**package** com.orange5s.constant;

**import** java.io.BufferedReader;

**import**java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileInputStream;

**import**java.io.FileOutputStream;

**import** java.io.InputStreamReader;

**import**java.io.OutputStreamWriter;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.Map;

**import**java.util.Set;

**import**java.util.Map.Entry;

**import** java.util.concurrent.Callable;

**import** java.util.concurrent.ExecutionException;

**import**java.util.concurrent.ExecutorService;

**import**java.util.concurrent.Executors;

**import** java.util.concurrent.FutureTask;

**import** org.apache.http.client.HttpClient;

**import**org.apache.http.conn.ClientConnectionManager;

**import**org.apache.http.conn.params.ConnManagerParams;

**import**org.apache.http.conn.scheme.PlainSocketFactory;

**import**org.apache.http.conn.scheme.Scheme;

**import**org.apache.http.conn.scheme.SchemeRegistry;

**import**org.apache.http.conn.ssl.SSLSocketFactory;

**import**org.apache.http.impl.client.DefaultHttpClient;

**import**org.apache.http.impl.conn.tsccm.ThreadSafeClientConnManager;

**import**org.apache.http.params.BasicHttpParams;

**import**org.apache.http.params.HttpConnectionParams;

**import**org.apache.http.params.HttpParams;

**import** org.json.JSONArray;

**import** org.json.JSONException;

**import** org.json.JSONObject;

**import** com.orange5s.bean.MessageInfo;

**import** com.orange5s.bean.UserInfo;

**import** com.orange5s.crash.CrashExceptionHandler;

**import** com.orange5s.util.HttpPostRequest;

**import** cn.jpush.android.api.JPushInterface;

**import**android.annotation.SuppressLint;

**import** android.app.Activity;

**import**android.app.AlarmManager;

**import** android.app.Application;

**import**android.app.PendingIntent;

**import**android.content.Context;

**import**android.content.Intent;

**import** android.content.SharedPreferences;

**import**android.os.Handler;

**import** android.os.Looper;

**import**android.os.Message;

**import** android.util.Log;

/\*\*

\* For developer startup JPush SDK

\*

\* 一般建议在自定义 Application 类里初始化。也可以在主 Activity 里。

\*/

**publicclass** MyApplication **extends** Application {

**public** UserInfo userInfo=**new** UserInfo();

**public** Map<String, Object>map=**new** HashMap<String, Object>();

**public** ArrayList<Activity>mList = **new** ArrayList<Activity>();

**public** ArrayList<MessageInfo>messageInfos=**new** ArrayList<MessageInfo>();

**public** HashMap<Integer, String>errMsg=**new** HashMap<Integer, String>();

**public** ArrayList<HttpClient>arrayList=**new** ArrayList<HttpClient>();

**public** String comment="";

**privatestaticfinal** String *TAG* = "JPush";

**private** SharedPreferences preferences;

**private** SharedPreferences msgPreferences;

**publicint**msgCount=0;

**publicstatic** MyApplication *application*;

@Override

**publicvoid** onCreate() {

Log.*d*(*TAG*, "[ExampleApplication] onCreate");

**super**.onCreate();

*application*=**this**;

JPushInterface.*setDebugMode*(**true**); // 设置开启日志,发布时请关闭日志

JPushInterface.*init*(**this**); // 初始化 JPush

preferences=getSharedPreferences(Constant.*FILE\_NAME*, Activity.*MODE\_PRIVATE*);

msgPreferences=getSharedPreferences(Constant.*MSG\_FILE\_NAME*, Activity.*MODE\_PRIVATE*);

initData();

initErrorData();

//创建（获得）异常日志捕获类实例

CrashExceptionHandler crashHandler = CrashExceptionHandler.*getInstance*();

crashHandler.init(getApplicationContext(),**this**);

**new** Thread(**new** Runnable() {

@Override

**publicvoid** run() {

Looper.*prepare*();

//上传所有本地保存的未上传成功的异常文件,一旦上传成功便删除本地文件

File fileDir = getFilesDir();

File[] subFile = fileDir.listFiles();

**for** (**int** i = 0; i < subFile.length; i++) {

Log.*i*("errorLogUpload", "当前读出的文件为："+subFile[i].toString());

**if** (!subFile[i].isDirectory()) {

String fileName = subFile[i].getName();

// 判断是否为捕获到的异常，或者是自定义的异常

**if** (fileName.contains("crash-") || fileName.contains("uploadError-")) {

Log.*e*("errorLogUpload", "当前读取到的文件名："+fileName);

String errInfo = print(fileName);

**if**(uploadErrorLog(errInfo)) {

subFile[i].delete();

Log.*e*("errorLogUpload", "上传错误日志成功，删除已上传的错误日志文件："+fileName);

}**else** {

Log.*e*("errorLogUpload", "上传错误信息不成功："+fileName);

}

}

}

}

Looper.*loop*();

}

}).start();

}

**privatevoid** initErrorData(){

errMsg.clear();

errMsg.put(1, "账号异常，请联系客服");

errMsg.put(10001, "参数错误");

errMsg.put(10002, "您还未登录，请先登录");

errMsg.put(10003, "证书错误");

errMsg.put(10004, "请求的方法不存在");

errMsg.put(10005, "用户名或密码不能为空");

errMsg.put(10006, "该用户名不存在");

errMsg.put(10007, "您输入的密码不正确");

errMsg.put(10008, "登录失败，请重新登录");

errMsg.put(10009, "您输入的密码不能为空");

errMsg.put(10010, "新密码与旧密码不能一致");

errMsg.put(10011, "您输入的旧密码不正确");

errMsg.put(10012, "密码修改失败");

errMsg.put(10013, "退出登录失败");

errMsg.put(20001, "工程id不能为空");

errMsg.put(20002, "工程计划id不能为空");

errMsg.put(101, "网络不好，请稍后重试");

errMsg.put(102, "请求失败");

errMsg.put(103, "返回数据格式有误，请联系客服");

errMsg.put(104, "数据异常");

}