

## The Start Guide of SkyEpub SDK for Android in Java.

SkyEpub SDK is the most advanced and widely used epub sdk, which provides almost every feature to create a high level epub reader/viewer with minimum efforts.

This document is designed to show the easy way how to start and set your Android application with SkyEpub sdk for Android. and this start guide focuses on only making the basic skeleton of an epub reader for the very first time.

The source code (SkyTJ) of this document is also available at <http://www.skyepub.net/downloads>.

You'd better refer to SkyEpub Advanced Demo for Java or SkyEpub Advanced Demo for Kotlin to find the fully implemented epub reader demo using SkyEpub sdk.

### Prerequisite Software

Android Studio 4.0.x or above

SkyEpub SDK 8.5.x or above

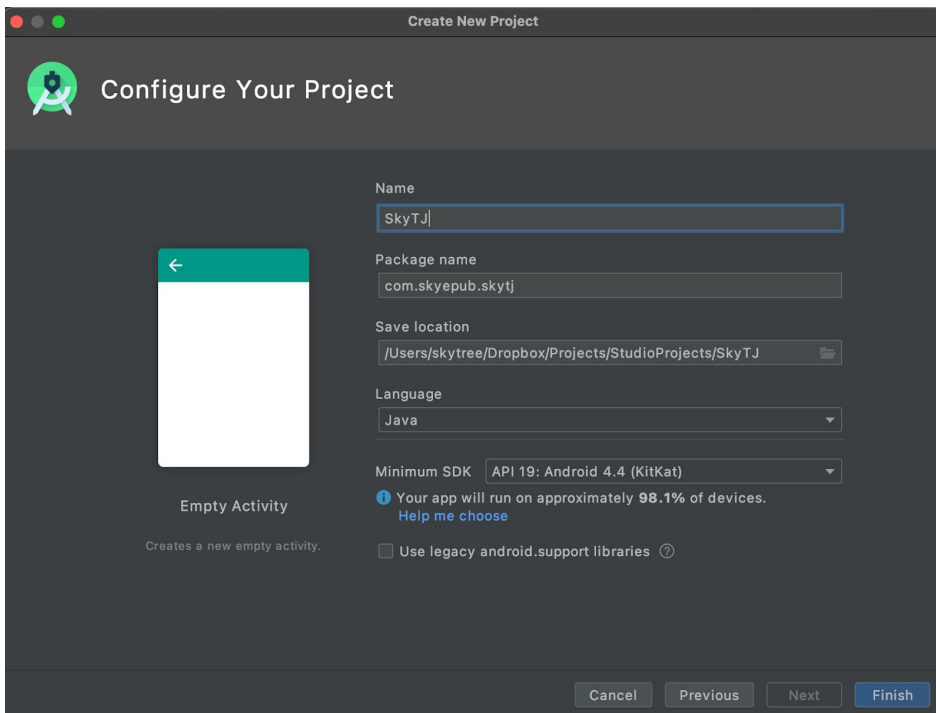
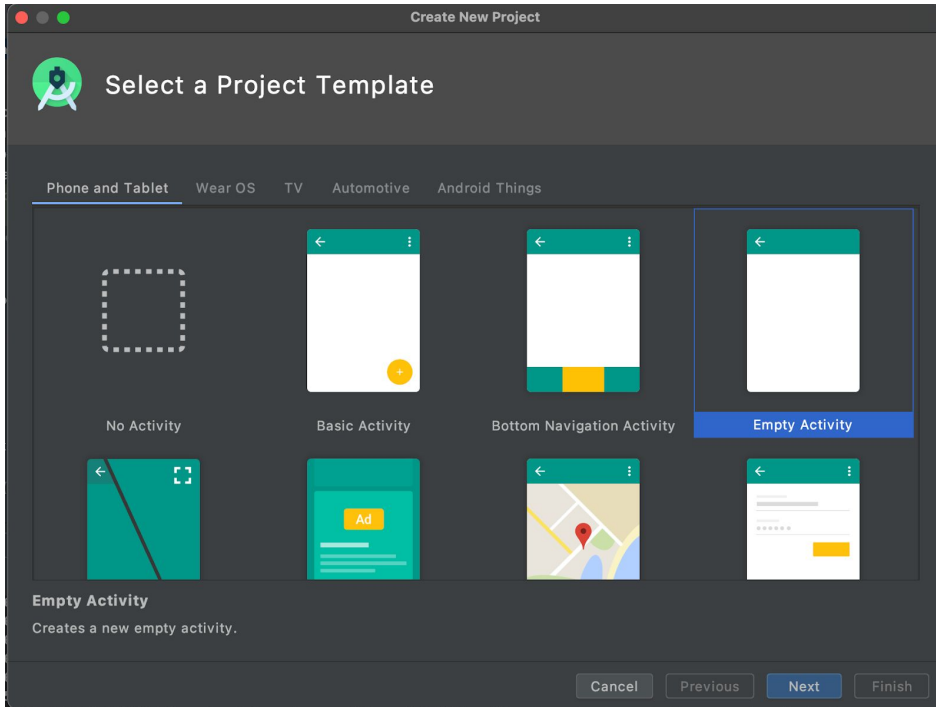
### Document History

1.1 23 Feb 2021

# Create Project

**New > New Project > Empty Project**  
and set project name as SkyTJ

2.



## AndroidManifest Setting for Permissions

a. Add permissions like below to AndroidManifest.xml.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

b. `android:requestLegacyExternalStorage="true"` is needed to be added to `<application>`.

```
<application
..
    android:requestLegacyExternalStorage="true"
```

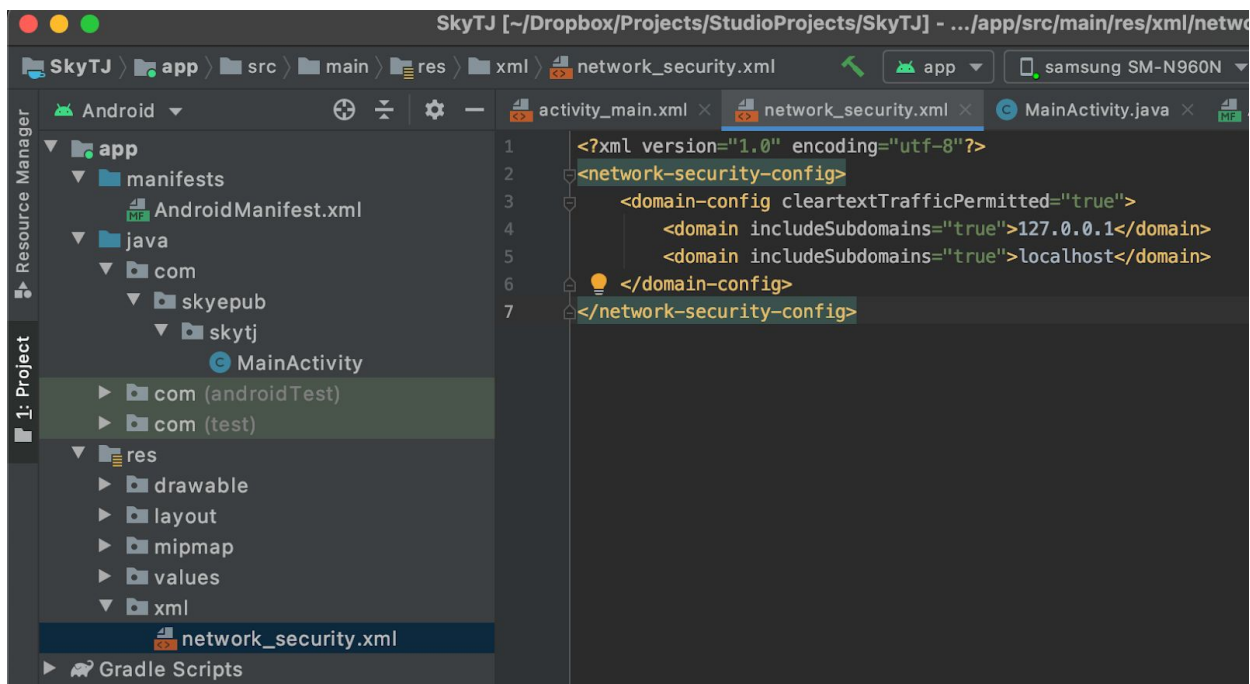
## Allow the Access to 'localhost'

Select "Android" in Project Pane by dropdown the menu.  
Make xml folder under app > res ( click right button on res folder and select New > Directory )

make **app > res > xml > network\_security.xml** file.

The content of this file is like the following.

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="true">127.0.0.1</domain>
    <domain includeSubdomains="true">localhost</domain>
  </domain-config>
</network-security-config>
```



Add networkSecurityConfig property to Application element in AndroidManifest.xml

```
<application
  android:requestLegacyExternalStorage="true"
  android:networkSecurityConfig="@xml/network_security"
```

## Import skyepub.jar and other jar into this project.

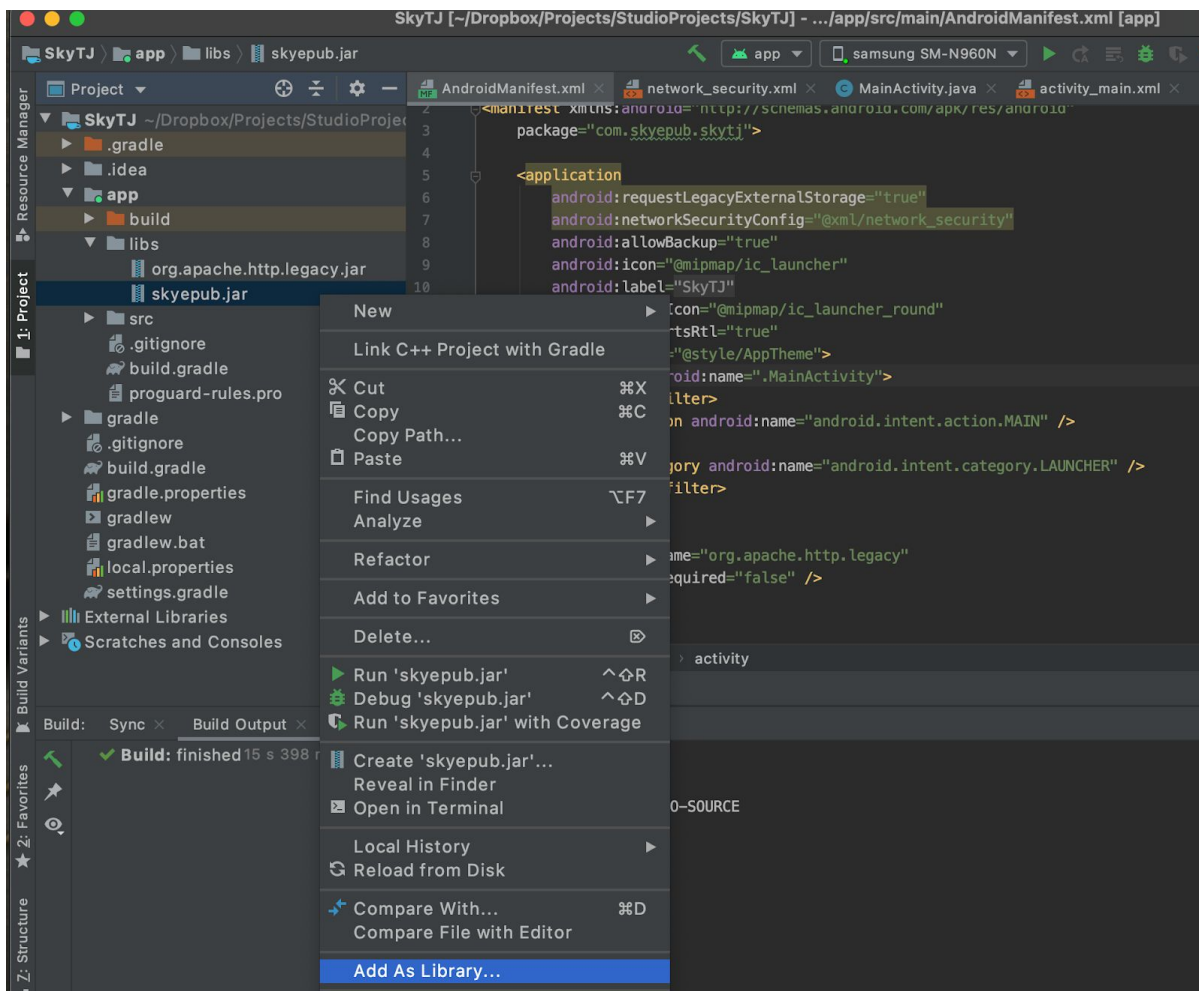
a.

### skyepub.jar

: the latest version is always available from <http://www.skyepub.net/downloads>

### org.apache.http.legacy.jar

: <http://hc.apache.org/downloads.cgi>



Drag these two files into **Project > Project > SkyTJ > app > libs**.

b. click the right button on skyepub.jar and choose 'Add As Library'.

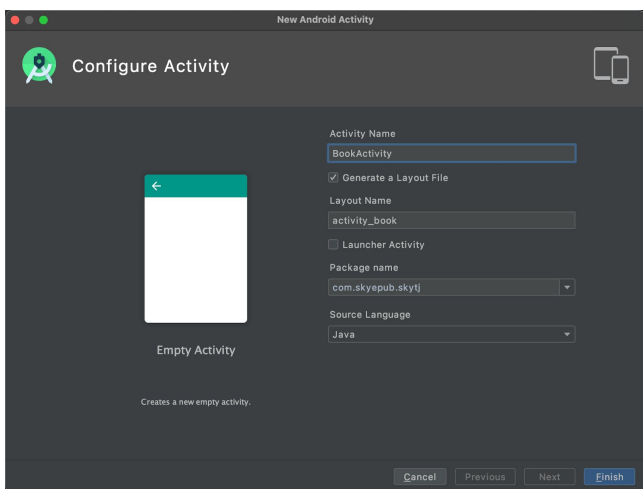
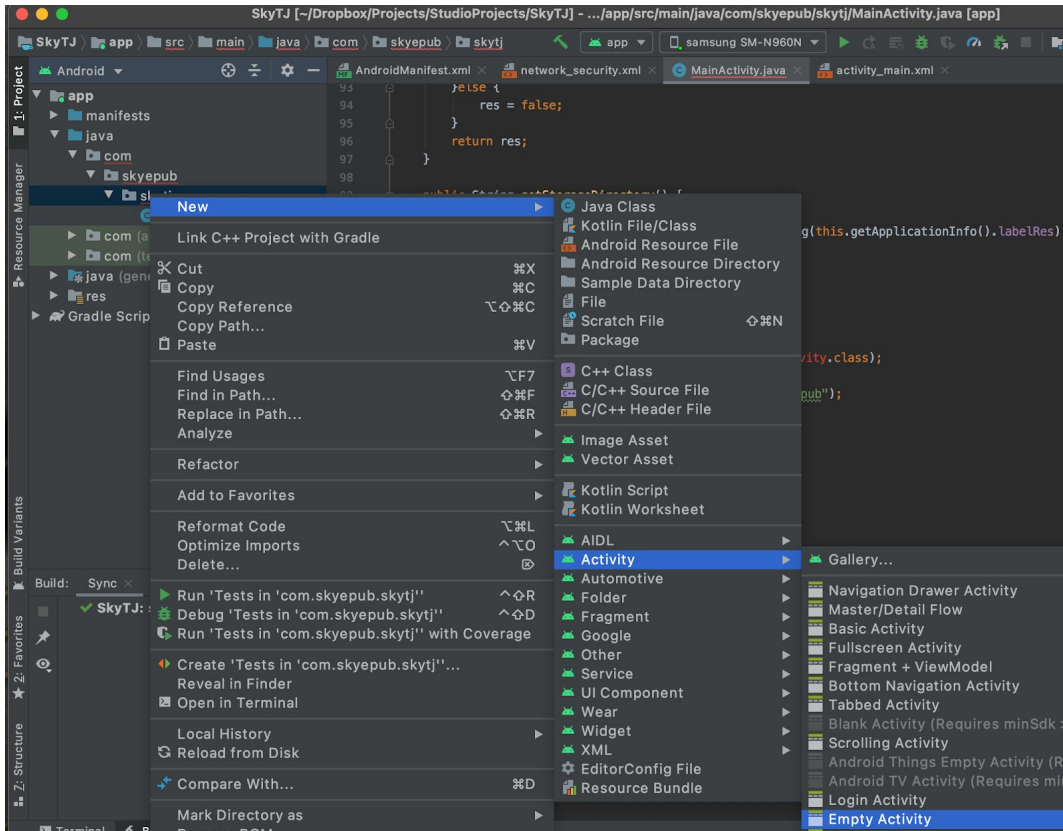
c. Add org.apache.http.legacy to AndroidManifest.xml

```
</activity>
  <uses-library
    android:name="org.apache.http.legacy"
    android:required="false" />
</application>
```

## Create BookActivity

Create new Activity by **New > Activity > Empty Activity**

Set the name of the new activity as 'BookActivity'.



## Make one button to open book in MainActivity

Add a button by modifying **res > layout > activity\_main.xml**.

set id to 'openBookButton'

and set text to 'Open Book'

and set onClick to 'onOpenBookPressed'

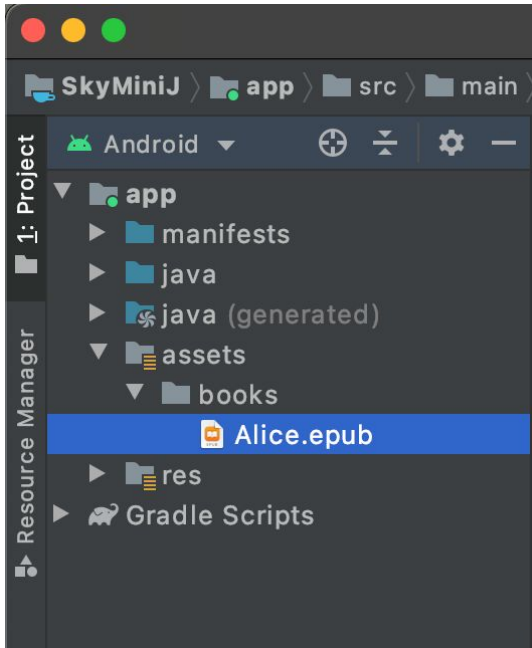
the modified xml looks like the below.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<Button
    android:id="@+id/openBookButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="60dp"
    android:layout_marginTop="40dp"
    android:onClick="onOpenBookPressed"
    android:text="Open Book"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Include sample epub in this project.

Make folder in **Project > Android > app > assets > books** and drag 'Alice.epub' into this folder.





## Make some code in MainActivity

When MainActivity starts, it will create folder 'books' under the document folder of the app. and it will copy Alice.epub from **assets > books > Alice.epub** to the folder above. These codes are just to copy one sample file to use. and these have nothing to do with SkyEpub sdk itself.

MainActivity.java is now like the below.

```
package com.skyepub.skytj;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;

public class MainActivity extends AppCompatActivity {
    final String TAG = "SKYEPUB";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        check();
        setup();
    }

    void check() {
        String path = this.getStorageDirectory() + "/books/"+"Alice.epub";
        File file = new File(path);
        if (file.exists()) {
            Log.w(TAG,"File installed");
        }else {
            Log.w(TAG,"File not installed");
        }
    }

    void setup() {
        try {
            if (isSetup()) return;
            this.makeDirectory("books");
            this.copyBookFromAssetsToDevice("Alice.epub");

            SharedPreferences pref = getSharedPreferences("SkyMiniJ",0);
            SharedPreferences.Editor edit = pref.edit();
            edit.putBoolean("isSetup", true);
            edit.commit();
        }
    }
}
```

```

    }catch(Exception e) {

    }
}

private boolean isSetup() {
    SharedPreferences pref = getSharedPreferences("SkyMiniJ",0);
    return pref.getBoolean("isSetup",false);
}

public void copyBookFromAssetsToDevice(String fileName) {
    try
    {
        String path = this.getStorageDirectory() + "/books/"+fileName;
        File file = new File(path);
        if (file.exists()) return;
        Context context;
        InputStream localInputStream = getAssets().open("books/"+fileName);
        FileOutputStream localFileOutputStream = new
        FileOutputStream(this.getStorageDirectory() + "/books/"+fileName);

        byte[] arrayOfByte = new byte[1024];
        int offset;
        while ((offset = localInputStream.read(arrayOfByte))>0)
        {
            localFileOutputStream.write(arrayOfByte, 0, offset);
        }
        localFileOutputStream.close();
        localInputStream.close();
        Log.d(TAG, fileName+" copied to phone");
    }
    catch (IOException localIOException)
    {
        localIOException.printStackTrace();
        Log.d(TAG, "failed to copy");
        return;
    }
}

public boolean makeDirectory(String dirName) {
    boolean res;
    String filePath = new String(this.getStorageDirectory() + "/" +dirName);
    File file = new File(filePath);
    if (!file.exists()) {
        res = file.mkdirs();
    }else {
        res = false;
    }
    return res;
}

public String getStorageDirectory() {
    String res = "";
    res = getFilesDir().getAbsolutePath()+"/"+getString(this.getApplicationInfo().labelRes);
    return res;
}

```

```
public void onOpenBookPressed(View view) {  
    Intent intent;  
    intent = new Intent(this, BookActivity.class);  
    intent.putExtra("BOOKCODE", 0);  
    intent.putExtra("BOOKNAME", "Alice.epub");  
    startActivity(intent);  
}  
}
```

onOpenBookPressed passes over bookCode, bookName to BookActivity when openButton is clicked.

## Add a ConstraintLayout named 'skyepubView' to BookActivity

and add the properties to cover the entire screen. skyepub sdk will be inserted into this 'skyepubView'.

the activity\_book.xml to which new constraintLayout is added is like the following.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".BookActivity">

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/skyepubView"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:background="#871F1F"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

## Complete the code to open epub in BookActivity

```
package com.skyepub.skytj;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.HapticFeedbackConstants;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.TextView;

import com.skytree.epub.Book;
import com.skytree.epub.KeyListener;
import com.skytree.epub.PageTransition;
import com.skytree.epub.PagingMode;
import com.skytree.epub.ReflowableControl;
import com.skytree.epub.SkyProvider;

import java.io.File;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;

public class BookActivity extends AppCompatActivity {
    ConstraintLayout skyepubView;
    ReflowableControl rv;
    String fileName;
    int bookCode;
    double pagePositionInBook;
    final String TAG = "SKYEPUB";

    public String getStorageDirectory() {
        String res = "";
        // All book related data will be stored /data/data/com...../files/appName/
        res = getFilesDir().getAbsolutePath()+"/"+getString(this.getApplicationInfo().labelRes);
        return res;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_book);
    }
}
```

```

this.makeFullscreen(this);
skyepubView = (ConstraintLayout)this.findViewById(R.id.skyepubView);
this.makeBookViewer();
}

public void makeBookViewer() {
    Bundle bundle = getIntent().getExtras();
    fileName = bundle.getString("BOOKNAME");
    bookCode = bundle.getInt("BOOKCODE");
    pagePositionInBook = 0;

    rv = new ReflowableControl(this); // in case that device supports
    transparent webkit, the background image under the content can be shown. in some devices,
    content may be overlapped.
    rv.setId(7777);

    // set the bookCode to identify the book file.
    rv.bookCode = this.bookCode;

    // Be sure that the file exists before setting.
    String bookPath = this.getStorageDirectory() + "/books/" + fileName;
    rv.setBookPath(bookPath);
    // if true, double pages will be displayed on landscape mode.
    rv.setDoublePagedForLandscape(true);
    // set the initial font style for book.
    rv.setFont("Book Fonts", this.getRealFontSize(3));
    // set the initial line space for book.
    rv.setLineSpacing(this.getRealLineSpace(2)); // the value is supposed to be percent(%).
    // set the horizontal gap(margin) on both left and right side of each page.
    rv.setHorizontalGapRatio(0.30);
    // set the vertical gap(margin) on both top and bottom side of each page.
    rv.setVerticalGapRatio(0.20);
    // rv.setMarginTopRatio(0.7);

    // SkyProvider is the default ContentProvider which is presented with SDK.
    // SkyProvider can read the content of epub file without unzipping.
    // SkyProvider is also fully integrated with SkyDRM solution.
    SkyProvider skyProvider = new SkyProvider();
    rv.setContentProvider(skyProvider);

    // set the start position to open the book.
    rv.setStartPositionInBook(pagePositionInBook);
    // if true, globalPagination will be activated.
    // this enables the calculation of page number based on entire book ,not on each chapter.
    // this globalPagination consumes huge computing power.
    // AVOID GLOBAL PAGINATION FOR LOW SPEC DEVICES.
    rv.setGlobalPagination(false);
    // set the navigation area on both left and right side to go to the previous or next page when
    the area is clicked.
    rv.setNavigationAreaWidthRatio(0.2f); // both left and right side.
    // set the navigation area enabled
    rv.setNavigationAreaEnabled(true);

    // set the device locked to prevent Rotation.
    rv.setRotationLocked(true);

```

```

    // If you want to get the license key for commercial use, please email us
    (skytree21@gmail.com).
    // Without the license key, watermark message will be shown in background.
    rv.setLicenseKey("0000-0000-0000-0000");

    // set PageTransition Effect
    int transitionType = bundle.getInt("transitionType");
    if (transitionType == 0) {
        rv.setPageTransition(PageTransition.None);
    } else if (transitionType == 1) {
        rv.setPageTransition(PageTransition.Slide);
    } else if (transitionType == 2) {
        rv.setPageTransition(PageTransition.Curl);
    }

    rv.setSystemSelectionEnabled(true);
    rv.setForegroundColor(Color.BLACK);
    rv.keepBackgroundColor(true);

    rv.setLayoutParams(new
    ConstraintLayout.LayoutParams(ConstraintLayout.LayoutParams.MATCH_PARENT,
    ConstraintLayout.LayoutParams.MATCH_PARENT));
    skyepubView.addView(rv);
}

int getRealFontSize(int fontSizeIndex) {
    int rs = 0;
    switch (fontSizeIndex) {
        case 0:
            rs = 24;
            break;
        case 1:
            rs = 27;
            break;
        case 2:
            rs = 30;
            break;
        case 3:
            rs = 34;
            break;
        case 4:
            rs = 37;
            break;
        default:
            rs = 27;
    }
    rs = (int) ((double) rs * 0.75f);

    return rs;
}

public int getRealLineSpace(int lineSpaceIndex) {
    int rs = -1;
    if (lineSpaceIndex == 0) {
        rs = 125;
    } else if (lineSpaceIndex == 1) {

```

```

        rs = 150;
    } else if (lineSpaceIndex == 2) {
        rs = 165;
    } else if (lineSpaceIndex == 3) {
        rs = 180;
    } else if (lineSpaceIndex == 4) {
        rs = 200;
    } else {
        rs = 150;
    }
    return rs;
}

@SuppressLint("InlinedApi")
public static void makeFullscreen(Activity activity) {
    activity.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    if (Build.VERSION.SDK_INT >= 19) {
        activity.getWindow().getDecorView().setSystemUiVisibility(
            View.SYSTEM_UI_FLAG_IMMERSIVE
                | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION
                | View.SYSTEM_UI_FLAG_FULLSCREEN
                | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY
                | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
                | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
                | View.SYSTEM_UI_FLAG_LAYOUT_STABLE
        );
    } else if (Build.VERSION.SDK_INT >= 11) {
        activity.getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LOW_PR
        OFFILE);
    }
}
}
}

```