

cronapi.json.Operations.getJsonOrMapField

Obter campo do Json

Obtém o valor do campo de um objeto JSON.

Parâmetros

Nome	Tipo	Doc
Parâmetro 1	Objeto JSON	Objeto a ser percorrido
Parâmetro 2	String	String representando o caminho a ser percorrido no objeto JSON

Retorno

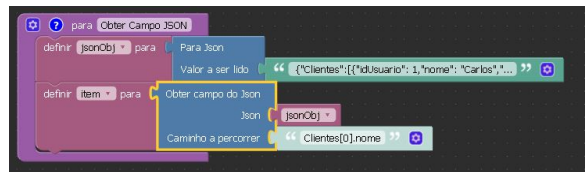
Retorna o conteúdo de um campo de um objeto JSON.

Compatibilidade

 Servidor

Exemplo 1

A função abaixo atribui a variável **item** o valor do campo **nome** do primeiro objeto do array **Cientes**.



Sobre JSON

JSON é um acrônimo para "JavaScript Object Notation". Um formato de padrão aberto que utiliza texto legível a humanos para transmitir objetos de dados consistindo de pares chave-valor. É o formato de dados mais comumente utilizado para comunicação assíncrona navegador/servidor.

Um JSON é construído com base em uma coleção de pares chave-valor (que definem as propriedades e seus valores), iniciado com "{" (chave aberta) e finalizado com "}" (chave fechada). Cada chave é seguida de ":" (dois pontos) e os pares são separados por "," (vírgula).

Exemplo

```
{
  "nome": "Pedro",
  "site": "cronApp.io"
}
```

É possível ainda criar arrays dentro de objetos JSON. Um array é iniciado por "[" (colchete aberto) e finalizado por "]" (colchete fechado). Cada valor do array é separado por "," (vírgula).

Exemplo

```
{
  "Clientes": [
    { "id": 1, "nome": "João" },
    { "id": 2, "nome": "Beatriz" }
  ]
}
```

Mais sobre



Arrays em JSON

É possível acessar os valores de um array dentro de um objeto JSON utilizando o **nome do array**, um número que representa o **índice** deste array e a **propriedade** que se deseja ter o valor retornado.

Ex.: O caminho **Usuarios[0].nome**, retorna o valor "Carlos".

Exemplo JSON

```
{
  "Usuarios": [
    { "idUsuario": 1, "nome": "Carlos" },
    { "idUsuario": 2, "nome": "Marcos" }
  ]
}
```

Obs.:



O primeiro elemento de um array JSON é representado pelo índice [0].