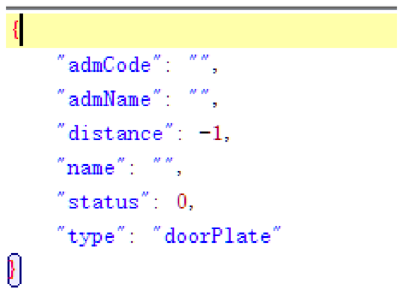
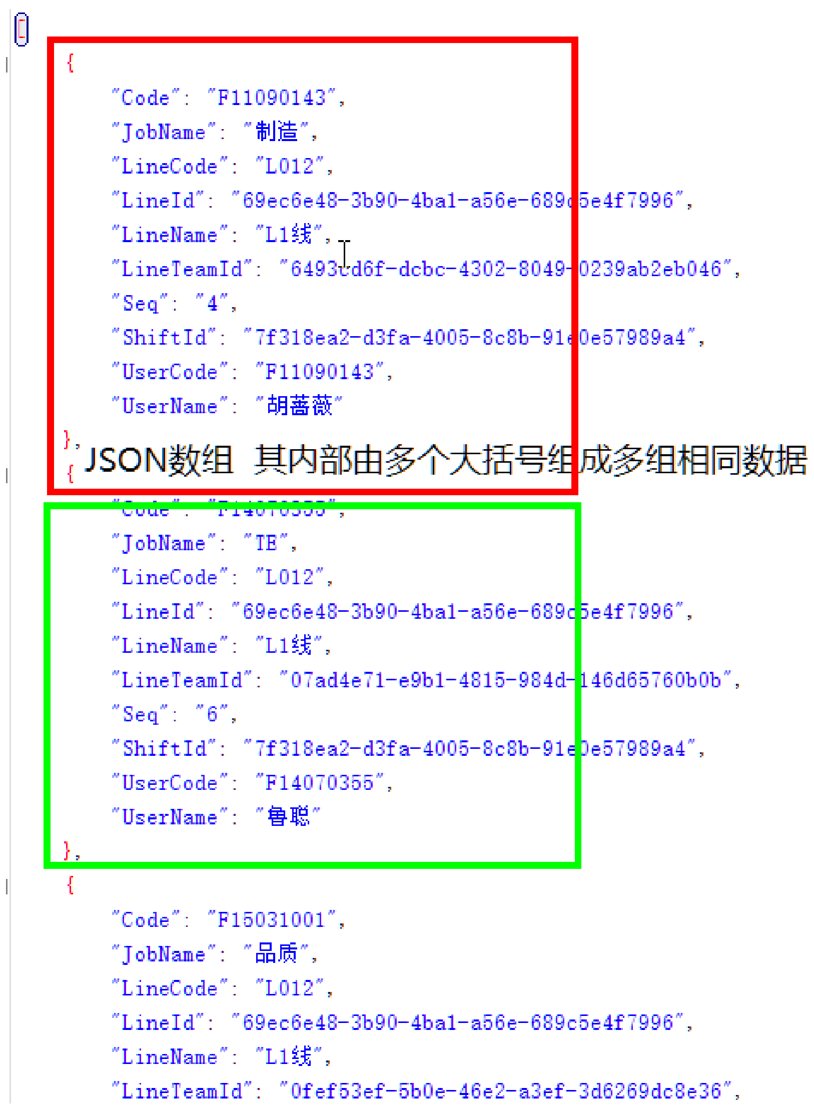
1、通过url获取JSON数据

2、确定返回数据是字符串还是数组



```
{
    "admCode": "",
    "admName": "",
    "distance": -1,
    "name": "",
    "status": 0,
    "type": "doorPlate"
}
```

JSON字符串



```
[
    {
        "Code": "F11090143",
        "JobName": "制造",
        "LineCode": "L012",
        "LineId": "69ec6e48-3b90-4ba1-a56e-689c5e4f7996",
        "LineName": "L1线",
        "LineTeamId": "6493cd6f-dcbc-4302-8049-0239ab2eb046",
        "Seq": "4",
        "ShiftId": "7f318ea2-d3fa-4005-8c8b-91e0e57989a4",
        "UserCode": "F11090143",
        "UserName": "胡蔷薇"
    },
    {
        "Code": "F14070355",
        "JobName": "TE",
        "LineCode": "L012",
        "LineId": "69ec6e48-3b90-4ba1-a56e-689c5e4f7996",
        "LineName": "L1线",
        "LineTeamId": "07ad4e71-e9b1-4815-984d-146d65760b0b",
        "Seq": "6",
        "ShiftId": "7f318ea2-d3fa-4005-8c8b-91e0e57989a4",
        "UserCode": "F14070355",
        "UserName": "鲁聪"
    },
    {
        "Code": "F15031001",
        "JobName": "品质",
        "LineCode": "L012",
        "LineId": "69ec6e48-3b90-4ba1-a56e-689c5e4f7996",
        "LineName": "L1线",
        "LineTeamId": "0fef53ef-5b0e-46e2-a3ef-3d6269dc8e36",
```

JSON数组 其内部由多个大括号组成多组相同数据

3、自动生成JavaBean对象类

4、在项目中通过网络请求获取JSONS数据，得到一个String的字符串

5、解析JSON字符串生成JavaBean对象（注意）

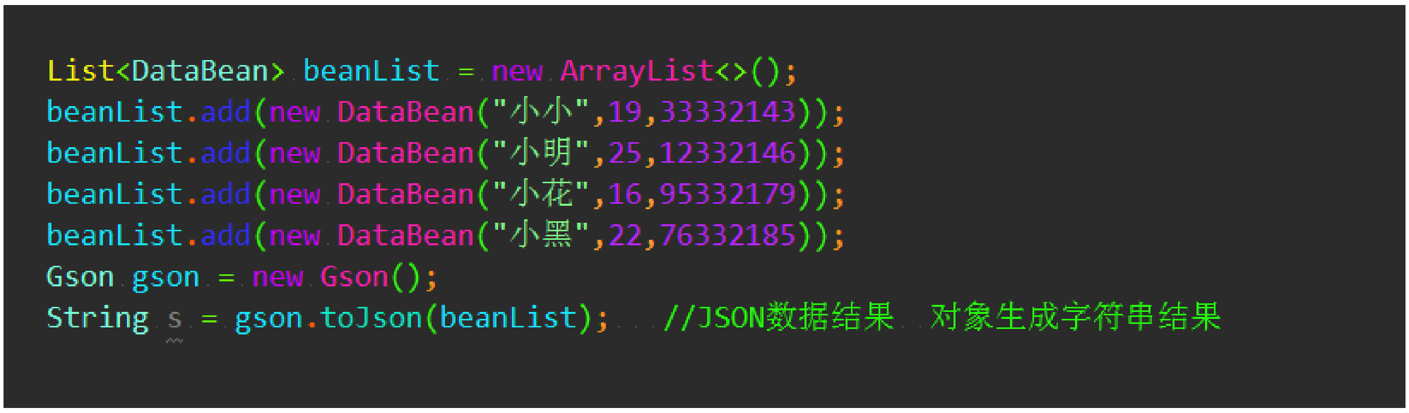
5.1如果返回的数据是JSON字符串的数据

```
TEST.class 是JavaBean对象类
  //解析数据
Gson gson= new Gson();
TEST test = gson.fromJson(bitmap, TEST.class);   //返回为Bean对象数据
//直接调用获取结果
String code = test.getCode();
//赋值给UI
mTextView.setText(code);
```

5.2如果返回的数据是JSON数组的数据

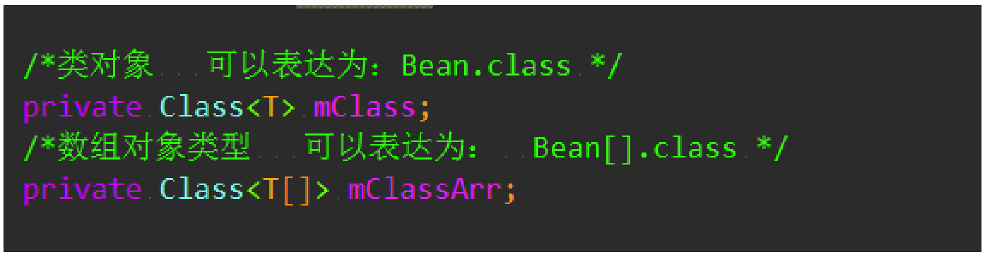
```
  //解析数据
private List<TEST> TT = new ArrayList<>();       //定义的成员变量 集合的泛型为
TEST是JavaBean对象类
bitmap: 网络请求回来的结果
//在获取到数据后进行解析必须要进行手动的进行异常的捕获处理，必免出现字段或数据
类型异常导致程序崩溃
    try{

        Gson gson= new Gson();
        TEST[] tests = gson.fromJson(bitmap,  TEST[].class);      //返回为Bean
        数组的数据
        List<TEST> tests1 = Arrays.asList(tests);          //数组转集合
        //拿值
        TT.addAll(tests1);                 //将生成的集合添加到成员变量中去
        String jobName = TT.get(1).getJobName();        //通过定义的成员变量获取
        值
        //赋值更新UI
        mTextView.setText(jobName);


    }catch(Exception  e){
                         Log.e("ERRR",  "网络异常，请求失败！");
}
```

```
对象转json字符串
Student  student  =  new  Student();
student.id  =  1;
student.nickName  =  "乔晓松";
student.age  =  22;
student.email  =  "965266509@qq.com";
gson.toJson(student));
```



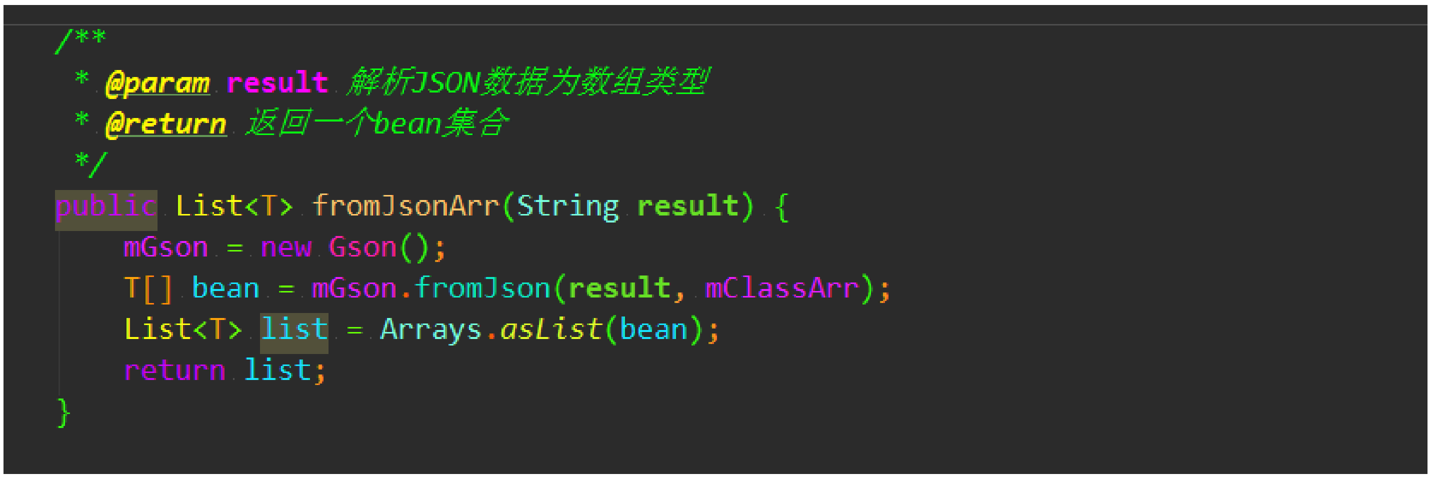
```
List<DataBean> beanList = new ArrayList<>();
beanList.add(new DataBean("小小",19,33332143));
beanList.add(new DataBean("小明",25,12332146));
beanList.add(new DataBean("小花",16,95332179));
beanList.add(new DataBean("小黑",22,76332185));
Gson gson = new Gson();
String s = gson.toJson(beanList);   //JSON数据结果  对象生成字符串结果
```

通用的泛型对象



```
/*类对象   可以表达为: Bean.class */
private Class<T> mClass;
/*数组对象类型   可以表达为:   Bean[].class */
private Class<T[]> mClassArr;
```

最主要的方法 解析JSON数组



```
/**
 * @param result 解析JSON数据为数组类型
 * @return 返回一个bean集合
 */
public List<T> fromJsonArr(String result) {
    mGson = new Gson();
    T[] bean = mGson.fromJson(result, mClassArr);
    List<T> list = Arrays.asList(bean);
    return list;
}
```

fastjson的基本使用

依赖：

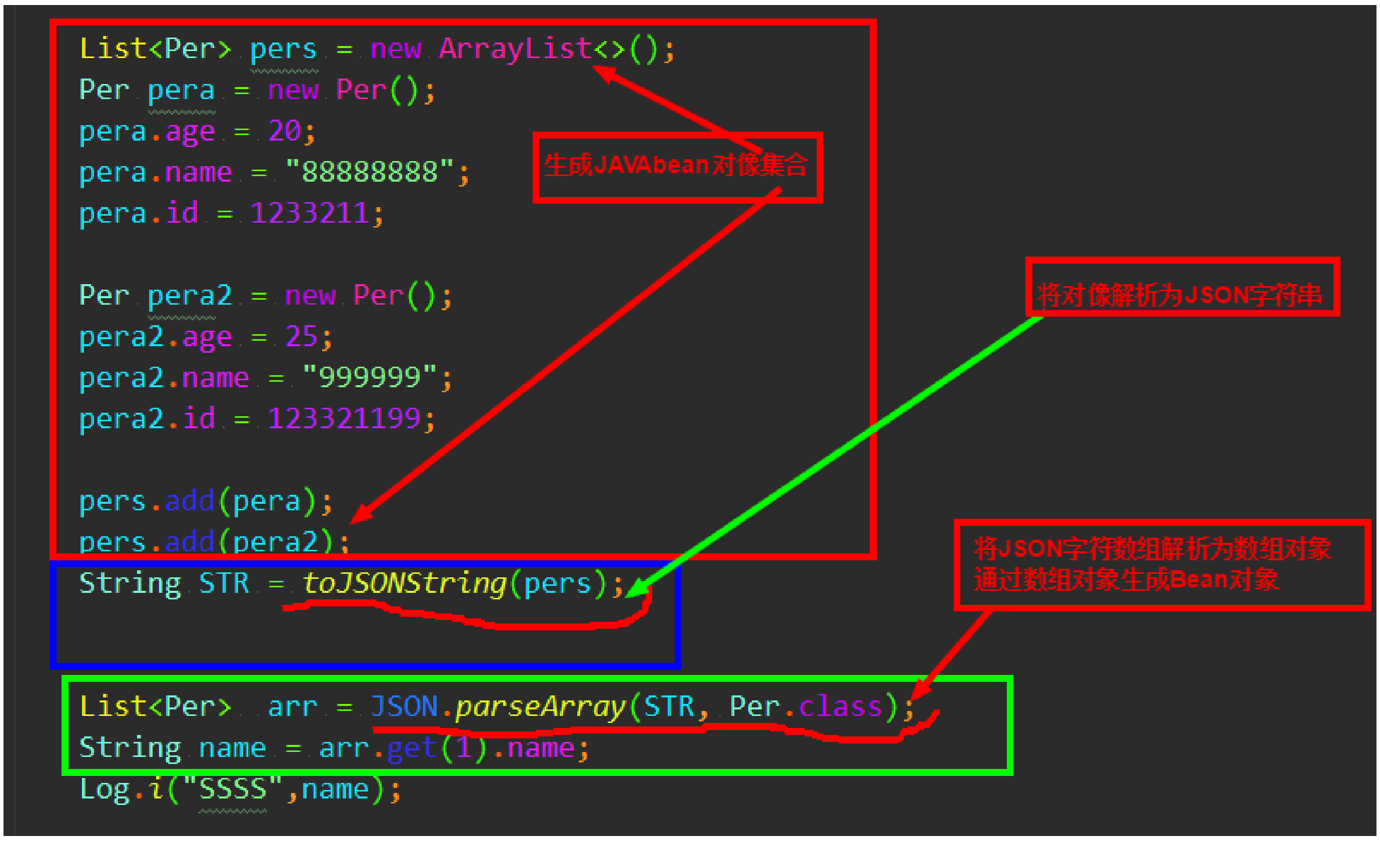
```
compile 'com.alibaba:fastjson:1.2.34'
compile 'com.alibaba:fastjson:1.1.59.android'
```

1、对象转换为JSON字符串

```
Map<String,String> map = new HashMap<>();
map.put("name1","liuJingrong");
map.put("name2","KangKai");
map.put("name3","ZengKuan");
String s = toJSONString(map);

String per1 = JSON.toJSONString(new Per("LJR", 20, 202020));
String ljr = JSON.toJSONString(new Per("LJR", 20, 202020));
```

2、将JSON字符串解析成对象

```
Per per = JSON.parseObject(ljr, Per.class);
```



```
List<Per> pers = new ArrayList<>();
Per pera = new Per();
pera.age = 20;
pera.name = "88888888";
pera.id = 1233211;

Per pera2 = new Per();
pera2.age = 25;
pera2.name = "999999";
pera2.id = 123321199;

pers.add(pera);
pers.add(pera2);
String STR = toJSONString(pers);

List<Per>  arr = JSON.parseArray(STR, Per.class);
String name = arr.get(1).name;
Log.i("SSSS",name);
```