<http://blog.csdn.net/gchb9527/article/details/8688279>

**2.将json字符串转换为java对象**

同样先将json字符串转换为json对象，再将json对象转换为java对象，如下所示。

JSONObject obj = new JSONObject().fromObject(jsonStr);//将json字符串转换为json对象

将json对象转换为java对象

Person jb = (Person)JSONObject.toBean(obj,Person.class);//将建json对象转换为Person对象

**3.将java对象转换为json字符串**

先将java对象转换为json对象，在将json对象转换为json字符串

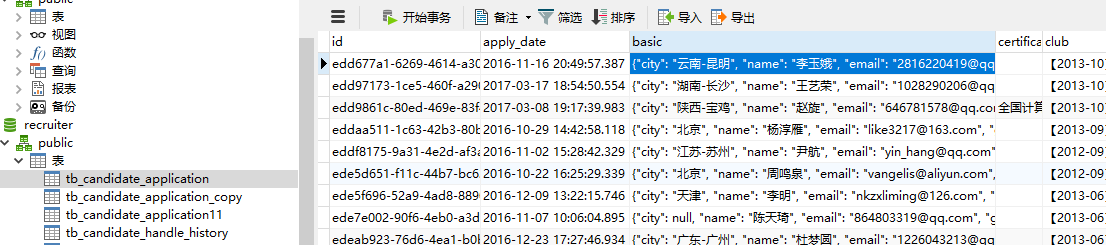
JSONObject json = JSONObject.fromObject(obj);//将java对象转换为json对象

String str = json.toString();//将json对象转换为字符串

# 依赖

*<!-- json jar包 -->*<**dependency**>  
 <**groupId**>net.sf.json-lib</**groupId**>  
 <**artifactId**>json-lib</**artifactId**>  
 <**version**>2.4</**version**>  
 <**classifier**>jdk15</**classifier**>  
</**dependency**>

# 1项目中postgresql的转化 java与json的转化



## 1.如果数据库中一个表的字段是jsonb的格式，可以将这个jsonb的对象制作成一个DTO将来数据传递肯定会用到，这里以 recruiter 中的项目为例 Basic 为一个jsonb的格式

**public class** Basic {  
  
 **private** String **name**;  
  
 **private** String **gender**;  
  
 **private** String **email**;  
  
 **private** String **mobile**;  
  
 **private** String **birthday**;  
  
 **private** String **province**;  
  
 **private** String **city**;  
  
 **private** String **political**;  
  
 */\*\*  
 \* 空构造函数.  
 \*/* **public** Basic() {  
 }

## 2. 导入 数据库，将对象转化为json ，applicationDTO中有 basic对象

**private** ObjectMapper objectMapper = **new** ObjectMapper();

*//将basic对象转换为json*String basic = **objectMapper**.writeValueAsString(applicationDTO.getBasic());  
applications.setBasic(basic);

## 3. 从数据库中读取String，然后将json转化为对象，applications 中是String 格式的josnb

Basic basic = **objectMapper**.readValue(applications.getBasic(), Basic.**class**);  
applicationDTO.setBasic(basic);