<http://www.cnblogs.com/songzhen/p/5794494.html>

一、导入jar包：

（因为之前使用的是jackson 1.x的jar包，所以在把json转换成list时候没有objectMapper.getTypeFactory()这个方法，而使用jackson 2.x的jar包会有）

jackson 1.x：jackson-all-1.7.6.jar（1个核心文件）

jackson 2.x：jackson-annotations-2.5.0.jar、jackson-core-2.5.0.jar、jackson-databind-2.5.0.jar（3个核心文件）

二、建立java对象User：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

package com.cn.jackson.entity;

public class User {

private String name;

private Integer age;

private String address;

public User(){

}

public User(String name, Integer age, String address) {

super();

this.name = name;

this.age = age;

this.address = address;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

[复制代码](javascript:void(0);)

三、测试jackson 2.x的使用方法：

[复制代码](javascript:void(0);)

package com.cn.jackson.test;

import java.io.IOException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import com.cn.jackson.entity.User;

import com.fasterxml.jackson.core.JsonEncoding;

import com.fasterxml.jackson.core.JsonGenerator;

import com.fasterxml.jackson.databind.JavaType;

import com.fasterxml.jackson.databind.ObjectMapper;

public class Test\_2\_x {

public static void main(String[] args) {

try {

ObjectMapper objectMapper = new ObjectMapper();

JsonGenerator jsonGenerator = objectMapper.getJsonFactory()

.createJsonGenerator(System.out, JsonEncoding.UTF8);

//对象转JSON

User user = new User("张三", 23, "深圳市");

System.out.println("ObjectMapper方式");

objectMapper.writeValue(System.out, user);//直接输出到控制台

String userJsonStr = objectMapper.writeValueAsString(user);//返回字符串，输出

System.out.println(userJsonStr);

System.out.println("JsonGenerator方式");

jsonGenerator.writeObject(user);

System.out.println();

//map转JSON

Map<String , Object> map = new HashMap<String ,Object>();

map.put("one", new User("张", 12, "深圳"));

map.put("two", new User("李", 22, "武汉"));

map.put("three", new User("王", 32, "北京"));

System.out.println("ObjectMapper方式");

objectMapper.writeValue(System.out, map);

//JsonGenerator方式同上

//list转JSON

List<User> list = new ArrayList<User>();

list.add(new User("张", 12, "深圳"));

list.add(new User("李", 22, "武汉"));

list.add(new User("王", 32, "北京"));

System.out.println("ObjectMapper方式");

objectMapper.writeValue(System.out, list);

//JsonGenerator方式同上

//JSON转java对象：

String json = "{\"name\":\"张三\",\"age\":23,\"address\":\"深圳市\"}";

User zhang = objectMapper.readValue(json, User.class);

System.out.println(zhang.getName()+"\n"+zhang.getAge()+"\n"+zhang.getAddress());

//JSON转list

String listJson = "[{\"name\":\"张三\",\"age\":21,\"address\":\"深圳\"},"

+ "{\"name\":\"李四\",\"age\":11,\"address\":\"武汉\"},"

+ "{\"name\":\"王五\",\"age\":31,\"address\":\"北京\"}]";

JavaType javaType1 = objectMapper.getTypeFactory().constructParametricType(ArrayList.class, User.class);

List<User> userList = (List<User>)objectMapper.readValue(listJson, javaType1);

for (User user1 : userList) {

System.out.println(user1.getName()+"\t"+user1.getAge()+"\t"+user1.getAddress());

}

//JSON转map

String mapJson = "{\"one\":{\"name\":\"张三\",\"age\":21,\"address\":\"深圳\"},"

+ "\"two\":{\"name\":\"李四\",\"age\":11,\"address\":\"武汉\"},"

+ "\"three\":{\"name\":\"王五\",\"age\":31,\"address\":\"北京\"}}";

JavaType javaType2 = objectMapper.getTypeFactory().constructParametricType(HashMap.class, String.class, User.class);

Map<String,User> userMap = (Map<String,User>)objectMapper.readValue(mapJson, javaType2);

User one = userMap.get("one");

User two = userMap.get("two");

User three = userMap.get("three");

System.out.println(one.getName()+"\t"+one.getAge()+"\t"+one.getAddress());

System.out.println(two.getName()+"\t"+two.getAge()+"\t"+two.getAddress());

System.out.println(three.getName()+"\t"+three.getAge()+"\t"+three.getAddress());

} catch (IOException e) {

e.printStackTrace();

}

}

}

[复制代码](javascript:void(0);)

四、测试jackson 1.x的使用方法：

[复制代码](javascript:void(0);)

package com.cn.jackson.test;

import java.util.List;

import java.util.Map;

import org.codehaus.jackson.map.ObjectMapper;

import org.codehaus.jackson.type.TypeReference;

import com.cn.jackson.entity.User;

public class Test\_1\_x {

public static void main(String[] args) {

try {

//后面几种转换同Test\_2\_x一样 //对象转JSON //map转JSON //list转JSON //JSON转java对象

//JSON转list

ObjectMapper objectMapper = new ObjectMapper();

String listJson = "[{\"name\":\"张三\",\"age\":21,\"address\":\"深圳\"},"

+ "{\"name\":\"李四\",\"age\":11,\"address\":\"武汉\"},"

+ "{\"name\":\"王五\",\"age\":31,\"address\":\"北京\"}]";

List<User> userList = objectMapper.readValue(listJson, new TypeReference<List<User>>() {});

for (User user1 : userList) {

System.out.println(user1.getName()+"\t"+user1.getAge()+"\t"+user1.getAddress());

}

//JSON转map

String mapJson = "{\"one\":{\"name\":\"张三\",\"age\":21,\"address\":\"深圳\"},"

+ "\"two\":{\"name\":\"李四\",\"age\":11,\"address\":\"武汉\"},"

+ "\"three\":{\"name\":\"王五\",\"age\":31,\"address\":\"北京\"}}";

Map<String,User> userMap = objectMapper.readValue(mapJson, new TypeReference<Map<String,User>>() {});

User one = userMap.get("one");

User two = userMap.get("two");

User three = userMap.get("three");

System.out.println(one.getName()+"\t"+one.getAge()+"\t"+one.getAddress());

System.out.println(two.getName()+"\t"+two.getAge()+"\t"+two.getAddress());

System.out.println(three.getName()+"\t"+three.getAge()+"\t"+three.getAddress());

} catch (Exception e) {

// TODO: handle exception

}

}

}

[复制代码](javascript:void(0);)