

Academia

Casos de uso

Anonymous · 08/10/2026

Tabla de contenidos

- Casos de uso 3
 - Casos genéricos 4
 - Añadir texto al resultado 10
 - Elegir valor a seleccionar 12

Academia » Leer datos de JSON con Selectores » Casos de uso

Casos de uso

Anonymous · 08/10/2026

Casos de uso

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos de uso » Casos genéricos

Casos genéricos

Casos genéricos

Anonymous · 08/10/2026

Casos genéricos

Los JSON, y por lo tanto índices y valores que se verán en los siguientes casos, son solo variables de ejemplo para mostrar el funcionamiento de los diferentes selectores y sus combinaciones. No son casos reales de órdenes de cierto comercio electrónico. Es importante que cada usuario, usando esta documentación, adapte el selector a los índices de su JSON.

Valor en índice de primer nivel

Si tenemos el siguiente JSON:

```
{
  "k_simple": "v_simple"
}
```

Para obtener el valor del índice `k_simple` usamos el siguiente selector:

```
k_simple
```

Valor en índice de segundo nivel (anidado)

Se utiliza un "." (punto) para ingresar a un nivel interior, o sea para entrar al objeto anidado.

Si tenemos el siguiente JSON:

```
{
  "k_nested1": {
    "k_nested2": "v_nested"
  }
}
```

Para obtener el valor del índice **k_nested2** usamos el siguiente selector:

```
k_nested1.k_nested2
```

Valor en índice de tercer nivel (anidado)

Este caso es idéntico al caso anterior. Se deja solo para ilustrar que se puede seguir agregando “.” para ingresar a tantos niveles como sea necesario según el JSON que tengamos.

Si tenemos el siguiente JSON:

```
{
  "k_nested1p": {
    "k_nested2p": {
      "k_nested3p": "v_nested"
    }
  }
}
```

Para obtener el valor del índice **k_nested3p** usamos el siguiente selector:

```
k_nested1p.k_nested2p.k_nested3p
```

Valor en índice de un arreglo

Si tenemos el siguiente JSON:

```
{
  "array": [1, 2, 3]
}
```

Para obtener el valor del índice **1 del arreglo** (segundo elemento, ya que los arreglos comienzan desde el índice 0) usamos el siguiente selector:

```
array[1]
```

Valor en índice de un arreglo anidado

Este caso es una combinación de 2 casos vistos previamente:

- Valor en índice de segundo nivel (anidado).
- Valor en índice de un arreglo.

Si tenemos el siguiente JSON:

```
{
  "nested_array": {
    "array": [1, 2, 3]
  }
}
```

Para obtener el valor del índice 1 del arreglo usamos el siguiente selector:

```
nested_array.array[1]
```

Valor en índice asociado a otro índice de un arreglo de diccionarios

Este es un caso bastante interesante, aquí tenemos un arreglo con diccionarios y necesitamos elegir un valor desde uno de los diccionarios pero no sabes cuál es su posición dentro del arreglo. Debido a lo anterior, la alternativa es buscar en los diccionarios por un “filtro” (valor conocido) y elegir otro índice dentro del mismo diccionario (diccionario filtrado) como el valor real que necesitamos obtener.

Se usarán 2 índices para este selector:

- Primer índice, lo llamaremos key, tiene el valor que estamos buscando para filtrar el diccionario que elegiremos del arreglo.
- Segundo índice, lo llamaremos value, tiene el valor real que estamos buscando obtener.

Si tenemos el siguiente JSON:

```
{
  "mixed": [
    {
      "key": 10,
      "value": "hola"
    },
    {
      "key": 20,
      "value": "mundo"
    },
    {
      "key": 30,
      "value": "chao"
    },
  ]
}
```

Para obtener el valor del índice **value** asociado al índice **key** que tiene como valor **20** usamos el siguiente selector:

```
mixed[key=20:value]
```

El formato de este selector es el siguiente:

```
array[key=filtro:value]
```

Donde:

- **array:** es el selector del arreglo que deseamos usar.
- **key:** es el índice que usaremos para hacer el filtro.
- **filtro:** es el valor que buscamos que tenga el índice key.
- **value:** es el valor asociado al índice key que queremos recuperar para usar.

Estos 4 campos irán cambiando según el JSON (al igual que todos los índices y valores de estos casos de ejemplo).

Valor en índice asociado a otro índice de un arreglo de diccionarios que está anidado

Este caso es una combinación de 2 casos vistos previamente:

- Valor en índice de segundo nivel (anidado).
- Valor en índice asociado a otro índice de un arreglo de diccionarios.

Si tenemos el siguiente JSON:

```
{
  "nested_mixed": {
    "mixed": [
      {
        "key": 10,
        "value": "hola"
      },
    ]
  }
}
```

Para obtener el valor del índice **value** asociado al índice **key** que tiene como valor **10** usamos el siguiente selector:

```
nested_mixed.mixed[key=10:value]
```

Valor en índice anidado dentro de un índice asociado a otro índice de un arreglo de diccionarios

Este caso es una combinación de 2 casos vistos previamente:

- Valor en índice asociado a otro índice de un arreglo de diccionarios.
- Valor en índice de segundo nivel (anidado).

Importante

El orden de la lista anterior si importa. Se usan los mismos casos que el caso previo, pero en otro orden. Acá primero se filtra y luego se extrae el valor desde el índice anidado.

Si tenemos el siguiente JSON:

```
{
  "mixed_with_chilts": [
    {
      "key": 20,
      "child": {
        "value": "hijo"
      }
    }
  ]
}
```

Para obtener el valor del índice **value** que está anidado dentro del índice child asociado al índice **key** que tiene como valor **20** usamos el siguiente selector:

```
mixed_with_chilts[key=20:child].value
```

Nota

Si en vez de seleccionar **value** usáramos este selector:

```
mixed_with_chilts[key=20:child]
```

Obtendríamos lo siguiente:

```
{
  "value": "hijo"
}
```

Como se podría suponer, esto significa que obtendremos un objeto, en este caso un diccionario. Los

selectores permiten obtener objetos. Sin embargo, en las configuraciones que se realizan en BillMySales esto nunca se debe hacer. En las configuraciones de BillMySales siempre debemos llegar al valor escalar (no objeto), ya sea: un texto (string), un número o un valor booleano.

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos de uso » Añadir texto al resultado

Añadir texto al resultado

Añadir texto al resultado

Anonymous · 08/10/2026

Añadir texto al resultado

Usando las cadenas literales y los selectores con paréntesis, tenemos 2 casos interesantes que nos permiten mejorar el formato de la salida de nuestro selector añadiendo texto y múltiples resultados concatenados.

Añadir texto al resultado del selector

Es posible añadir un texto fijo al resultado del selector. Esto se logra colocando el texto entre comillas dobles.

Por ejemplo, el siguiente selector siempre entregará el texto Hola Mundo:

```
"Hola Mundo"
```

Ya que al estar todo entre comillas dobles se interpreta como el texto tal cual está escrito y no como un selector que debe buscar datos en el JSON.

Para usar esto con un selector que extraiga datos, solo se debe escribir el selector entre paréntesis acompañado del texto entre comillas dobles. Esto permite hacer cosas como la siguiente:

```
"Valor anidado es: "(k_nested1.k_nested2)"
```

O la siguiente:

```
"$(total)" CLP"
```

Concatenar selectores

Para concatenar selectores solo se deben escribir entre paréntesis cada uno de ellos.

Ejemplos:

```
(k_simple)(k_nested1.k_nested2)
```

Sin embargo, es más interesante cuando lo unimos con lo anterior agregando un texto:

```
"Simple es "(k_simple)" y anidado es "(k_nested1.k_nested2)"
```

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos de uso » Elegir valor a seleccionar

Elegir valor a seleccionar

Elegir valor a seleccionar

Anonymous · 08/10/2026

Elegir valor a seleccionar

Gracias al operador OR de los selectores podemos elegir qué valor usar de una lista de selectores, tomando el primer elemento con valor (que no sea null). También podemos definir valores por defecto cuando un selector no entrega resultado (es null o vacío).

Seleccionar primer selector con un valor de una lista de selectores

Para poder separar diferentes selectores y elegir el primero que exista y tenga un valor se utiliza el OR con los símbolos “||” (doble barra vertical o doble pipe).

Si tenemos el siguiente JSON:

```
{
  "selector_k1": null,
  "selector_k3": "valor3",
  "selector_k4": "valor4",
}
```

Y usamos el siguiente selector:

```
selector_k1||selector_k2||selector_k3||selector_k4
```

Obtendremos el valor:

```
valor3
```

Porque es el primer valor encontrado que existe y no es null.

Valor por defecto de un selector

Este uso es una combinación de 2 usos vistos previamente:

- Añadir texto al resultado del selector.
- Seleccionar primer selector con un valor de una lista de selectores.

Si combinamos esos 2 usos podemos escribir el siguiente selector.

```
(selector) || "Valor por defecto"
```

Donde se buscará el valor asociado al **selector**, y si no existe, se pasará al siguiente elemento de la lista, que en este caso es un selector formado solo por el texto **Valor por defecto**. Con esto, si el selector no tiene un valor, siempre, por el valor por defecto, se tendrá uno.

Última actualización el 08/10/2026