# Json对应javabean

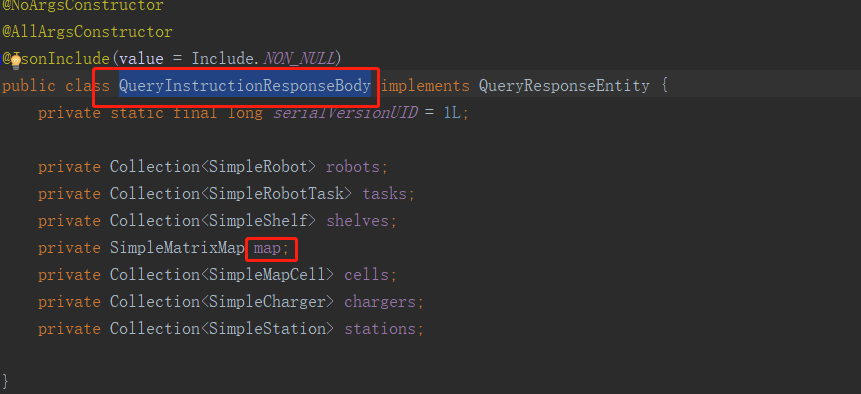
## 1．Json解释

注意，响应javabean类头一般用@JsonInclude(value = Include.NON\_NULL)标注，意思是当

把bean转化未json时，会把java属性为null的的字段过滤掉，收到的json响应不会有改字段，一般请求体的bean不用它，发什么就是什么，不发应该也是bean的默认值，前端有检验

1

对应javabean



整个body对应的是bean类QueryInstructionResponseBody

Key属性map对应的是SimpleMatrixMap 类

打开类看对应关系



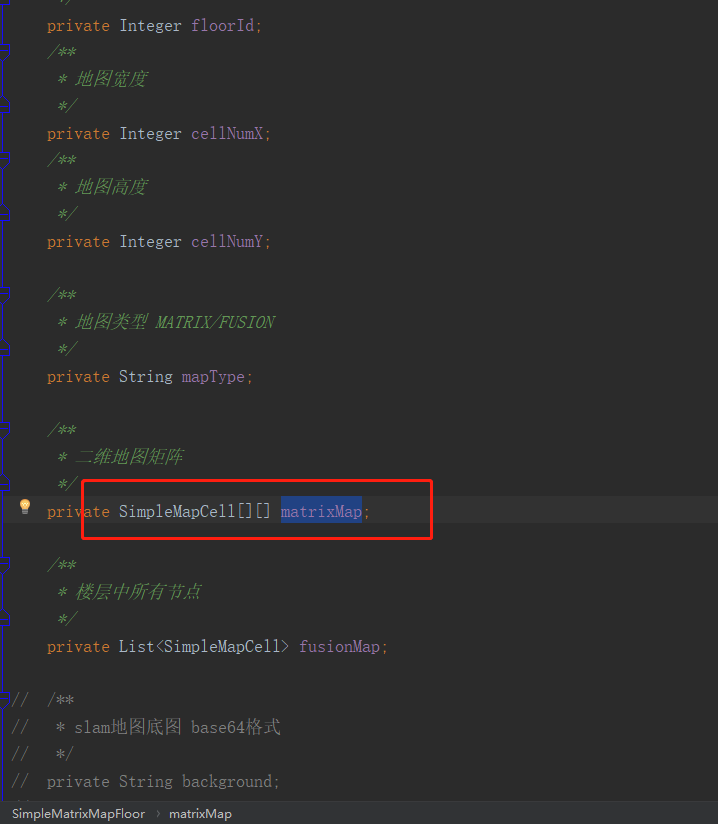
其中floors 是个List<SimpleMatrixMapFloor>

对应jsong格式｛｝是list的每个元素，切记最外层是个方括号[]代表数组，在此是个只有一个元素的数组，比如foloors2层的话就是[｛｝，｛｝]

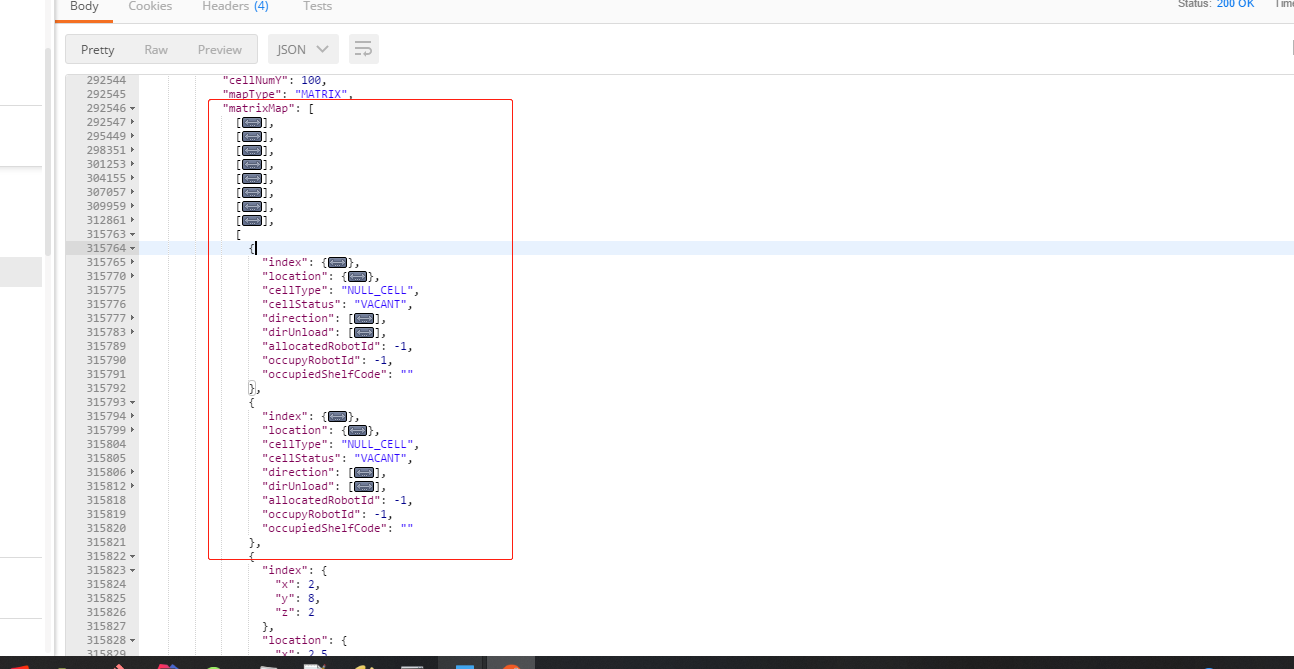


二维数组对应关系

我们打开SimpleMatrixMapFloor



看到matrixMap 是个SimpleMapCell类型的二维数组， 即每个二维元素存储的都是SimpleMapCell 包含属性index,locttion,celltype,cellstatus,等等但是注意返回的字段比SimpleMapCell属性字段少，因为用@JsonInclude(value = Include.NON\_NULL)注解了，null的属性不参与转化json。看json格式



二维数组就是一个平面矩阵，最外层中括号就是一个一维数组，第一个元素又是一个一维数组，代表矩阵的第一行，每行有多个SimpleMapCell（index,location,celltype….）数组元素组成，每个bean数组元素用大括号｛｝，

## 2.大括号，中括号，冒号，直接写的三种的用法｛｝，【】

一．｛｝：

只要里面是类：就用大括号比如

1本例子的map 字段为 java类的类型，所以json格式map:{ }，

2还有SimpleMapCell类型的数组中也是用｛｝大括号括住SimpleMapCell（index,location,celltype….）

3. SimpleMapCell属性index也是个类 Ipoint类，

二．中括号【】

数组，或者list(其实就是1维数组)类型，，

1floors字段类型为List<SimpleMatrixMapFloor> ，所以是 floors：【｛SimpleMatrixMapFloor｝，｛SimpleMatrixMapFloor｝】代表有两层，两个数组元素，又因为SimpleMatrixMapFloor是bean所以用大括号围住

2数组 direction

protected int[] direction;

整数行的，所以json格式为 derection:[1,2,3,4,4.6]

3.二维数组

如matrixMap 字段

private SimpleMapCell[][] matrixMap;

二维数组就是一个平面矩阵，最外层中括号就是一个一维数组，第一个元素又是一个一维数组，代表矩阵的第一行，每行有多个SimpleMapCell（index,location,celltype….）数组元素组成，每个bean数组元素用大括号｛｝,

Json格式 matrixMap：【

【｛bean1｝,｛bean2｝】，

【｛bean1｝,｛bean2｝】，

【【｛bean1｝,｛bean2｝】，

【｛bean1｝,｛bean2｝】，

【｛bean1｝,｛bean2｝】，

】j就是个5行2列的二维数组



三．普通类型

如，string,int,boole,float直接在冒号后写

如celltype string 类型 json格式

Celltype: shelfcell