json一键转换为易语言自定义数据类型

大家好,今天给大家带来的是将JSON数据类型转换为易语言自定义数据类型的实现算法与样例软件.

如果图片过小,请点击查看原图.

涉及到的技术:JavaScript & E

本文章只发布于精易论坛(bbs.125.la),谢绝新手盗转据为己有.

目录:

1 简要介绍

2 生成算法简介

3 如何使用?

4 关于样例软件的一些说明

1 简要介绍

一般来说我们做开发,拿到了JSON数据自然是很容易解析的,那么此时就会衍生出一个需求:JSON对象对应的数据结构我们要自己定义.这样才能一一映射到对象中.

那么我们如果有一个如下所示的JSON结果:

{"retCode":"0","message":"成功.","data":{"token":"QNF124BID823NFUCBA=="}}

这是一个简短的请求返回,我们可以选择手动来编写:

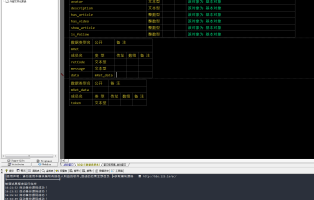


图 1-1

这是一个很简单的组装数据.

那么有时候遇到很庞大的数据的时候呢?



图 1-2

该数据返回大小为64.7kb,在github源码中已提供该simple.

总不能一个个手动复制写吧?很累.那么此时该算法应运而生.

2 生成算法浅析

首先看图

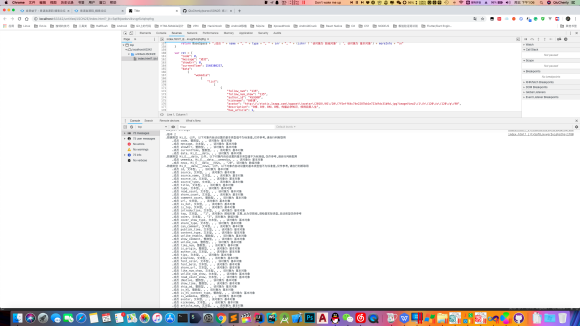


图 2-1

可以看到很熟悉的易语言自定义数据类型源码.

我为了最大化利用语言特性,使用JavaScript,原生支持读写JSON,易语言内嵌调用.

关键代码(全部代码见帖子底部下载地址):

*/\*\*  
 \* 生成单独的完整数据类型  
 \** ***@param*** *tests 测试的完整 json 对象  
 \** ***@param*** *className 默认生成的易语言顶级数据类型名称  
 \** ***@return*** *返回一个完整的数据类型数据  
 \*/***function** getAll(tests, className) {  
 **var** BaseInfo = '.数据类型 ' + className + ', 公开, 以下对象内自动设置的基本类型值不为标准值,仅作参考,请自行判断取用\n';  
 **for** (**var** theObjName **in** tests) {  
 **if** (tests.hasOwnProperty(theObjName)) {  
 **var** valueType = getObjTypeName(tests[theObjName]);  
 //console.log(a + "|" + valueType);  
 **var** newClsName = className + BaseAdditionSymbols + theObjName;  
 **if** (isObj(valueType)) {  
 //单独对象的处理  
 BaseInfo += getChild(theObjName, newClsName);  
 newClass.push(getAll(tests[theObjName], newClsName))  
 } **else if** (isObjArr(valueType)) {  
 //对象数组的递归处理  
 //有时候数组会为[]这种形式，此时无法确定具体的类型,默认设为基本数据类型 文本型 数组  
 **var** isBaseTypeArr = checkAllType(tests[theObjName]);  
 **var** arrSize = tests[theObjName].length;  
 **if** (isBaseTypeArr) {  
 newClsName = getTypeName(getObjTypeName(tests[theObjName][0]));  
 } **else** {  
 newClass.push(getAll(tests[theObjName][0], newClsName))  
 }  
 BaseInfo += getChild(theObjName, newClsName, **true**, arrSize, ((arrSize <= 0) ? " 注意,此为空数组,请检查实际类型,自动类型仅供参考" : ""));  
 } **else** {  
 //其他基本类型处理  
 BaseInfo += getChild(theObjName, getTypeName(valueType));  
 }  
 }  
 }  
 **return** BaseInfo;  
}

代码中所有方法全部引用注释方式:



图 2-2

你也许注意到了,生成的代码中字段是以' \_\_\_' 三个下划线来分隔.原因是因为有些JSON数据返回的子字段中以' info\_Base'该种形式存在,这会导致将来生成一键取JSON值算法带来不便.

是的,不久的将来我将实现一键生成读取JSON数据结构的方法,生成一个子程序,提供HTTP返回的JSON文件即可直接转换到易语言数据类型,开发人员只需要根据返回的数据类型来读取需要的数据即可,从而拒绝手工编写像这样的代码:



图 2-3

尽力再进一步简化开发难度.

关于 调试 此项目 代码:

**输出信息按照console标准执行,请勿使用document.writeln方法.**

关于contribute code:

**如果你有更好的想法或者有对该源码有改进的建议,请将改进后的代码Pull Request给项目master分支.(禁止强制推送)**

3 如何使用?

我提供了一个使用样例软件,如**[简要介绍]**一节中**图1-2**所示.

该源码实现了调用案例.

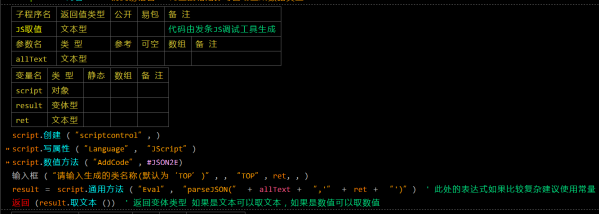


图 3-1

使用该方法即可获得JSON对象的易语言对应数据类型.

将JSON数据粘贴到编辑框中



图 3-2

单击:按钮即可生成数据.

期间会提示你提供一个类名:



图 3-2 输入易语言生成的类型名称

粘贴到IDE任意位置即可生成:

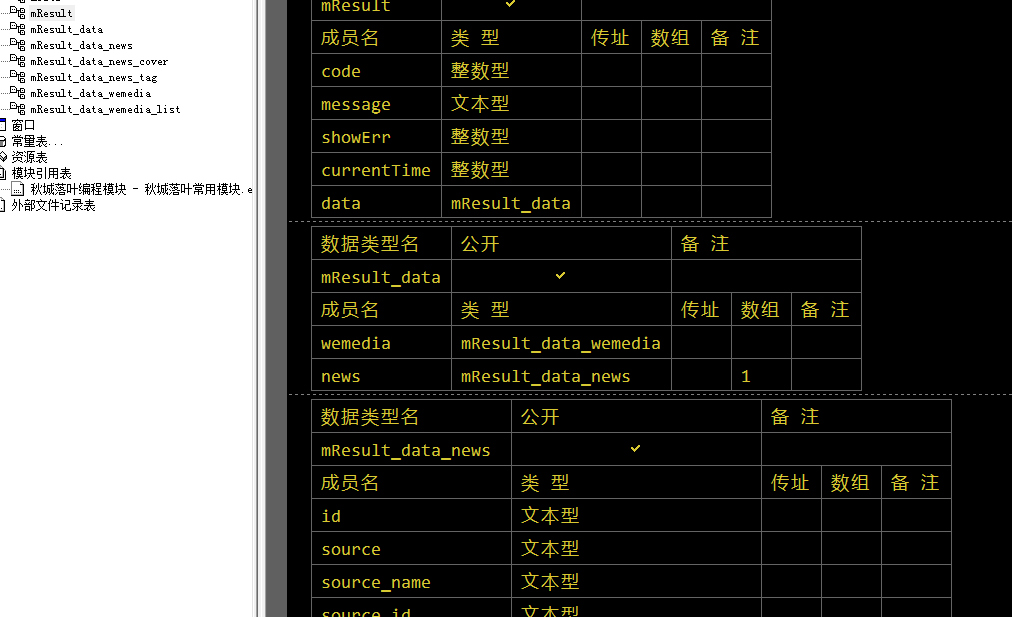


图 3-3

4 关于样例软件

你也许可能注意到了,我提供的软件还有更多的功能.

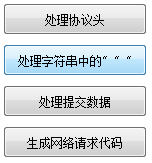


图 4-1

     4.1 处理协议头

关于本功能,实现逻辑请参阅工具源代码.

功能:将协议头生成为易语言风格.

演示:



图 4-1-1

将此文本复制到工具的编辑框中(此过程不再赘述)

你还会注意到:该选项.

该选项如果选中,则会自动去掉Cookie一栏数据与Accept-Encoding的GZIP一栏数据.



图 4-1-2

点击按钮后粘贴到IDE内:



图 4-1-3

4.2 处理字符串中的”””号



功能:将任意字符串中的”号修改为#引号.

演示:

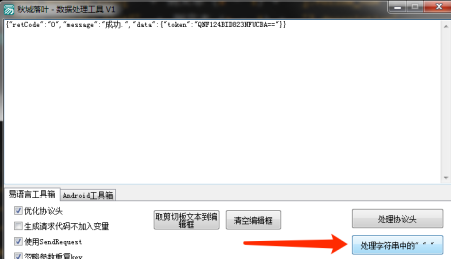


图 4-2-1

将会生成如下数据:

“{” + #引号 + “retCode” + #引号 + “:” + #引号 + “0” + #引号 + “,” + #引号 + “message” + #引号 + “:” + #引号 + “成功.” + #引号 + “,” + #引号 + “data” + #引号 + “:{” + #引号 + “token” + #引号 + “:” + #引号 + “QNF124BID823NFUCBA==” + #引号 + “}}”

4.3 处理提交数据(有解析BUG)

功能:将POST提交的文本数据序列化为易语言代码,该功能支持POST的提交参数解析.(不完美)

演示:



图 4-3-1

生成:

"https://translate.google.cn/translate\_a/single?client="＋"webapp"＋"&sl="＋"en"＋"&tl="＋"zh-CN"＋"&hl="＋"zh-CN"＋"&dt="＋"at"＋"&dt="＋"bd"＋"&dt="＋"ex"＋"&dt="＋"ld"＋"&dt="＋"md"＋"&dt="＋"qca"＋"&dt="＋"rw"＋"&dt="＋"rm"＋"&dt="＋"ss"＋"&dt="＋"t"＋"&source="＋"btn"＋"&ssel="＋"0"＋"&tsel="＋"0"＋"&kc="＋"0"＋"&tk="＋"209572.369339"＋"&q="＋"columns"

4.4 生成网络请求代码



功能:将抓包抓到的数据一键解析为易语言代码.支持Fiddler与Charles.

演示:



图 4-4-1



图 4-4-2

你会注意到,与本功能关联的是:.建议取消勾选,使用精易的网页\_访问\_对象.

新加你一个子程序,在子程序下粘贴生成(代码生成仅供参考,请自行检查代码是否补全.):



图 4-4-3

注1:parseJSON2E项目已在github开源,欢迎贡献你的idea.如果你有更好的代码贡献,请自行commit pull request.

注2:地址:https://github.com/QiuChenly/parseJSON2E

注3:如果你想联系我,那我在这个群里.(注:我并非此群群主或管理,与群主或管理并无往来.)

