JSON转化工具（需要Jackson依赖）——JsonMapper类

作用：将一个类转换成Json对象，也可以将一个Json的字符串转换成系统中一个类的对象。

1. 首先通过ObjectMapper设置序列化和反序列化时候的属性
2. 通过静态代码块初始化ObjectMapper

*/\*\*  
 \* Json工具类：  
 \* 将一个类转换成Json对象，也可以将一个Json的字符串转换成系统中一个类对象。  
 \*/*@Slf4j  
**public class** JsonMapper {  
  
 */\*\*  
 \* 构建全局的转换器ObjectMapper  
 \*/* **private static** ObjectMapper *objectMapper* = **new** ObjectMapper();  
  
 **static** {  
 *// config 排除掉为空的字段。  
objectMapper*.disable(DeserializationConfig.Feature.***FAIL\_ON\_UNKNOWN\_PROPERTIES***); *objectMapper*.configure(SerializationConfig.Feature.***FAIL\_ON\_EMPTY\_BEANS***, **false**);  
 *objectMapper*.setFilters(**new** SimpleFilterProvider().setFailOnUnknownId(**false**)); *objectMapper*.setSerializationInclusion(JsonSerialize.Inclusion.***NON\_EMPTY***);  
 }  
  
 */\*\*  
 \* Object转json字符串  
 \*  
 \** ***@param:*** *[src]  
 \** ***@return:*** *java.lang.String  
 \** ***@author:*** *nino  
 \** ***@date:*** *2019/5/27 13:01  
 \*/* **public static** <T> String obj2String(T src) {  
 **if** (src == **null**) {  
 **return null**;  
 }  
 **try** {  
 **return** src **instanceof** String ? (String) src : *objectMapper*.writeValueAsString(src);  
 } **catch** (Exception e) {***log***.warn(**"parse object to String exception, error:{}"**, e);  
 **return null**;  
 }  
 }  
  
 */\*\*  
 \* json字符串转Object  
 \*  
 \** ***@param:*** *[src, typeReference]  
 \** ***@return:*** *T  
 \** ***@author:*** *nino  
 \** ***@date:*** *2019/5/27 13:04  
 \*/* **public static** <T> T string2Obj(String src, TypeReference<T> typeReference) {  
 **if** (src == **null** || typeReference == **null**) {  
 **return null**;  
 }  
 **try** {  
 **return** (T) (typeReference.getType().equals(String.**class**) ? src : *objectMapper*.readValue(src, typeReference));  
 } **catch** (Exception e) {***log***.warn(**"parse String to Object exception, String:{}, TypeReference<T>:{}, error:{}"**, src, typeReference.getType(), e);  
 **return null**;  
 }  
 }  
}

ObjectMapper类是Jackson库的主要类。提供了一些功能将转换Java对象匹配JSON结构，反之亦然。

json是默认以linkedHashMap来存放object对象，所以当我们反序列化时,必须进行数据类型绑定,指明我们的map中的value类型。

类似的如集合中的其他元素在序列化和反序列化时,通过都需要进行类似的操作,否则无法进行数据元素获取。

## 总结：

1. jackson基本用法步骤
2. 定义转化器
3. java对象转json
4. json反序列化成对象
5. 常用错误

TypeReference -- 让Jackson Json在List/Map中识别自己的Object

1. jackson优缺点

优点：jackson适合对任意对象做序列化成json传输

缺点：jackson适合对简单对象进行序列化和反序列化，，一旦涉及到复杂对象，jackson的反序列化往往会出现问题。