

cJSON 库说明书

版本	日期	作者	说明
1.0	2016-7-19		1、cJSON

1. 主要功能

模块主要功能：

- 1、负责创建 ittiarn 模块所需 json 文件

2. JSON 文件优缺点

- 1、优点：

- A: 数据格式比较简单，易于读写，格式都是压缩的，占用带宽小；
- B: 支持多种语言，包括 `ActionScript`, `C`, `C#`, `ColdFusion`, `Java`, `JavaScript`, `Perl`, `PHP`, `Python`, `Ruby` 等服务器端语言，便于服务器端的解析；
- C: 易于解析，客户端 `JavaScript` 可以简单的通过 `eval()` 进行 `JSON` 数据的读取；
- D: 因为 `JSON` 格式能直接为服务器端代码使用，大大简化了服务器端和客户端的代码开发量，且完成任务不变，并且易于维护。

- 2、缺点：

- A: 没有 `XML` 格式这么推广的深入人心和喜用广泛，没有 `XML` 那么通用性
- B: `JSON` 格式目前在 `Web Service` 中推广还属于初级阶段

3. 输入

- 1、无

4. 输出

- 1、无

5. 目录结构

`Objs` 目录：存放编译时产生的 `.o` 等文件，是一个临时文件

`.c/.h` 文件：模块的源文件

`Makefile` 文件：编译所需文件

`Build.sh` 文件：一键编译文件，产生的静态库文件 `libcjson_x64.a` 将被放到上级目录的 `lib` 文件中

依赖库文件：无

6. 流程

一、cJSON 定义

1、cJSON 库基本数据格式 cJSON 定义

```
/* The cJSON structure: */
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 *valstring; /* 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;
```

图 1 cJSON 定义

二、构造 JSON 文件：

1、创建 JSON 对象

```
cJSON* root = cJSON_CreateObject();
```

2、创建数组对象

```
cJSON* array = cJSON_CreateArray();
```

3、向对象中增加（字符）节点

```
cJSON_AddStringToObject(root, "名称", "字符串");
```

4、向对象中增加对象（字符串、数组、数字）

```
cJSON_AddItemToObject(root, "名称", cJSON_CreateString("字符串"));
```

```
cJSON_AddItemToObject(root, "名称", cJSON_CreateNumber(数字));
```

5、向数组中增加元素；

```
cJSON_AddItemToArray(array, cJSON_CreateString("字符串"));
```

```
cJSON_AddItemToArray(array, cJSON_CreateNumber(数字));
```

三、解析 JSON 文件

1、首先获取字符串形式的 JSON 数据，例：存储于 char data[1024];

2、获取 JSON 对象

```
cJSON* root = cJSON_Parse(data);
```

3、获取指定名称的节点指针

```
cJSON* tmp=cJSON_GetObjectItem(root,"name");
```

4、解析数组，首先按上步获取数组指针 如 **cJSON* array**

```
获取数组大小 int size=cJSON_GetArraySize(array);获取数组元素 cJSON*  
tmp=cJSON_GetArrayItem(array,1);
```

5、释放对象

```
cJSON_Delete(root);
```

6、根据对象指针获取 JSON 字符串形式数据

```
char *data=cJSON_Print(root);
```

7. 鸣谢

感谢之前对这个模块有过贡献的同事们。