Demo54-Service实现后台下载任务

1. app模块下添加OkHttp的依赖

implementation **'com.squareup.okhttp3:okhttp:3.4.1'**

1. 创建回调接口 DownloadListener.java

public interface DownloadListener {

void onProgress(int progress);

void onSuccess();

void onFailed();

void onPaused();

void onCanceled();

}

1. AsyncTask的实现类

import android.os.AsyncTask;

import android.os.Environment;

import java.io.File;

import java.io.IOException;

import java.io.InputStream;

import java.io.RandomAccessFile;

import okhttp3.OkHttpClient;

import okhttp3.Request;

import okhttp3.Response;

public class DownloadTask extends AsyncTask<String, Integer, Integer> {

public static final int TYPE\_SUCCESS = 0;

public static final int TYPE\_FAILED = 1;

public static final int TYPE\_PAUSED = 2;

public static final int TYPE\_CANCELED = 3;

private DownloadListener listener;

private boolean isCanceled = false;

private boolean isPaused = false;

private int lastProgress;

public DownloadTask(DownloadListener listener) {

this.listener = listener;

}

@Override

protected Integer doInBackground(String... params) {

InputStream is = null;

RandomAccessFile savedFile = null;

File file = null;

try {

long downloadedLength = 0; // 记录已下载的文件长度

String downloadUrl = params[0];

String fileName = downloadUrl.substring(downloadUrl.lastIndexOf("/"));

String directory = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_DOWNLOADS).getPath();

file = new File(directory + fileName);

if (file.exists()) {

downloadedLength = file.length();

}

long contentLength = getContentLength(downloadUrl);

if (contentLength == 0) {

return TYPE\_FAILED;

} else if (contentLength == downloadedLength) {

// 已下载字节和文件总字节相等，说明已经下载完成了

return TYPE\_SUCCESS;

}

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()

// 断点下载，指定从哪个字节开始下载

.addHeader("RANGE", "bytes=" + downloadedLength + "-")

.url(downloadUrl)

.build();

Response response = client.newCall(request).execute();

if (response != null) {

is = response.body().byteStream();

savedFile = new RandomAccessFile(file, "rw");

savedFile.seek(downloadedLength); // 跳过已下载的字节

byte[] b = new byte[1024];

int total = 0;

int len;

while ((len = is.read(b)) != -1) {

if (isCanceled) {

return TYPE\_CANCELED;

} else if(isPaused) {

return TYPE\_PAUSED;

} else {

total += len;

savedFile.write(b, 0, len);

// 计算已下载的百分比

int progress = (int) ((total + downloadedLength) \* 100 / contentLength);

publishProgress(progress);

}

}

response.body().close();

return TYPE\_SUCCESS;

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (is != null) {

is.close();

}

if (savedFile != null) {

savedFile.close();

}

if (isCanceled && file != null) {

file.delete();

}

} catch (Exception e) {

e.printStackTrace();

}

}

return TYPE\_FAILED;

}

@Override

protected void onProgressUpdate(Integer... values) {

int progress = values[0];

if (progress > lastProgress) {

listener.onProgress(progress);

lastProgress = progress;

}

}

@Override

protected void onPostExecute(Integer status) {

switch (status) {

case TYPE\_SUCCESS:

listener.onSuccess();

break;

case TYPE\_FAILED:

listener.onFailed();

break;

case TYPE\_PAUSED:

listener.onPaused();

break;

case TYPE\_CANCELED:

listener.onCanceled();

default:

break;

}

}

public void pauseDownload() {

isPaused = true;

}

public void cancelDownload() {

isCanceled = true;

}

private long getContentLength(String downloadUrl) throws IOException {

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()

.url(downloadUrl)

.build();

Response response = client.newCall(request).execute();

if (response != null && response.isSuccessful()) {

long contentLength = response.body().contentLength();

response.close();

return contentLength;

}

return 0;

}

}

1. DownloadService.java

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.app.Service;

import android.content.Intent;

import android.graphics.BitmapFactory;

import android.os.Binder;

import android.os.Environment;

import android.os.IBinder;

import android.support.v4.app.NotificationCompat;

import android.widget.Toast;

import java.io.File;

public class DownloadService extends Service {

private DownloadTask downloadTask;

private String downloadUrl;

private DownloadListener listener = new DownloadListener() {

@Override

public void onProgress(int progress) {

getNotificationManager().notify(1, getNotification("Downloading...", progress));

}

@Override

public void onSuccess() {

downloadTask = null;

// 下载成功时将前台服务通知关闭，并创建一个下载成功的通知

stopForeground(true);

getNotificationManager().notify(1, getNotification("Download Success", -1));

Toast.makeText(DownloadService.this, "Download Success", Toast.LENGTH\_SHORT).show();

}

@Override

public void onFailed() {

downloadTask = null;

// 下载失败时将前台服务通知关闭，并创建一个下载失败的通知

stopForeground(true);

getNotificationManager().notify(1, getNotification("Download Failed", -1));

Toast.makeText(DownloadService.this, "Download Failed", Toast.LENGTH\_SHORT).show();

}

@Override

public void onPaused() {

downloadTask = null;

Toast.makeText(DownloadService.this, "Paused", Toast.LENGTH\_SHORT).show();

}

@Override

public void onCanceled() {

downloadTask = null;

stopForeground(true);

Toast.makeText(DownloadService.this, "Canceled", Toast.LENGTH\_SHORT).show();

}

};

private DownloadBinder mBinder = new DownloadBinder();

@Override

public IBinder onBind(Intent intent) {

return mBinder;

}

class DownloadBinder extends Binder {

public void startDownload(String url) {

if (downloadTask == null) {

downloadUrl = url;

downloadTask = new DownloadTask(listener);

downloadTask.execute(downloadUrl);

startForeground(1, getNotification("Downloading...", 0));

Toast.makeText(DownloadService.this, "Downloading...", Toast.LENGTH\_SHORT).show();

}

}

public void pauseDownload() {

if (downloadTask != null) {

downloadTask.pauseDownload();

}

}

public void cancelDownload() {

if (downloadTask != null) {

downloadTask.cancelDownload();

} else {

if (downloadUrl != null) {

// 取消下载时需将文件删除，并将通知关闭

String fileName = downloadUrl.substring(downloadUrl.lastIndexOf("/"));

String directory = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_DOWNLOADS).getPath();

File file = new File(directory + fileName);

if (file.exists()) {

file.delete();

}

getNotificationManager().cancel(1);

stopForeground(true);

Toast.makeText(DownloadService.this, "Canceled", Toast.LENGTH\_SHORT).show();

}

}

}

}

private NotificationManager getNotificationManager() {

return (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

}

private Notification getNotification(String title, int progress) {

Intent intent = new Intent(this, MainActivity.class);

PendingIntent pi = PendingIntent.getActivity(this, 0, intent, 0);

NotificationCompat.Builder builder = new NotificationCompat.Builder(this);

builder.setSmallIcon(R.mipmap.ic\_launcher);

builder.setLargeIcon(BitmapFactory.decodeResource(getResources(), R.mipmap.ic\_launcher));

builder.setContentIntent(pi);

builder.setContentTitle(title);

if (progress >= 0) {

// 当progress大于或等于0时才需显示下载进度

builder.setContentText(progress + "%");

builder.setProgress(100, progress, false);

}

return builder.build();

}

}

1. activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

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

<Button

android:id="@+id/start\_download"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Start Download" />

<Button

android:id="@+id/pause\_download"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Pause Download" />

<Button

android:id="@+id/cancel\_download"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Cancel Download" />

</LinearLayout>

1. MainActivity.java

import android.Manifest;

import android.content.ComponentName;

import android.content.Intent;

import android.content.ServiceConnection;

import android.content.pm.PackageManager;

import android.os.IBinder;

import android.support.v4.app.ActivityCompat;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

private DownloadService.DownloadBinder downloadBinder;

private ServiceConnection connection = new ServiceConnection() {

@Override

public void onServiceDisconnected(ComponentName name) {

}

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

downloadBinder = (DownloadService.DownloadBinder) service;

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button startDownload = (Button) findViewById(R.id.start\_download);

Button pauseDownload = (Button) findViewById(R.id.pause\_download);

Button cancelDownload = (Button) findViewById(R.id.cancel\_download);

startDownload.setOnClickListener(this);

pauseDownload.setOnClickListener(this);

cancelDownload.setOnClickListener(this);

Intent intent = new Intent(this, DownloadService.class);

startService(intent); // 启动服务

bindService(intent, connection, BIND\_AUTO\_CREATE); // 绑定服务

if (ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.WRITE\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(MainActivity.this, new String[]{ Manifest.permission. WRITE\_EXTERNAL\_STORAGE }, 1);

}

}

@Override

public void onClick(View v) {

if (downloadBinder == null) {

return;

}

switch (v.getId()) {

case R.id.start\_download:

String url = "https://raw.githubusercontent.com/guolindev/eclipse/master/eclipse-inst-win64.exe";

downloadBinder.startDownload(url);

break;

case R.id.pause\_download:

downloadBinder.pauseDownload();

break;

case R.id.cancel\_download:

downloadBinder.cancelDownload();

break;

default:

break;

}

}

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {

switch (requestCode) {

case 1:

if (grantResults.length > 0 && grantResults[0] != PackageManager.PERMISSION\_GRANTED) {

Toast.makeText(this, "拒绝权限将无法使用程序", Toast.LENGTH\_SHORT).show();

finish();

}

break;

default:

}

}

@Override

protected void onDestroy() {

super.onDestroy();

unbindService(connection);

}

}