

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

[Tell me more](#) ×

Using cJSON to read in a JSON array

CAREERS 2.0
by stackoverflow



+



Have projects on Codeplex?
Import them easily to your profile

First off, this is a very broad question, and it might come across as me asking for the community to write my code for me. That is not my intent, but I am so lost, I don't know how to give enough information.

I am attempting to use the cJSON library, written by Dave Gamble, to read in the following JSON array:

```
"items":  
[  
  {  
    "name": "command",  
    "index": "X",  
    "optional": "0"  
  },  
  {  
    "name": "status",  
    "index": "X",  
    "optional": "0"  
  }  
]
```

From reading his [documentation](#), I found ways to read in individual Objects, but nothing regarding Arrays, and I wasn't able to surmise how to do it from the examples given.

Here's what I'm trying:

```
cJSON* request_json = NULL;  
cJSON* items = cJSON_CreateArray();  
cJSON* name = NULL;  
cJSON* index = NULL;  
cJSON* optional = NULL;  
  
request_json = cJSON_Parse(request_body);  
  
items = cJSON_GetObjectItem(request_json, "items");  
  
name = cJSON_GetObjectItem(items, "name");  
index = cJSON_GetObjectItem(items, "index");  
optional = cJSON_GetObjectItem(items, "optional");
```

I know this is wrong, and not just because it's not working, but I can't figure out how to make it right.

Obviously I'm going to need to loop the process of reading in all of the entries for each index of the array. I have no idea how I'm going to do that though, because I don't know where I should be using the

indexes in this code, or if it is even the right start. There is a `cJSON_GetArrayItem()`, but it takes only a number (presumably an index) and no string to indicate which field it wants.

Please ask for more information as needed, and I will make edits to the question. I in no way expect an answer right away. Thank you.

c | json

asked Jun 3 at 15:55



Nealon

251 1 13

If the value of "items" is an array, it seems to me you should be indexing an array in there somewhere. Then that will give you back an "object" which you can then search by key. – [Hot Licks](#) Jun 3 at 16:01

(Note that you erroneously omitted the surrounding `{}` when you quoted the JSON. The `{}` denotes an "object", and without it the above text is invalid JSON.) – [Hot Licks](#) Jun 3 at 16:05

That wasn't how it was defined in the compiler, just showing it as an example – [Nealon](#) Jun 3 at 16:32

It's important to understand the syntax of JSON. That's the key to reading it. – [Hot Licks](#) Jun 3 at 16:35

yeah, i get that, again, I was just quoting the text for people reading to understand it – [Nealon](#) Jun 3 at 16:39

2 Answers

Document mentions about `parse_object()`.

I think this is what you need to do.

```
void parse_object(cJSON *root)
{
    cJSON* name = NULL;
    cJSON* index = NULL;
    cJSON* optional = NULL;

    int i;

    cJSON *item = cJSON_GetObjectItem(items,"items");
    for (i = 0 ; i < cJSON_GetArraySize(item) ; i++)
    {
        cJSON * subitem = cJSON_GetArrayItem(item, i);
        name = cJSON_GetObjectItem(subitem, "name");
        index = cJSON_GetObjectItem(subitem, "index");
        optional = cJSON_GetObjectItem(subitem, "optional");
    }
}
```

Call this function as

```
request_json = cJSON_Parse(request_body);
parse_object(request_json);
```

edited Jun 3 at 16:29

answered Jun 3 at 16:20



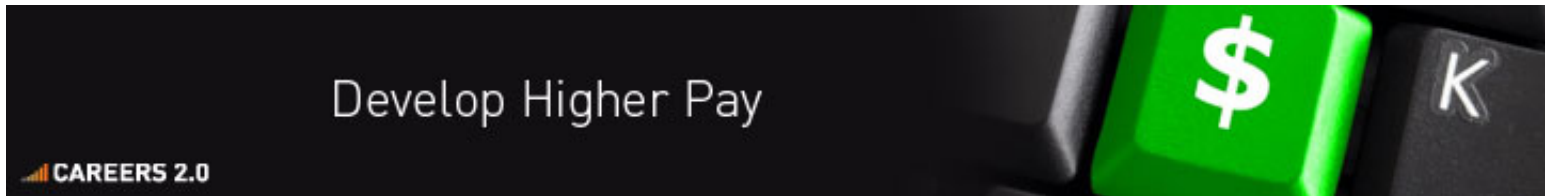
Denny Mathew
112 1 5

looks solid to me, I'll give it a shot, thanks. — Nealon Jun 3 at 16:35

you are welcome... :) — Denny Mathew Jun 3 at 16:36

worked perfectly, you're the man. — Nealon Jun 3 at 17:24

SIDE NOTE: Storing values in a structure worked best for me. — Nealon Jun 3 at 17:46



My guess (not having read the spec, and being a bit rusty with C):

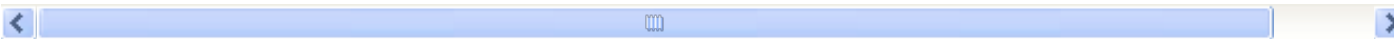
```
request_json = cJSON_Parse(request_body);

items = cJSON_GetObjectItem(request_json, "items");
for (int i = 0; i < max; i++) { // Presumably "max" can be derived from "items" s

    cJSON* item = cJSON_GetArrayItem(items, i);

    name = cJSON_GetObjectItem(item, "name");
    index = cJSON_GetObjectItem(item, "index");
    optional = cJSON_GetObjectItem(item, "optional");

    // Stash above info somewhere
}
```



answered Jun 3 at 16:27



Hot Licks

20.6k 4 27 66

Not the answer you're looking for? Browse other questions tagged [c](#) [json](#) or [ask your own question](#).