Pre- Request

#Nội dung

1. Script flow in postman
2. Data format: Json
3. Data types in Javascript
4. JSON object
5. Variables
6. Build Request Body

#Script flow in postman

Luồng sơ đồ

#Data format: Json

JSON (JavaScript Object Notation) là một format trao đổi dữ liệu tự nơi này sang nơi khác (vd từ

server <->client)

Json data được biểu diễn bởi cặp key-value trong đó:

* **Chuỗi JSON được bao lại bởi dấu ngoặc nhọn {}**
* **Keys: properties of the Object**
* **Values: value of each key**
  + **Object – {}**
  + **Array – []**
  + **String – "hello"**
  + **Number – 20**
  + **Boolean – true/false**
  + **null**

**No trailing comma**

* Ví dụ

{ "name": "John Doe", "age": 30, "address": { "street": "123 Main St", "city": "Anytown", "state": "CA" }, "hobbies": ["reading", "traveling"] }

Trong vd trên , dữ liệu Json có 4 keys: “name", “age", “address", “hobbies". Mỗi key có 1 value tương ứng , có thể là một string, 1 number, 1 object, hoặc 1 array.

* + “name” có value là 1 string: “John “
  + “ age” có value là 1 số” 30
  + “address” có value là 1 object :{"street": "123 Main St", "city": "Anytown", "state": "CA"}.
  + “hobbies” có value là 1 array: ["reading", "traveling"]

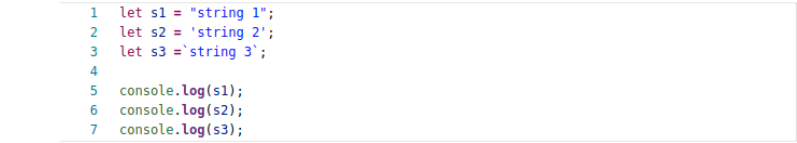
Trong ví dụ này, các keys nằm ở bên trái dấu hai chấm (:) và các value nằm ở bên phải. Các cặp khóa-giá trị được phân tách bằng dấu phẩy (,) và được đặt trong dấu ngoặc nhọn ({}).

#Data types in Javascript

* + Khai báo biến string

String: text between • Double quote: "" • Single quote: '' • Backtick character: ``

Vd cơ bản



Gọi biến bên trong

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

let s1 = "string 1";

let s2 = 'string 2';

let s3 = `${s1},${s2}`;

Methods của string

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

* + Khai báo biến number

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

//khai báo

let num1 = 100;

let num2 = 0,5;

let num3 = num1 + num2;

let num3 = num1 - num2;

let num3 = num1 \* num2;

let num3 = num1 / num2;

console.log(num1);

//biến number thành string

let s1 = num1.toString();

* + Khai báo biến Boolean

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

//khai báo

let boo1 = truse;

let boo2 = false;

//và

console.log(boo1&&boo2);

//hoặc

console.log(boo1||boo2);

//so sánh

console.log(5>200);

//bằng

console.log("string 1"==="string 2");

//khác

console.log("string 2" !=="string 2");

* + Object

ví dụ object dạng Json

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

let obj1= {

"data":{

"Name":"thuy",

"age":20,

"is\_admin": true

},

};

Get giá trị vào 1 object

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

let obj1= {

"data":{

"Name":"thuy",

"age":20,

"is\_admin": true

},

};

//lấy giá trị value của 1 key

cách 1

let person\_name = obj1.data.name;

cách 2

let age = obj1["data"]["age"];

cách 3

let is\_admin = obj1.data["is\_admin"];

console.log(person\_name);

console.log(age);

console.log(is\_admin);

// lấy data chỉ dùng 1 dòng

let {

data: { name, age, is\_admin }

} = obj1;

//thêm thông tin trong object

obj1.data.address ="Namdih";

obj1.address ="Namdinh";

//Sửa thông tin trong object

obj1.data.name ="Test";

//xóa thông tin trong object

delete obj.address;

//copy thông tin trong object(biến obj thành string, sau đó biến string thành obj)

let temp =JSON.stringify(obj1);

let obj2 = JSON.parse(temp);

* 1. Array
     1. ví dụ

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

//khai báo

let arr1 =[];

let arr2 = ["abb", 1, true, {a: "b}];

//duyệt mảng

arr2.forEach((item, index) =>{

console.log(index, item);

});

//thêm item trong array

Push -> đẩy vào đuôi

unshift -> đầy vào đâu

vd:

let arr = ["A", "B"];

arr.push("C", "D");

arr.unshift("E","F");

//remove trong array

pop ->remove ở đầu

shift ->remove ở đuôi

let arr = ["A", "B", "C", "D"];

let s1 = arr.pop();

let s2 = arr.shift();

//remove đoạn giữavà thay thế

arr.splice(1,2," test", "name");

//sửa trong array

let arr = ["A", "B", "C", "D"];

arr[0] ="test";

#JSON Obiect

JSON is a object help to convert: json <-> string

JSON.parse(text) ---> json

JSON.stringify(json) ---> string

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

let obj1= {

"data":{

"Name":"thuy",

"age":20,

"is\_admin": true

},

};

//biến thành string

let data = JSON.stringify(obj1);

console.log(data);

//parse từ string thành Json

let obj2= JSON.sparse(data);

console.log(obj2);

#JSON - Postman

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

//cú pháp biến response thành Json

pm.response.Json() ---> Json

//biến thành string

JSON.stringify(Json) ---.> string

Vd

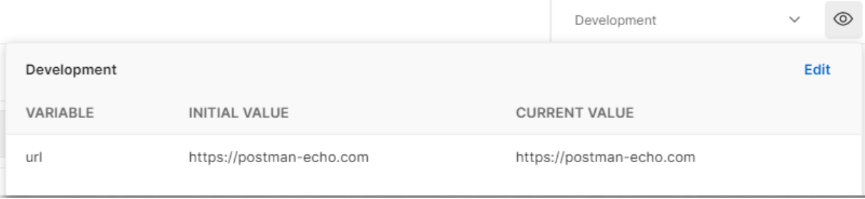
let res = pm.response.Json();

console.log(res);

consolo.log(JSON.stringify(res));

#Variables

Tạo biến



NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

vd tạo biến "name"

//Lấy 1 Biến từ 1 vị trí ra trong script

console.log(pm.globals.get("name"));

console.log(pm.collectionVariables.get("name"));

console.log(pm.enviroment.get("name"));

#BUILD REQUEST BODY

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

//lấy biến trong môi trường

let id = pm.environment.get("id");

let programs = ["IT", "Economy","Frinance","Enviroment"];

//lấy random 1 giá trị

let programme = \_.sample(programs);

//save thành biến environment

pm.environment.set("programme",programme);

Test

1. Nội dung

|  |
| --- |
| **– Extract value from: array, object with/ without condition** |
| **– Write tests: structure, style, keywords** |
| **– Reuse script** |
| **– Send request by code** |
| **– Branching and loop** |
| **– Data-driven test** |
| **– Use Newman as library** |

#Extract data

Get by index

NoneBashCSSCC#ElixirErlangGoGraphQLGroovyHaskellHCLHTMLINIJavaJavaScriptJSONJSXKotlinLispLuaMermaid DiagramNixObjective-COCamlPerlPHPPowershellPythonRubyRustScalaSQLSoliditySwiftTOMLTSXTypeScriptVisual BasicYAMLZig

vd 1 json sau

let res ={

"store": {

"book": [

{

"category":"reference",

"author": "Nigel Rees",

"title": "Sayings of the Century",

"price":8.95

},

{

"category":"fiction",

"author": "Evelyn Waugh",

"title": "Sword of Honnour",

"price":10.09

},

{

"category":"fiction",

"author": "Evelyn Waugh",

"title": "Sword of Honnour",

"price":22.99

}

]

}

}

//lấy giá trị đầu

let frirstAuthor = res.store.book[0].author;

cosole.log(firstAuthor);

//Lấy giá trị cuối cùng

let frirstAuthor = res.store.book[res.store.book.length -1].author;

cosole.log(firstAuthor);

//Lấy cả object

let firstBook = res.store.book[0];

cosole.log(firstBook);

//Lấy giá trị trong object. vd lấy giá trị title

let books =res.store.book;

let titlelist = [];

//dyệt từng thành phần trong mảng list book

books.forEach(element) => {

titleList.push(elemenr.title);//đẩy iteam vào 1 array

});

console.log(titlelist);

//Lấy từng thành phần trong mảng sử dụng hàm map

let books =new.store.book;

let categories =\_.map(books,\_.property("title"));

console.log(categories);

// lấy check điều kiện. vd lấy cuốn sách có price >10

let books =res.store.book;

let titlelist = [];

//dyệt từng thành phần trong mảng list book

books.forEach(element) => {

if (element.price >10) {

titleList.push(elemenr.title);//đẩy iteam vào 1 array

}

});

console.log(titlelist);

//lấy check nhiều điều kiện. vd lấy cuốn sách có price >10 và category ="fiction"

let books =res.store.book;

let titlelist = [];

//dyệt từng thành phần trong mảng list book

books.forEach(element) => {

if (element.price >10 && element.category === "fiction") {

titleList.push(elemenr.title);//đẩy iteam vào 1 array

}

});

console.log(titlelist);

Cách 2. Lấy giá trị kết hợp nhiều điều kiện dùng hàm lọc filter

let books =res.store.book;

let titlelist = \_.filter(books, (element) => {

return element.price >10 && element.category === "fiction";

});

let titles =\_.map(bookList, \_.property("title"));

console.log(titlelist);