

Bài 11_ JSON

11.1. Giới thiệu về cấu trúc dữ liệu JSON

- JSON là viết tắt của "JavaScript Object Notation" (Ghi chú về Đối tượng JavaScript). Đây là một định dạng truyền tải dữ liệu phổ biến trong lập trình và giao tiếp giữa các máy chủ và trình duyệt web, cũng như giữa các hệ thống khác nhau.
- JSON rất tuyệt vời dùng cho các dạng dữ liệu phức tạp, hỗn hợp nhiều kiểu dữ liệu khác nhau.
- JSON sử dụng một cú pháp nhẹ dựa trên cặp key - value, tương tự như các đối tượng và mảng trong JavaScript. Mỗi đối tượng JSON bao gồm một tập hợp các cặp "key" và "value", trong khi mỗi mảng JSON là một tập hợp các giá trị.
- Để phân tích (parse) được dữ liệu JSON (đọc, ghi, hiển thị) trong C, thì có 2 cách. Một là chuyển chuỗi JSON thành chuỗi ký tự và phân tích từng ký tự 1 để phân tách ra đâu là key, value, chuỗi string, chuỗi số, boolean,...
- Hai là ...

JSON có 2 kiểu dữ liệu chính là Object và mảng Array

Object:

Gồm các cặp keys : values, keys là 1 chuỗi string, còn value là bất cứ kiểu dữ liệu này.

Đặc điểm:

- Object đặt trong dấu {},
- (Keys, Values) ngăn cách bởi dấu “:”, mỗi cặp ngăn cách bởi dấu “,”

```
{  
  "name": "John Doe",  
  "age": 30.1234,  
  "city": "New York",  
  "isStudent": true,  
  "grades": [85, 90, 78]  
}
```

JSON kiểu mảng array

Đặc Điểm:

- array nằm trong dấu ngoặc vuông []
- Phần tử của mảng là các object và bất kỳ kiểu dữ liệu nào khác.

```
[  
  {  
    "name": "John Doe",  
    "age": 30,  
    "city": "New York",  
    "occupation": "Software Engineer",  
    "isStudent": false  
  },  
  {  
    "name": "Jane Smith",  
    "age": null,  
    "city": "Los Angeles",  
    "contact": {  
      "email": "jane.smith@example.com",  
      "phone": "555-1234"  
    }  
  }  
],
```

```
{
  "name": "Bob Johnson",
  "age": 35,
  "city": "Chicago"
},
20, 3.14, "Hello word", true, null
]
```

Ưu điểm của JSON so với Struct:

- Struct chỉ dùng khi biết rõ số lượng thành viên
- JSON linh hoạt hơn dùng cho các dữ liệu phức tạp hơn. Ví dụ: Khi đối tượng là Sinh viên chẳng hạn, có nhưng thông tin bắt buộc điền, thông tin không bắt buộc. hoặc SV1 có 3 thông tin, SV2 có 5 thông tin, SV3 có 6 thông tin chẳng hạn. thì phải dùng kiểu JSON

```
I- Tên*
- Tuổi:
- SDT*:
- Địa chỉ*:
- Nghề nghiệp:
```

11.2 Cách xử lý dữ liệu JSON

Khi chuyển dữ liệu Json vào C để phân tích, đọc, ghi, hiển thị:

biện pháp phân tích các kiểu dữ liệu JSON là ta biến tất cả thành chuỗi trong C, và phân tích từng ký tự để bóc tách từng kiểu dữ liệu.

Việc phân tích qua 2 bước :

- 1. Phân tích thô => khi duyệt đến ký tự mà bạn nghi ngờ nó là bắt đầu kiểu dữ liệu nào đó như ký tự n = null, t, f: true, false => boolean, \ : bắt đầu string, [: bắt đầu array, { : bắt đầu object
return : Lập tức trả về 1 hàm phân tích chuyên sâu parse_null, parse_boolean, parse_string, parse_array, parse_object tương ứng
- 2. Hàm phân tích chuyên sâu có nhiệm vụ phân tích và (return) trả giá trị về biến con trỏ (có nghĩa bạn có nhiệm vụ gán giá trị cho các thành viên, phần tử trong con trỏ đó)

Để chuyển chuỗi JSON vào C phân tích có 2 cách :

Cách 1: Dùng hàm đọc file json.

```

628
629 int main(int argc, char const *argv[])
630 {
631     /* Đọc file JSON */
632     const char *json = read_json_file("data.json");
633     if (!json) return 1;
634
635     /* Phân tích cú pháp chuỗi JSON */
636     JSONValue* json_value = parse_json(&json);
637
638     /* Kiểm tra kết quả phân tích cú pháp */
639     test(json_value);
640
641     /* Khởi tạo cấu trúc lưu dữ liệu JSON */
642     DeviceConfig config = {0};
643
644     /* Trích xuất dữ liệu JSON vào cấu trúc */
645     extract_device_config(json_value, &config);
646
647     /* Hiển thị thông tin */
648     print_device_config(config);
649
650     /* Giải phóng bộ nhớ được cấp phát */
651     free_json_value(json_value);
652
653     return 0;
654 }

```

Cách 2: copy thẳng chuỗi json vào C.

Do trong chuỗi Json có nhiều chuỗi con khác nên các chuỗi con phải đặt trong dấu \“ sub-string\” thì ngôn ngữ C mới hiểu được.

Ví dụ:

```

// Chuỗi JSON đầu vào
const char* json_str = "{
    \"1001\":{
        \"SoPhong\":3,
        \"NguoiThue\":{
            \"Ten\": \"Nguyen Van A\",
            \"CCCD\": \"1920517781\",
            \"Tuoi\":26,
            \"ThuongTru\":{
                \"Duong\": \"73 Ba Huyen Thanh Quan\",
                \"Phuong_Xa\": \"Phuong 6\",
                \"Tinh_TP\": \"Ho Chi Minh\"
            }
        },
        \"SoNguoiO\":{
            \"1\": \"Nguyen Van A\",
            \"2\": \"Nguyen Van B\",
            \"3\": \"Nguyen Van C\"
        },
        \"TienDien\": [24, 56, 98],
        \"TienNuoc\": 30.000
    }
}

```

```
    }, "
    \"1002\": {
        \"SoPhong\": 5,
        \"NguoiThue\": {
            \"Ten\": \"Phan Hoang Trung\",
            \"CCCD\": \"012345678912\",
            \"Tuoi\": 24,
            \"ThuongTru\": {
                \"Duong\": \"53 Le Dai Hanh\",
                \"Phuong_Xa\": \"Phuong 11\",
                \"Tinh_TP\": \"Ho Chi Minh\"
            }
        }
    }, "
    \"SoNguoiO\": {
        \"1\": \"Phan Van Nhat\",
        \"2\": \"Phan Van Nhi\",
        \"2\": \"Phan Van Tam\",
        \"3\": \"Phan Van Tu\"
    }, "
    \"TienDien\": 23.000,
    \"TienNuoc\": 40.000
    }
    }";
```