JSon 封装及解析

qt 下来封装 JSon

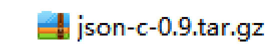
```
void MyQQ::slot_LoginCommit(QString name,QString password)
{
    std::string strTmp = name.toStdString();
    char* namebuf = (char*)strTmp.c_str();
    strTmp =password.toStdString();
    char* passwordbuf = (char*)strTmp.c_str();

   //构建 Json 对象
    QJsonObject json;
    json.insert("type", DEF_PACK_LOGIN_RQ);
    json.insert("name", namebuf );
    json.insert("password", passwordbuf);

    // 构建 Json 文档
    QJsonDocument document;
    document.setObject(json);
    QByteArray bt = document.toJson(QJsonDocument::Compact);
    qDebug() << bt;
    m_client->SendData( bt.data() , bt.size() );
}
```

linux 下解析 Json

[GitHub - json-c/json-c: https://github.com/json-c/json-c is the official code repository for json-c. See the wiki for release tarballs for download. API docs at http://json-c.github.io/json-c/]

可以在官方下载安装包.

下载完数据包



json-c-0.9.tar.gz

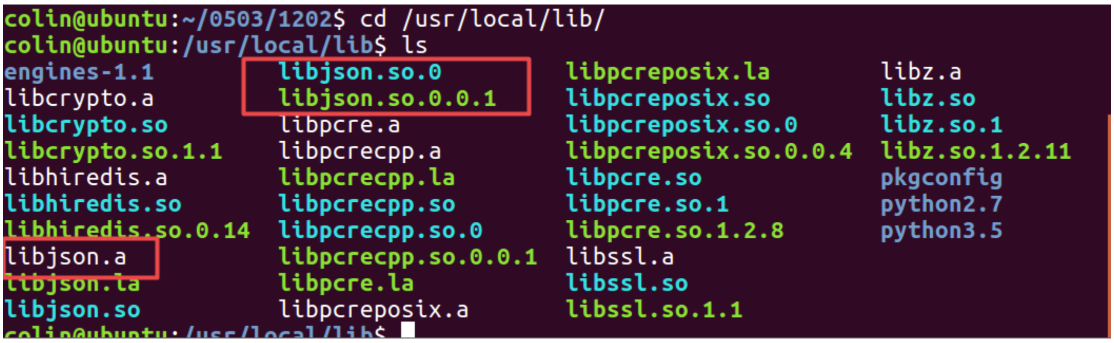
然后，将其解压并安装

sudo su → password: colin123

```
sudo cp   json-c-0.9.tar.gz     /usr/local
cd /usr/local
sudo tar -xvf   json-c-0.9.tar.gz
cd   json-c-0.9
```

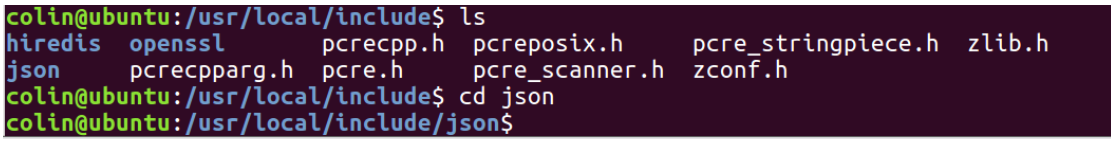
```
./configure
make
sudo make install
```

.so 文件 安装在 /usr/local/lib



```
colin@ubuntu:~/0503/1202$ cd /usr/local/lib/
colin@ubuntu:/usr/local/lib$ ls
engines-1.1          libjson.so.0         libpcreposix.la        libz.a
libcrypto.a          libjson.so.0.0.1     libpcreposix.so        libz.so
libcrypto.so         libpcre.a            libpcreposix.so.0      libz.so.1
libcrypto.so.1.1     libpcrecpp.a         libpcreposix.so.0.0.4  libz.so.1.2.11
libhiredis.a         libpcrecpp.la        libpcre.so             pkgconfig
libhiredis.so        libpcrecpp.so        libpcre.so.1           python2.7
libhiredis.so.0.14   libpcrecpp.so.0      libpcre.so.1.2.8       python3.5
libjson.a            libpcrecpp.so.0.0.1  libssl.a
libjson.la           libpcre.la           libssl.so
libjson.so           libpcreposix.a       libssl.so.1.1
colin@ubuntu:/usr/local/lib$
```

头文件在 /usr/local/include/json/ 下面



```
colin@ubuntu:/usr/local/include$ ls
hiredis  openssl       pcrecpp.h  pcreposix.h     pcre_stringpiece.h  zlib.h
json     pcrecpparg.h  pcre.h     pcre_scanner.h  zconf.h
colin@ubuntu:/usr/local/include$ cd json
colin@ubuntu:/usr/local/include/json$
```

运行时添加库

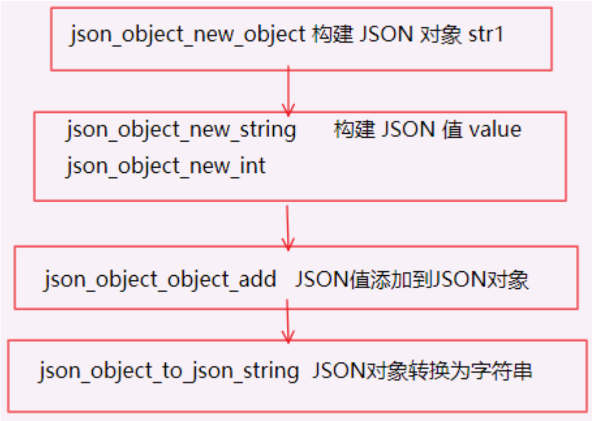
```
LIBS += -L/usr/local/lib/   -ljson
INCLUDEPATH += /usr/local/include/json/
```

测试代码如下:

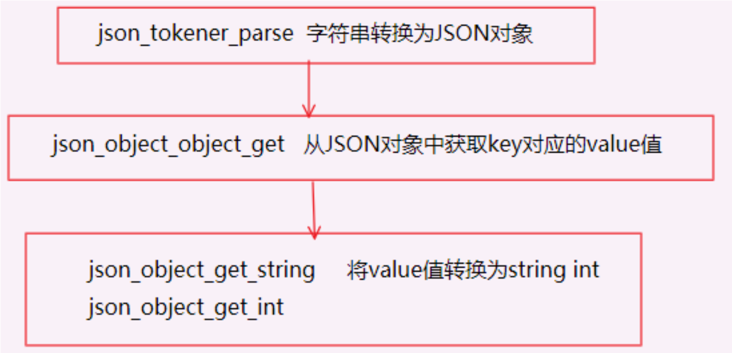
```
#include <stdio.h>
#include "json.h"
/*
实现把点菜的信息以 json 打包与解析:
{"name":"烤土豆"，“price”:20 ,"weight:300"}
*/
int main(int argc, char const *argv[])
{
    //1、创建两个字符串对象，可以以理解为小容器{ }
    struct json_object *str1=json_object_new_object();

    //2、把要存放的数据转为对象
    struct json_object *value1=json_object_new_string("烤土豆");
    struct json_object *value2=json_object_new_int(20);
    struct json_object *value3=json_object_new_int(300);

    //3、把数值对象添加到字符串对象中
    json_object_object_add(str1,"name", value1);
    json_object_object_add(str1,"price", value2);
    json_object_object_add(str1,"weight", value3);

    //4、把数组对象转为字符流进行发送
    const char *temp=json_object_to_json_string(str1);

    printf("%s\n",temp);

    //1、把得到的字符流转为数组对象
    str1=json_tokener_parse(temp);

    //2、在字符串 {} 对象中根据 key 值找到数值对象
    value1=json_object_object_get(str1,"name");
    value2=json_object_object_get(str1,"price");
    value3=json_object_object_get(str1,"weight");

    //3、数值对象转换为对应的数值
    temp=json_object_get_string(value1);
    int n=json_object_get_int(value2);
    int m=json_object_get_int(value3);
    printf("%s : %d : %d\n",temp,n ,m);

    //释放空间
    json_object_put(str1);
    json_object_put(value1);
    json_object_put(value2);
    json_object_put(value3);

    return 0;
}
```

输出 ： { "name": "烤土豆", "price": 20, "weight": 300 }
烤土豆 ：20 : 300

JSON 封装


json_object_new_object 构建 JSON 对象 str1
json_object_new_string 构建 JSON 值 value
json_object_new_int
json_object_object_add JSON值添加到JSON对象
json_object_to_json_string JSON对象转换为字符串

JSON 解析



json_tokener_parse 字符串转换为JSON对象

json_object_object_get 从JSON对象中获取key对应的value值

json_object_get_string 将value值转换为string int

json_object_get_int