链滴

继“Java 如何方便地访问深层的 map 的值？”

作者：shuiniu

原文链接：https://ld246.com/article/1527064880763

来源网站：链滴

昨晚问了个问题：https://hacpai.com/article/1526989190264?p=1&m=1#1527061155002，今写了个工具解决，暂时能用，边用边完善。

感谢

@88250 @Vanessa 的回复；

先看使用方法：

```
package utils;


import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.junit.Test;

import com.alibaba.fastjson.JSON;

public class T {

    @Test
    public void jsonMap() {
        SuperMap json = SuperMap.createInstence("{\"test\":{\"pp\":\"8983431w\"}, \"user\":{\"id
":123, \"username\":\"linus\", \"emails\":[\"email1\", \"email2\"]}}");

        System.out.println(json.get("test.pp", int.class));
        System.out.println(json.get("user", User.class));
        System.out.println(json.get("user.emalis[0]", String.class));
    }

    @Test
    public void testReg() {
        Pattern p = Pattern.compile("(\\w+)\\[(\\d+)\\]$");

        Matcher a = p.matcher("nihao[4]");
        a.find();
        System.out.println(a.group(2));
    }
}

class User {
    private Long id;
    private String username;
    private String[] emails;

    public String toString() {
        return JSON.toJSONString(this);
    }

    public Long getId() {
        return id;
    }
    public void setId(Long id) {
```

```
        this.id = id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String[] getEmails() {
        return emails;
    }
    public void setEmails(String[] emails) {
        this.emails = emails;
    }
}```

# 再看实现
```java
package utils;

import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Queue;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;

/**
 * 一个map的子类，使用key path 获取map的值，以来fastjson
 *
 * @author 水牛叔叔(cnliangwei@foxmail.com)
 *
 */
public class SuperMap extends HashMap<String, T> {
    private static final long serialVersionUID = 1L;
    private static final Pattern listReg = Pattern.compile("(\\w+)\\[(\\d+)\\]$");

    /**
     *
     * @param jsonString
     * @return
     */
    public static SuperMap createInstence(String jsonString) {
        try {
            return JSONObject.parseObject(jsonString, SuperMap.class);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
```
```

```
@SuppressWarnings({ "unchecked", "hiding" })
public <T> T get(String key, Class<T> clazz){
    if(key==null || "".equals(key.trim())) {
        return null;
    }

    key = key.replaceAll("\\.+", ".");
    String keys[] = key.split("\\.");

    Queue<String> queue = new LinkedList<String>();
    for(String k : keys) {
        queue.add(k);
    }

    Object tmp = this;
    Matcher m;
    while(!queue.isEmpty() && !tmp.getClass().isPrimitive()) {
        key = queue.poll();
        m = listReg.matcher(key);

        if(m.find()) {
            if(tmp instanceof JSONObject) {
                tmp = ((JSONObject)tmp).get(m.group(1));
            } else if(tmp instanceof SuperMap){
                tmp = ((SuperMap)tmp).get(m.group(1));
            }

            if(!(tmp instanceof List)) {
                tmp = null;
            } else {
                int index = Integer.valueOf(m.group(2));

                if(index>(((List<?>)tmp).size()-1)) {
                    tmp = null;
                } else {
                    tmp = ((List<?>)tmp).get(index);
                }
            }
        } else {
            if(tmp instanceof JSONObject) {
                tmp = ((JSONObject)tmp).get(key);
            } else if(tmp instanceof SuperMap){
                tmp = ((SuperMap)tmp).get(key);
            }
        }

        if(tmp==null) {
            return null;
        }
    }

    if(!queue.isEmpty()) {
        return null;
    }
```

原文链接：继“Java 如何方便地访问深层的 map 的值？”

```
        if(clazz==null) {
            return (T) tmp;
        }

        Class<?> tc = tmp.getClass();
        if(tc.equals(clazz)) {
            return (T) tmp;
        } else if(clazz.isPrimitive() || isPrimitive(clazz)) {
            try {

                if (clazz.equals(Integer.class) || clazz.equals(Integer.TYPE)) {
                    tmp =  Integer.valueOf(tmp.toString());
                } else if (clazz.equals(Boolean.class) || clazz.equals(Boolean.TYPE)) {
                    tmp =  Boolean.valueOf(tmp.toString());
                } else if (clazz.equals(Long.class) || clazz.equals(Long.TYPE)) {
                    tmp =  Long.valueOf(tmp.toString());
                } else if (clazz.equals(Double.class) || clazz.equals(Double.TYPE)) {
                    tmp =  Double.valueOf(tmp.toString());
                } else if (clazz.equals(Float.class) || clazz.equals(Float.TYPE)) {
                    tmp =  Float.valueOf(tmp.toString());
                } else if (clazz.equals(Short.class) || clazz.equals(Short.TYPE)) {
                    tmp =  Short.valueOf(tmp.toString());
                } else if (clazz.equals(Byte.class) || clazz.equals(Byte.TYPE)) {
                    tmp =  Byte.valueOf(tmp.toString());
                } else {
                    tmp = null;
                }
            } catch(Exception e) {
                tmp = null;
            }
            return (T) tmp;
        } else if(clazz.equals(String.class)) {
            tmp = tmp.toString();
            return (T) tmp;
        } else if(tmp instanceof JSONArray){
            return ((JSONArray)tmp).toJavaObject(clazz);
        } else if(tmp instanceof JSONObject){
            return ((JSONObject)tmp).toJavaObject(clazz);
        } else {
            return null;
        }
    }

    private boolean isPrimitive(Class<?> clazz) {
        try {
            return ((Class<?>) clazz.getField("TYPE").get(null)).isPrimitive();
        } catch (Exception e) {
            return false;
        }
    }
}
```