## cJSON的构造和解析

    对于cJSON的使用，我主要是用来模拟远程服务器端返回的一个json类型的目录结构，客户端进行获取并进行解析，把解析出来的目录按照原本的结构显示在本地。

cJSON是一个超轻巧，携带方便，单文件，简单的可以作为ANSI-C标准的JSON解析器。

进入cJSON.h头文件中可以查看cJSON的相关信息。主要包括：cJSON结构体、cJSON类型、cJSON的一些内部的函数等。

// cJSON结构体:

typedef struct cJSON {

     struct cJSON \*next,\*prev;   // next/prev allow you to walk array/object chains. Alternatively, use GetArraySize/GetArrayItem/GetObjectItem

     struct cJSON \*child;        // An array or object item will have a child pointer pointing to a chain of the items in the array/object.

     int type;                   // The type of the item, as above.

     char \*valuestring;          // The item's string, if type==cJSON\_String

     int valueint;               // The item's number, if type==cJSON\_Number

     double valuedouble;         // The item's number, if type==cJSON\_Number

     char \*string;               // The item's name string, if this item is the child of, or is in the list of subitems of an object.

} cJSON;

// cJSON 类型:

#define cJSON\_False 0

#define cJSON\_True 1

#define cJSON\_NULL 2

#define cJSON\_Number 3

#define cJSON\_String 4

#define cJSON\_Array 5

#define cJSON\_Object 6

用法：

1、需要包含cJSON.h头文件，然后和cJSON.c或库文件libcJSON.a一起编译即可使用。

2、具体函数用法详见cJSON.h中的注释

更多介绍机器使用请参考：<http://sourceforge.net/projects/cjson/>.

# cJSON构造与解析json结构体

基本代码如下：

#include

#include

#include

#include "cJSON.h"

char \* create1()

{

     cJSON \*root,\*dir1,\*dir2,\*dir3;

     char \*out;

     //创建json数组型结构体

     root = cJSON\_CreateArray();

     //为数组添加对象

     cJSON\_AddItemToArray(root,dir1=cJSON\_CreateObject());

     //为对象添加字符串键值对

     cJSON\_AddStringToObject(dir1,"name",".");

     cJSON\_AddStringToObject(dir1,"path","uploads/");

     cJSON\_AddStringToObject(dir1,"flag","true");

     cJSON\_AddItemToArray(root,dir2=cJSON\_CreateObject());

     cJSON\_AddStringToObject(dir2,"name","..");

     cJSON\_AddStringToObject(dir2,"path","uploads");

     cJSON\_AddStringToObject(dir2,"flag","true");

     cJSON\_AddItemToArray(root,dir3=cJSON\_CreateObject());

     cJSON\_AddStringToObject(dir3,"name","wang.txt");

     cJSON\_AddStringToObject(dir3,"path","uploads/wang.txt");

     cJSON\_AddStringToObject(dir3,"flag","false");

     //将json结构体转换为字符串

     out=cJSON\_Print(root);

     //删除

     cJSON\_Delete(root);

     return out;

}

char \* create2()

{

     cJSON \*root,\*dir,\*child,\*subdir,\*dir1,\*dir2,\*dir3;

     char \*out;

     root=cJSON\_CreateObject();

     cJSON\_AddItemToObject(root,"Root",dir=cJSON\_CreateObject());

     cJSON\_AddStringToObject(dir,"name","/");

     cJSON\_AddStringToObject(dir,"path","/");

     cJSON\_AddStringToObject(dir,"flag","true");

     cJSON\_AddItemToObject(root,"Child",subdir = cJSON\_CreateArray());

     cJSON\_AddItemToObject(subdir,"dira",dir1=cJSON\_CreateObject());

     cJSON\_AddStringToObject(dir1,"name",".");

     cJSON\_AddStringToObject(dir1,"path","/./");

     cJSON\_AddStringToObject(dir1,"flag","true");

     cJSON\_AddItemToObject(subdir,"dira",dir2=cJSON\_CreateObject());

     cJSON\_AddStringToObject(dir2,"name","..");

     cJSON\_AddStringToObject(dir2,"path","/../");

     cJSON\_AddStringToObject(dir2,"flag","true");

     cJSON\_AddItemToObject(subdir,"dira",dir3=cJSON\_CreateObject());

     cJSON\_AddStringToObject(dir3,"name","uploads");

     cJSON\_AddStringToObject(dir3,"path","/uploads/");

     cJSON\_AddStringToObject(dir3,"flag","true");

     out=cJSON\_Print(root);

     cJSON\_Delete(root);

     return out;

}

char \* create3()

{

     cJSON \*root,\*img,\*thm;

     char \*out;

     int nums[4]={100,200,300,400};

     root=cJSON\_CreateObject();

     cJSON\_AddItemToObject(root, "Root", img=cJSON\_CreateObject());

     cJSON\_AddNumberToObject(img,"key",800);

     cJSON\_AddNumberToObject(img,"value",600);

     cJSON\_AddStringToObject(img,"Title","Sugon");

     cJSON\_AddItemToObject(img,"child",thm=cJSON\_CreateObject());

     cJSON\_AddNumberToObject(thm,"key",125);

     cJSON\_AddStringToObject(thm,"value","100");

         cJSON\_AddStringToObject(thm,"Url","www.sugon.com");

     cJSON\_AddItemToObject(img,"nums", cJSON\_CreateIntArray(nums,4));

     out=cJSON\_Print(root);

     cJSON\_Delete(root);

     return out;

}

char \* create4()

{

     cJSON \*root,\*dir1,\*dir2;

     char \*out;

     const char \*ro = "Root";

     root=cJSON\_CreateObject();

     cJSON\_AddItemToObject(root,ro,dir1=cJSON\_CreateArray());

     cJSON\_AddNumberToObject(dir1,"key",800);

     cJSON\_AddNumberToObject(dir1,"value",600);

     cJSON\_AddStringToObject(dir1,"Title","key and value");

     cJSON\_AddItemToObject(root,ro,dir2=cJSON\_CreateArray());

     cJSON\_AddNumberToObject(dir2,"value",125);

     cJSON\_AddStringToObject(dir2,"key","100");

     cJSON\_AddStringToObject(dir2,"Title","value and key");

     out=cJSON\_Print(root);

     cJSON\_Delete(root);

     return out;

}

void parse1(char \*out)

{

     cJSON \* root,\*arrayItem,\*item,\*name,\*path,\*flag;

     int i = 0,size = 0;

     char \*pr = NULL,\*na = NULL,\*pa = NULL,\*fl = NULL;

     //将字符串解析成json结构体

     root = cJSON\_Parse(out);

     //根据结构体获取数组大小

     size = cJSON\_GetArraySize(root);

     //printf("%d\n",size);

     //遍历数组

     for(i=0;i

     {

         //获取第i个数组项

         arrayItem = cJSON\_GetArrayItem(root,i);

         if(arrayItem)

         {

              //printf("%s\n","start......");

              //讲json结构体转换成字符串

              pr = cJSON\_Print(arrayItem);

              item = cJSON\_Parse(pr);

              name = cJSON\_GetObjectItem(item,"name");

              path = cJSON\_GetObjectItem(item,"path");

              flag = cJSON\_GetObjectItem(item,"flag");

              na = cJSON\_Print(name);

              pa = cJSON\_Print(path);

              fl = cJSON\_Print(flag);

              //printf("%s\n",pr);

              printf("name:%s\n",na);

              printf("path:%s\n",pa);

              printf("flag:%s\n\n",fl);

         }

     }

}

void parse2(char \*out)

{

     cJSON \* root,\*Root,\*Child,\*arrayItem,\*item,\*name,\*path,\*flag;

     int i = 0,size = 0;

     char \*pr = NULL,\*na = NULL,\*pa = NULL,\*fl = NULL;

     root = cJSON\_Parse(out);

     if(root)

     {

         Root = cJSON\_GetObjectItem(root,"Root");

         if(Root)

         {

              name = cJSON\_GetObjectItem(Root,"name");

              path = cJSON\_GetObjectItem(Root,"path");

              flag = cJSON\_GetObjectItem(Root,"flag");

              na = cJSON\_Print(name);

              pa = cJSON\_Print(path);

              fl = cJSON\_Print(flag);

              printf("Root:\n");

              printf("name:%s\n",na);

              printf("path:%s\n",pa);

              printf("flag:%s\n\n",fl);

         }

         Child = cJSON\_GetObjectItem(root,"Child");

         if(Child)

         {

              size = cJSON\_GetArraySize(Child);

              //printf("%d\n",size);

              printf("Child:\n");

              for(i=0;i

              {

                   arrayItem = cJSON\_GetArrayItem(Child,i);

                   if(arrayItem)

                   {

                       //printf("%s\n","start......");

                       pr = cJSON\_Print(arrayItem);

                       item = cJSON\_Parse(pr);

                       name = cJSON\_GetObjectItem(item,"name");

                       path = cJSON\_GetObjectItem(item,"path");

                       flag = cJSON\_GetObjectItem(item,"flag");

                       na = cJSON\_Print(name);

                       pa = cJSON\_Print(path);

                       fl = cJSON\_Print(flag);

                       //printf("%s\n",pr);

                       printf("name:%s\n",na);

                       printf("path:%s\n",pa);

                       printf("flag:%s\n\n",fl);

                   }

              }

         }

     }

}

int main()

{

     char \*out1 = create1();

     char \*out2 = create2();

     char \*out3 = create3();

     char \*out4 = create4();

     printf("%s\n\n\n",out1);

     parse1(out1);

     printf("%s\n\n\n",out2);

     parse2(out2);

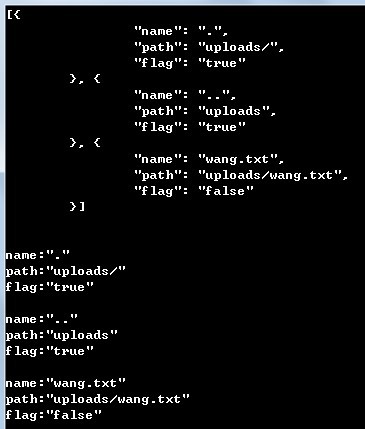
     printf("%s\n\n\n",out3);

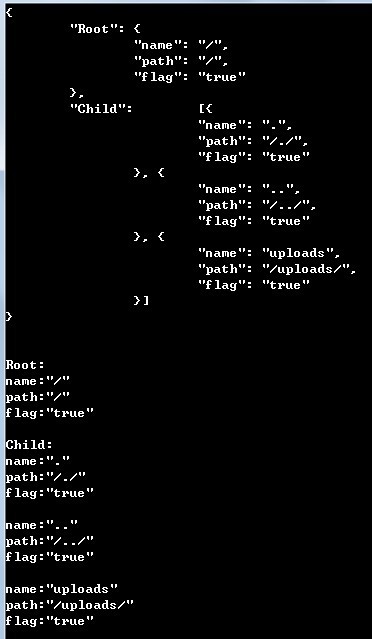
     printf("%s\n\n\n",out4);

     return 0;

}

运行结果如下图所示：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=a6fb6cc90101ffme&url=http://album.sina.com.cn/pic/a6fb6cc9gx6CmUB2G7a24)

[](http://photo.blog.sina.com.cn/showpic.html#blogid=a6fb6cc90101ffme&url=http://album.sina.com.cn/pic/a6fb6cc9gx6CmUCLQWj57)

上图为创建json结构体和解析后的结果图（分别为create1，parse1、create2，parse2），后两个（create3、create4）创建了没有进行解析，因为很简单，自己动手试试吧！相信自己，有些事情其实还是会很容易做到的。