_Url*="http://123.56.187.157:8888/hlts/upload/1500882446308\_1.pdf";  
 private int pageNumber = 1;  
 private PDFView pdfview;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 getSupportActionBar().hide();//隐藏掉整个ActionBar，包括下面的Tabs  
 setContentView(R.layout.*activity\_pdfview*);  
 pdfview = (PDFView) findViewById(R.id.*pdfview*);  
 downloadPdf();  
 }  
 private void downloadPdf(){  
 try {  
 com.lidroid.xutils.HttpUtils http = new com.lidroid.xutils.HttpUtils();  
 http.download(*Pdf\_Url*, Environment.*getExternalStorageDirectory*()+"/provisional.pdf", true, false, new RequestCallBack<File>() {  
 @Override  
 public void onStart() {}  
 @Override  
 public void onLoading(long total, long current, boolean isUploading) {}  
 @Override  
 public void onFailure(HttpException error, String msg) {}  
 @Override  
 public void onSuccess(ResponseInfo<File> responseInfo) {  
 File file = new File(Environment.*getExternalStorageDirectory*()+"/provisional.pdf");  
 readPdf(file);  
 }  
 });  
 } catch (Exception e) {  
 Log.*d*("error", e.toString());  
 }  
 }  
 private void readPdf(File file) {  
 pdfview.fromFile(file)  
 .defaultPage(pageNumber)  
 .load();  
 }  
 //这个下载网络上的pdf到sd卡上的代码 然后进行本地读取 在失去焦点的时候删除下载的文件  
// 删除代码  
 @Override  
 protected void onPause() {  
 super.onPause();  
 File file = new File(Environment.*getExternalStorageDirectory*()+"/provisional.pdf");  
 if (file.exists()) {  
 file.delete();  
 }  
 }  
 //继承接口后可以继承的方法  
 @Override  
 public void onPageChanged(int page, int pageCount) {  
 pageNumber = page;  
 }  
 @Override  
 public void onBackPressed() {  
 super.onBackPressed();  
 }  
  
}

**方法二：1.在build.gradle的dependencies中引用**

**compile 'com.lidong.pdf:android\_pdf:1.0.1'**

**2.在需要引用的XML中引用：**

<com.lidong.pdf.PDFView  
 android:id="@+id/pdfView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 />

**2.在Activity中的使用：**

public class Download\_the\_PDF\_and\_display extends AppCompatActivity implements OnPageChangeListener  
 , OnLoadCompleteListener, OnDrawListener {  
 private PDFView pdfView;  
 private ProgressDialog pd;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*pdf*);  
 pdfView = (PDFView) findViewById(R.id.*pdfView*);  
 displayFromFile1("http://123.56.187.157:8888/hlts/upload/1499049299321\_1.pdf", "1499049299321\_1.pdf");  
 //读取本地PDF  
// displayFromAssets("http://123.56.187.157:8888/hlts/upload/1499049299321\_1.pdf");  
 }  
 private void displayFromAssets(String assetFileName ) {  
 pdfView.fromAsset(assetFileName) //设置pdf文件地址  
 .defaultPage(1) //设置默认显示第1页  
 .onPageChange(this) //设置翻页监听  
 .onLoad(this) //设置加载监听  
 .onDraw(this) //绘图监听  
 .showMinimap(true) //pdf放大的时候，是否在屏幕的右上角生成小地图  
 .swipeVertical(true) //pdf文档翻页是否是垂直翻页，默认是左右滑动翻页  
 .enableSwipe(true) //是否允许翻页，默认是允许翻页  
 .load();  
 }  
 */\*\*  
 \* 获取打开网络的pdf文件  
 \*  
 \** ***@param*** *fileUrl  
 \** ***@param*** *fileName  
 \*/* private void displayFromFile1(String fileUrl, String fileName) {  
 showProgress();  
 pdfView.fileFromLocalStorage(this, this, this, fileUrl, fileName); //设置pdf文件地址  
  
 }  
  
 */\*\*  
 \* 翻页回调  
 \*  
 \** ***@param*** *page  
 \** ***@param*** *pageCount  
 \*/* @Override  
 public void onPageChanged(int page, int pageCount) {  
 /\*Toast.makeText(Download\_the\_PDF\_and\_display.this , "page= " + page +  
 " pageCount= " + pageCount , Toast.LENGTH\_SHORT).show();\*/  
 }  
  
 */\*\*  
 \* 加载完成回调  
 \*  
 \** ***@param*** *nbPages 总共的页数  
 \*/* @Override  
 public void loadComplete(int nbPages) {  
 Toast.*makeText*(Download\_the\_PDF\_and\_display.this, "加载完成" + nbPages, Toast.*LENGTH\_SHORT*).show();  
 hideProgress();  
 }  
  
 @Override  
 public void onLayerDrawn(Canvas canvas, float pageWidth, float pageHeight, int displayedPage) {  
 // Toast.makeText( MainActivity.this , "pageWidth= " + pageWidth + "  
 // pageHeight= " + pageHeight + " displayedPage=" + displayedPage , Toast.LENGTH\_SHORT).show();  
 }  
  
 */\*\*  
 \* 显示对话框  
 \*/* private void showProgress() {  
 pd = new ProgressDialog(Download\_the\_PDF\_and\_display.this);  
 pd.setProgressStyle(ProgressDialog.*STYLE\_SPINNER*);// 设置进度条风格，风格为圆形，旋转的  
 pd.setMessage("资源加载中,请稍候...");// 设置ProgressDialog提示信息  
 // 设置ProgressDialog 的进度条是否不明确 false 就是不设置为不明确  
 pd.setIndeterminate(false);  
 pd.setCancelable(false); // 设置ProgressDialog 是否可以按退回键取消  
 pd.show();  
 }  
  
 */\*\*  
 \* 关闭等待框  
 \*/* private void hideProgress() {  
 pd.dismiss();  
 }

**注意：两种方法只能用一种，否则包会冲突（推荐使用第二种），两种方式都需要网络权限以及SD卡读写权限**

<!--网络权限-->  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />