

Android Json解析库的使用

Json解析

GSON

Fast-json

GSON和Fast-json的特点和优势

JSON的简介

JSON(JavaScript Object Notation)是一种轻量级的数据交换格式。用于数据转换传输，通用

于PHP,Java,C++,C#,Python等编程语言数据交换传输。

易于阅读和编写，同样也易于机器解析和生成。

基本结构：对象(JsonObject)、数组(JsonArray)。

Gson的简介和特点：

Gson是Google提供的采用Java对象和JSON数据之间进行映射的Java类库。可以将一个JSON字符串转换成一个Java对象，或者反过来。

- 特点

- 1.快速、高效
- 2.代码量少、简洁
- 3.面向对象
- 4.数据传递和解析方便

Fast-json简介和特点

Fast-json是一个性能很好的Java语言实现的JSON解析器和生成器，来自阿里巴巴的工程师开发。具有极快的性能，超越其他任何的Java Json parser，

- 特点

- 1.快速FAST(比其他任何的基于Java的解析器和生成器更快，包括jackson)
- 2.强大(支持普通JDK类包括任意Java Bean Class、Collection、Map、Date或enum)
- 3.零依赖(没有依赖其他任何类库除了JDK)
- 4.支持注解、支持全类型序列化

GSON的基本用法

基本用法：

- 定义实体类
 - 根据需要可以将JSON生成单个实体或列表实体集合
- 代码演示：
- 使用Gson解析JsonObject
 - 使用Gson解析JsonArray
 - 使用Gson将实体转化为JSON数据

代码演示

在build.gradle添加依赖库

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:22.2.0'  
    compile 'com.mcxiaoke.volley:library:1.0.18'  
    compile 'com.jakewharton:butterknife:7.0.1'  
    compile 'com.google.code.gson:gson:2.2.4'  
}
```

具体实现

```
/**
 * @param result:传入的json串
 */
private void dealData(String result) {
    Gson gson = new Gson();
    Books book = gson.fromJson(result, Books.class);
    Log.i("info", book.toString());
    mTvTitle.setText(book.getTitle());
    mTvPublisher.setText(book.getPublisher());
    List list = book.getAuthor();
    String authors = "";
    for (int i = 0; i < list.size(); i++) {
        authors += list.get(i);
    }
    mTvAuthor.setText(authors);

    volley_netWorkImageView(book.getImage());
}

/*
 *用Volley加载图片
 */
private void volley_netWorkImageView(String imageUrl) {

    ImageLoader imageLoader = new
    ImageLoader(MyApplication.getHttpQueue(), new BitmapCache());

    mNetworkImageView.setDefaultImageResId(R.mipmap.ic_launcher);

    mNetworkImageView.setErrorImageResId(R.mipmap.ic_launcher);
    mNetworkImageView.setImageUrl(imageUrl, imageLoader);

}
```

在实体类中的属性名要与Json串中的字段名一致

```
/*
 * 实体类中的字段要与json中的字段一致
 * json中的数组要定义为实体类中的list
 * json中的对象要定义为实体类中的对象
 */
public class Books {
    private String title;
    private String image;
    private List<String> author;
    private String publisher;
    private ArrayList<Tag> tags;
```

Fast-json的基本用法

基本用法

- 定义实体类
- 根据需要可以将JSON生成单个实体或列表实体集合

代码演示

- 使用FastJson解析JsonObject
 - 使用FastJson解析JsonArray
 - 使用FastJson将实体转换为JSON数据
- 添加依赖

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
    compile 'com.mcxiaoke.volley:library:1.0.18'
    compile 'com.jakewharton:butterknife:7.0.1'
    compile files('libs/fastjson-1.2.2.jar')
    compile files('libs/fastjson-1.2.2-sources.jar')
}
```

```

/**
 * @param result:传入的json串
 */
private void dealData(String result) {
    //用Gson解析需两句
    //Gson gson = new Gson();
    //Books book = gson.fromJson(result, Books.class);

    //用FastJson解析就这一句
    Books book= JSON.parseObject(result,Books.class);

    //当result返回的是一个书的集合，我们就可如下
    List<Books> books = JSON.parseObject(result, new
TypeReference<List<Books>>(){});

    Log.i("info", book.toString());
    mTvTitle.setText(book.getTitle());
    mTvPublisher.setText(book.getPublisher());
    List list = book.getAuthor();
    String authors = "";
    for (int i = 0; i < list.size(); i++) {
        authors += list.get(i);
    }
    mTvAuthor.setText(authors);

    volley_netWorkImageView(book.getImage());

}

/*
 *用Volley加载图片
 */
private void volley_netWorkImageView(String imageUrl) {

    ImageLoader imageLoader = new
ImageLoader(MyApplication.getHttpQueue(), new BitmapCache());

    mNetworkImageView.setDefaultImageResId(R.mipmap.ic_launcher);

    mNetworkImageView.setErrorImageResId(R.mipmap.ic_launcher);
    mNetworkImageView.setImageUrl(imageUrl, imageLoader);

}

```

主要代码

FastJson解析单个

```
//用FastJson解析就这一句
Books book = JSON.parseObject(result, Books.class);
```

FastJson解析的是一个对象的集合

```
//当result返回的是一个书的集合，我们就可如下
List<Books> books = JSON.parseObject(result, new
TypeReference<List<Books>>() {
});
```

生成Json

```
//生成json
Books book1 = new Books();
book1.setTitle("book1");
Books book2 = new Books();
book1.setTitle("book1");
Books book3 = new Books();
book1.setTitle("book1");

//将book1对象转换为Json
JSON.toJSON(book1);

//将一系列book对象转化为json
List<Books> lists = new ArrayList<>();
lists.add(book1);
lists.add(book2);
lists.add(book3);
JSON.toJSON(lists);
```

遇到问题

- **Fastjson.jar的下载**

请移步到CSDN下载
下载挤压后有两个jar包需要添加
fastjson-1.2.2-sources.jar和fastjson-1.2.2.jar
项目中添加，copy到libs目录下,右键add as library

- **Duplicate files copied in APK META-INF/LICENSE.txt解决方法**
解决方案：

```
android{
    packagingOptions {
        exclude 'META-INF/NOTICE.txt'
        exclude 'META-INF/LICENSE.txt'
    }
}
```

GSON和Fast-json的实际应用

Json格式

```
{
  "reason": "success",
  "result": [
    {
      "movieId": "226955",
      "movieName": "纯真年代",
      "pic_url": "http://v.juhe.cn/movie/picurl?28137515"
    },
    {
      "movieId": "225955",
      "movieName": "魔镜奇缘",
      "pic_url": "http://v.juhe.cn/movie/picurl?28132456"
    },
    {
      "movieId": "225744",
      "movieName": "百团大战",
      "pic_url": "http://v.juhe.cn/movie/picurl?28132459"
    }
  ]
}
```

主要代码

- **FastJson解析**

```

/**
 * 用fastJson来解析Json
 * @param response
 */
private void getDataJsonPre(String response) {
    ArrayList<Movie> movies = null;
    JSONObject object= null;
    try {
        object = new JSONObject(response);
        movies= (ArrayList<Movie>)
JSON.parseArray(object.getString("result"),Movie.class);

    } catch (JSONException e) {
        e.printStackTrace();
    }
    movieAdapter = new ListMovieAdapter(this, movies);
    moiveList.setAdapter(movieAdapter);
}

```

• Gson解析

```

/**
 * 用Gson来解析Json
 * @param result
 */
private void getData(String result) {
    ArrayList<Movie> movies = null;
    Gson gson = new Gson();
    Type listType = new TypeToken<ArrayList<Movie>>() {
    }.getType();
    try {
        JSONObject object = new JSONObject(result);
        movies = gson.fromJson(object.getString("result"),
listType);
    } catch (JSONException e) {
        e.printStackTrace();
    }

    movieAdapter = new ListMovieAdapter(this, movies);
    moiveList.setAdapter(movieAdapter);
}

```


Demo
[链接地址](#)