Jackson——json和object之间转换的工具

1.ObjectMapper用法

public class JacksonTester {

public static void main(String args[]){

ObjectMapper mapper = new ObjectMapper();

String jsonString = "{\"name\":\"Mahesh\", \"age\":21}";

//map json to student

**Student** student = mapper.**readValue**(jsonString, **Student.class**);

System.out.println(student);

//map student to json

mapper.enable(SerializationConfig.Feature.INDENT\_OUTPUT);

jsonString = mapper.**writeValueAsString**(student);

System.out.println(jsonString);

}

}

class Student {

private String name;

private int age;

public Student(){}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String toString(){

return "Student [ name: "+name+", age: "+ age+ " ]";

}

}

验证输出

Student [ name: Mahesh, age: 21 ]

{

"name" : "Mahesh",

"age" : 21

}

2.平台封装的JacksonUtil.java

1）获取post的json数据的对象，

public static Object getPostJsonObject(String data, Class<?>[] clsArray)

{

try {

if (clsArray.length == 0)

return null;

if (clsArray.length == 1) {

Object returnBean = getObjectMapper().readValue(data, clsArray[0]);

return returnBean;

}

//创建一个新的class数组，抛弃原来第一个元素

Class[] clsChild = new Class[clsArray.length - 1];

for (int i = 1; i < clsArray.length; i++) {

clsChild[(i - 1)] = clsArray[i];

}

//注册将要转化的对象类型JavaType

JavaType javaType = getObjectMapper()

.getTypeFactory()

.constructParametrizedType(clsArray[0], clsArray[0], clsChild);

//通常将JavaType转化为集合类泛型对象 如Map(String, String)

Object returnBean = getObjectMapper().readValue(data, javaType);

return returnBean;

}

catch (Exception e) {

e.printStackTrace();

}return null;

}

3.  [Jackson 处理复杂类型(List,map)两种方法](http://blog.csdn.net/zhuyijian135757/article/details/38269715)

方法一:

**[java]** [view plain](http://blog.csdn.net/zhuyijian135757/article/details/38269715) [copy](http://blog.csdn.net/zhuyijian135757/article/details/38269715)

1. String jsonString="[{'id':'1'},{'id':'2'}]";
2. ObjectMapper mapper = **new** ObjectMapper();
3. JavaType javaType = mapper.getTypeFactory().constructParametricType(List.**class**, Bean.**class**);
4. //如果是Map类型  mapper.getTypeFactory().constructParametricType(HashMap.class,String.class, Bean.class);
5. List<Bean> lst =  (List<Bean>)mapper.readValue(jsonString, javaType);

方法二:

**[java]** [view plain](http://blog.csdn.net/zhuyijian135757/article/details/38269715) [copy](http://blog.csdn.net/zhuyijian135757/article/details/38269715)

1. String jsonString="[{'id':'1'},{'id':'2'}]";
2. ObjectMapper mapper = **new** ObjectMapper();
3. List<Bean> beanList = mapper.readValue(jsonString, **new** TypeReference<List<Bean>>() {});

4. 自定义类型用到了**TypeFactory**的**constructParametrizedType**方法

org.codehaus.jackson.map.type.TypeFactory

该方法用来**定义JSON转化对象的泛型类型**!

查看API：

（<http://tool.oschina.net/apidocs/apidoc?api=jackson-1.9.9>）

我们看到TypeFactory的两个重载的constructParametrizedType：

1）

public [JavaType](http://tool.oschina.net/uploads/apidocs/jackson-1.9.9/org/codehaus/jackson/type/JavaType.html) constructParametricType([Class](http://docs.oracle.com/javase/6/docs/api/java/lang/Class.html?is-external=true)<?> parametrized,

[Class](http://docs.oracle.com/javase/6/docs/api/java/lang/Class.html?is-external=true)<?>... parameterClasses)

Factory method for constructing [JavaType](http://tool.oschina.net/uploads/apidocs/jackson-1.9.9/org/codehaus/jackson/type/JavaType.html) that represents a parameterized type. For example, to represent type List<Set<Integer>>, you could call

**TypeFactory.parametricType(List.class, Integer.class);**

NOTE: type modifiers are NOT called on constructed type **itself**; but are called for contained types.

从以下版本开始:

1.5

2）

public [JavaType](http://tool.oschina.net/uploads/apidocs/jackson-1.9.9/org/codehaus/jackson/type/JavaType.html) constructParametricType([Class](http://docs.oracle.com/javase/6/docs/api/java/lang/Class.html?is-external=true)<?> parametrized,

[JavaType](http://tool.oschina.net/uploads/apidocs/jackson-1.9.9/org/codehaus/jackson/type/JavaType.html)... parameterTypes)

Factory method for constructing [JavaType](http://tool.oschina.net/uploads/apidocs/jackson-1.9.9/org/codehaus/jackson/type/JavaType.html) that represents a parameterized type. For example, to represent type **List<Set<Integer>>**, you could call

**JavaType inner = TypeFactory.parametricType(Set.class, Integer.class);**

**TypeFactory.parametricType(List.class, inner);**

NOTE: type modifiers are NOT called on constructed type itself; but are called for contained types.

从以下版本开始:

1.5

可以看到：

1）二者返回JavaType对象，可用于objectMapper.readValue(str, javatype)，json字符串将转化成该类对象

2）两个constructParametrizedType的第一个参数是基本类型，第二个参数开始是可变类型，指定基本类型的泛型类型,比如：(List.class,String.class,Integer.class)就是List<String, Integer>

3）第二个constructParametrizedType的第二个参数开始传入JavaType，将成为嵌套类型，比如JavaType为Set<String>，则(List.class, JavaType)就是List<Set<String>>