

# JSONata Cheatsheet

This functionality is available in a Stedi module. Contact us for details.

All mappings expressions are based on JSONata language - this page showcases its most important features useful when creating a mapping.

## 1. JSON object source document

The source for each example is derived from the following JSON:

Unset

```
{
  "senderName": "STEDI",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Los Angeles",
    "state": "CA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2021/03/17",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2021/10/12",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2021/01/07",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,

```

```
      "volume": "35"
    }
  ]
}
```

## 1.1 Retrieving data

Root-level field

### Source

Unset

```
"senderName": "CHAINIO",
"customerID": "999333",
"shipmentID": 3312412,
"shipment type": "ASAP",
"address": {
  "street": "1234 Main St.",
  "city": "Philadelphia",
  "state": "PA",
  "country name": "USA",
  "is-europe": false
},
"orders": [
  {
    "orderDate": "2021/03/17",
    "productID": "DEG32",
    "quantity": 3,
    "pricePerUnit": 5,
    "volume": "10"
  },
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "volume": "15"
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
```

```
      "volume": "35"
    }
  ]
}
```

## JSONata

```
Unset
senderName
```

## Output

```
Unset
"CHAINIO"
```

Nested field in the root object

## Source

```
Unset
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2021/03/17",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
  ],
}
```

```
{
  "orderDate": "2021/10/12",
  "productID": "OIU98",
  "quantity": 100,
  "pricePerUnit": 1,
  "volume": "15"
},
{
  "orderDate": "2021/01/07",
  "productID": "PWE47",
  "quantity": 45,
  "pricePerUnit": 500,
  "volume": "35"
}
]
```

## JSONata

```
Unset
address.city
```

## Output

```
Unset
"Los Angeles"
```

Nested field with a dash in its name in the root object

## Source

```
Unset
"senderName": "CHAINIO",
"customerID": "999333",
"shipmentID": 3312412,
"shipment type": "ASAP",
"address": {
  "street": "1234 Main St.",
  "city": "Philadelphia",
```

```
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2021/03/17",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2021/10/12",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2021/01/07",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

## JSONata

```
Unset
address."is-europe"
```

## Output

```
Unset
false
```

Nested field in a root level array

## Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2021/03/17",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2021/10/12",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2021/01/07",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "senderName": "CHAINIO",
  "customerID": "999333",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  }
}
```

## JSONata

Unset

```
orders[1].productID
```

## Output

Unset

"OIU98"

Retrieve an array of items from a root level array

### Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2021/03/17",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2021/10/12",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2021/01/07",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "senderName": "CHAINIO",
  "customerID": "999333",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  }
}
```

**JSONata**

```
Unset
orders.orderDate
```

## Output

```
Unset
[
  "2021/03/17",
  "2021/10/12",
  "2021/01/07"
]
```

Retrieve a root-level field with whitespace in its key

## Source

```
Unset
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}
```

## JSONata

```
Unset
`shipment type`
```

## Output

```
Unset
"ASAP"
```

Retrieve a nested field with whitespace in its key

## Source

Unset

```
{
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333"
}
```

## JSONata

Unset

```
address.`country name`
```

## Output

Unset

```
"USA"
```

## 1.2 Operating on data

Concatenate two strings, separated by a space

### Source

Unset

```
{
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333"
}
```

```
}
```

## JSONata

```
Unset  
address.street & " " & address.city
```

## Output

```
Unset  
"1234 Main St. Philadelphia"
```

Multiply two values retrieved from an array together

It is not recommended to use JSONata for floating point arithmetic, such as financial calculations. Floating-point arithmetic can introduce rounding errors, which can accumulate and lead to incorrect results. We recommend representing monetary values as integers (e.g. cents) and performing calculations on those integers.

## Source

```
Unset  
{  
  "orders": [  
    {  
      "orderDate": "2024/03/15",  
      "productID": "DEG32",  
      "quantity": 3,  
      "pricePerUnit": 5,  
      "volume": "10"  
    },  
    {  
      "orderDate": "2023/09/09",  
      "productID": "OIU98",  
      "quantity": 100,  
      "pricePerUnit": 1,  
      "volume": "15"  
    },  
  ],  
}
```

```
{
  "orderDate": "2022/07/01",
  "productID": "PWE47",
  "quantity": 45,
  "pricePerUnit": 500,
  "volume": "35"
},
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "CA",
    "country name": "USA",
    "is-europe": false
  }
}
```

## JSONata

```
JavaScript
orders[-1].quantity * orders[-1].pricePerUnit
```

## Output

```
Unset
22500
```

(`[-1]` retrieves the last item in an array)

Remove a given character from a string

## Source

```
Unset
{
  "assigned_location_id": 3183479,
  "destination": {
```

```
"id": 54433189,
"zip": "K2P0V6",
"city": "Ottawa",
"email": "bob@mouseco.global",
"phone": "(555)555-5555",
"company": "",
"country": "Canada",
"address1": "123-Gruber-St",
"address2": "Unit 806",
"province": "Ontario",
"last_name": "Oleson",
"first_name": "Bob"
},
"line_items": [
  {
    "id": 123456789987,
    "shop_id": 3998762,
    "quantity": 3,
    "price": 32.78,
    "variant_id": 2385087,
    "line_item_id": 466157049,
    "inventory_item_id": 123456789987,
    "fulfillable_quantity": 1,
    "fulfillment_order_id": 1568020,
    "token": "",
    "position": 2
  },
  {
    "id": 123456789987,
    "shop_id": 3998762,
    "quantity": 2,
    "price": 12.98,
    "variant_id": 2385088,
    "line_item_id": 466157040,
    "inventory_item_id": 123456789987,
    "fulfillable_quantity": 1,
    "fulfillment_order_id": 1568021,
    "token": "",
    "position": 2
  },
  {
    "id": 123456789987,
    "shop_id": 3998762,
    "quantity": 3,
    "price": 9.92,
    "variant_id": 2385089,
    "line_item_id": 466157041,
```

```
      "inventory_item_id": 123456789987,
      "fulfillable_quantity": 1,
      "fulfillment_order_id": 1568022,
      "token": "",
      "position": 2
    }
  ],
  "order_id": 3183479,
  "request_status": "unsubmitted",
  "shop_id": 255858046,
  "status": "open",
  "assigned_location": {
    "zip": "K2P0V6",
    "city": "Ottawa",
    "name": "Bob Oleson",
    "phone": "(555)555-5555",
    "address1": "123 Gruber St",
    "address2": "Unit 806",
    "province": "Ontario",
    "location_id": 17232953366,
    "country_code": "CA"
  }
}
```

## JSONata

```
Unset
$replace(destination.address1, "-", "")
```

## Output

```
Unset
"123GruberSt"
```

Convert a value to a string

## Source

Unset

```
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

## JSONata

Unset

```
$string(shipmentID)
```

## Output

Unset

"3312412"

(\$string is a JSONata function)

Convert a value to a number

### Source

Unset

```
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

```
}
```

## JSONata

```
Unset  
$number(customerID)
```

## Output

```
Unset  
997321
```

(\$number is a JSONata function)

## 1.3 String manipulation

Convert a value to a lowercase string

### Source

```
Unset  
{  
  "senderName": "CHAINIO",  
  "customerID": "999333",  
  "shipmentID": 3312412,  
  "shipment type": "ASAP",  
  "address": {  
    "street": "1234 Main St.",  
    "city": "Philadelphia",  
    "state": "PA",  
    "country name": "USA",  
    "is-europe": false  
  },  
  "orders": [  
    {  
      "orderDate": "2023/03/15",  
      "productID": "DEG32",  
      "quantity": 3,  
    }  
  ]  
}
```

```
    "pricePerUnit": 5,  
    "volume": "10"  
  },  
  {  
    "orderDate": "2022/09/27",  
    "productID": "OIU98",  
    "quantity": 100,  
    "pricePerUnit": 1,  
    "volume": "15"  
  },  
  {  
    "orderDate": "2022/07/01",  
    "productID": "PWE47",  
    "quantity": 45,  
    "pricePerUnit": 500,  
    "volume": "35"  
  }  
]  
}
```

## JSONata

```
Unset  
$lowercase(senderName)
```

## Output

```
Unset  
"chainio"
```

Convert a value to an uppercase string

## Source

```
Unset  
{  
  "senderName": "CHAINIO",  
  "customerID": "999333",  
  "shipmentID": 3312412,  
}
```

```
{
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

## JSONata

Unset

`$uppercase(address.street)`

## Output

Unset

"1234 MAIN ST."

Truncate a string to only the first 4 characters

## Source

Unset

```
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

**JSONata**

Unset

```
$substring(address.street, 0, 4)
```

## Output

Unset

```
"1234"
```

Check if a string contains a given substring

## Source

Unset

```
{
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
```

```
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

## JSONata

```
Unset
$contains(address.street, "Main")
```

## Output

```
Unset
true
```

Get all characters in a string after a given substring

## Source

```
Unset
{
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",

```

```

      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}

```

## JSONata

```

Unset
$substringAfter(address.street, "1234 ")

```

## Output

```

Unset
"Main St."

```

Replace all /s with - in a string

## Source

```

Unset
{
  "orders": [
    {

```

```

        "orderDate": "2023/03/15",
        "productID": "DEG32",
        "quantity": 3,
        "pricePerUnit": 5,
        "volume": "10"
    },
    {
        "orderDate": "2022/09/27",
        "productID": "OIU98",
        "quantity": 100,
        "pricePerUnit": 1,
        "volume": "15"
    },
    {
        "orderDate": "2022/07/01",
        "productID": "PWE47",
        "quantity": 45,
        "pricePerUnit": 500,
        "volume": "35"
    }
],
"address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
},
"senderName": "CHAINIO",
"customerID": "999333",
"shipmentID": 3312412,
"shipment type": "ASAP"
}

```

## JSONata

```

Unset
$join($split(orders[0].orderDate, "/"), '-')

```

## Output

Unset

"2023-03-15"

Replace all /s with - in in a string (using ~> operator)

### Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}
```

### JSONata

Unset

```
$split(orders[0].orderDate, "/") ~> $join('-')
```

## Output

Unset

```
"2023-03-15"
```

## 1.4 Conditionals

Use a conditional to return a value based on a condition

### Source

Unset

```
{
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    }
  ]
}
```

```
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

## JSONata

```
Unset
address."is-europe" ? "EUR" : "USD"
```

## Output

```
Unset
"USD"
```

Use a conditional with a function to return a value based on a condition

## Source

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,

```

```
    "pricePerUnit": 1,  
    "volume": "15"  
  },  
  {  
    "orderDate": "2022/07/01",  
    "productID": "PWE47",  
    "quantity": 45,  
    "pricePerUnit": 500,  
    "volume": "35"  
  }  
],  
"address": {  
  "street": "1234 Main St.",  
  "city": "Philadelphia",  
  "state": "PA",  
  "country name": "USA",  
  "is-europe": false  
},  
"senderName": "CHAINIO",  
"customerID": "999333",  
"shipmentID": 3312412,  
"shipment type": "ASAP"  
}
```

## JSONata

Unset

```
$count(orders) > 3 ? "LARGE" : "SMALL"
```

## Output

Unset

```
"SMALL"
```

## 1.5 Filtering data

Filter data if a given value is greater than N

### Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}
```

## JSONata

Unset

```
orders[quantity > 10]
```

## Output

Unset

```
[
  {
    "orderDate": "2022/09/27",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "volume": "15"
  },
  {
    "orderDate": "2022/07/01",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "volume": "35"
  }
]
```

Filter data if a given value is equal to X

### Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}
```

```
],
"address": {
  "street": "1234 Main St.",
  "city": "Philadelphia",
  "state": "PA",
  "country name": "USA",
  "is-europe": false
},
"senderName": "CHAINIO",
"customerID": "999333",
"shipmentID": 3312412,
"shipment type": "ASAP"
}
```

## JSONata

```
Unset
orders[productID = "PWE47"].orderDate
```

## Output

```
Unset
"2022/07/01"
```

Filter data based on a complex condition

## Source

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
```

```

      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}

```

## JSONata

```

Unset
orders[quantity * pricePerUnit > 200].productID[]

```

## Output

```

Unset
[
  "PWE47"
]

```

The `[]` is required at the end of an expression to convert the result to an array.

## 1.6 Counting data

Count fields based on a greater-than filter expression

### Source

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}
```

### JSONata

Unset

```
$count(orders[quantity > 50])
```

## Output

Unset

1

Count fields based on a substring value filter expression

## Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  }
}
```

```
    },
    "senderName": "CHAINIO",
    "customerID": "999333",
    "shipmentID": 3312412,
    "shipment type": "ASAP"
  }
}
```

## JSONata

Unset

```
$count(orders[$substring(orderDate, 0, 4) = "2021"])
```

## Output

Unset

3

Count fields based on a substring value with `~>` operator

## Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {

```

```

        "orderDate": "2022/07/01",
        "productID": "PWE47",
        "quantity": 45,
        "pricePerUnit": 500,
        "volume": "35"
      }
    ],
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    },
    "senderName": "CHAINIO",
    "customerID": "999333",
    "shipmentID": 3312412,
    "shipment type": "ASAP"
  }
}

```

## JSONata

```

Unset
orders[$substring(orderDate, 0, 4) = "2021"] ~> $count

```

## Output

```

Unset
3

```

Add up all values in an array

## Source

```

Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",

```

```

      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}

```

## JSONata

Unset

```
$sum(orders.quantity)
```

## Output

Unset

148

Add up all values in an array with `~>` operator

It is not recommended to use JSONata for floating point arithmetic, such as financial calculations. Floating-point arithmetic can introduce rounding errors, which can accumulate and lead to incorrect results. We recommend representing monetary values as integers (e.g. cents) and performing calculations on those integers.

Use `$map`, `$sum` and `~>` operator to add up all values in an array

### Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
}
```

```
"senderName": "CHAINIO",
"customerID": "999333",
"shipmentID": 3312412,
"shipment type": "ASAP"
}
```

## JSONata

```
Unset
orders[quantity > 10].quantity ~> $sum
```

## Output

```
Unset
145
```

`$map` is used to convert all items in `orders.volume` array to a `$number`. **Source**

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
```

```

      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}

```

## JSONata

```

Unset
$map(orders.volume, $number) ~> $sum

```

## Output

```

Unset
60

```

## 1.7 Variables

In JSONata, any name that starts with a **\$** is a variable (e.g. **\$streetName := address.street**). A variable can be one of any type in JSONata's type system.

### Built-in variables

- **\$** – the variable with no name refers to the context value at any point in the input JSON hierarchy.
- **\$\$** – the root of the input JSON. You can use it to break out of the current context and navigate down a different path.

Convert a list of fields to the desired format using the built-in `$` variable

### Source

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}
```

### JSONata

```
Unset
orders.orderDate.{
  "originalDate": $,
  "date": $convertDateTime($, "yyyy/MM/dd", "yyyy-MM-dd")
}
```

## Output

```
Unset
[
  {
    "originalDate": "2023/03/15",
    "date": "2023-03-15"
  },
  {
    "originalDate": "2022/09/27",
    "date": "2022-09-27"
  }
]
```

Populate an object field using the built-in **\$\$** variable for every item while looping over an array

## Source

```
Unset
{
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP",
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
```

```

      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ]
}

```

## JSONata

```

Unset
orders.{
  "productId": productID,
  "address": $$address
}

```

## Output

```

Unset
[
  {
    "productId": "DEG32",
    "address": {
      "is-europe": false,
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA"
    }
  }
]

```

```

    },
    {
      "productId": "OIU98",
      "address": {
        "is-europe": false,
        "street": "1234 Main St.",
        "city": "Philadelphia",
        "state": "PA",
        "country name": "USA"
      }
    },
    {
      "productId": "PWE47",
      "address": {
        "is-europe": false,
        "street": "1234 Main St.",
        "city": "Philadelphia",
        "state": "PA",
        "country name": "USA"
      }
    }
  ]

```

Convert value to a string and store it in a variable

### Source

```

Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {

```

```

        "orderDate": "2022/07/01",
        "productID": "PWE47",
        "quantity": 45,
        "pricePerUnit": 500,
        "volume": "35"
      },
    ],
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    },
    "senderName": "CHAINIO",
    "customerID": "999333",
    "shipmentID": 3312412,
    "shipment type": "ASAP"
  }
}

```

## JSONata

```

Unset
$shipmentIDAsString := $string(shipmentID)

```

## Output

```

Unset
"3312412"

```

`$variableName := value` is how assigns `value` to a JSONata `$variableName` variable

Count all items in an array and return a different result based on its size

## Source

```

Unset
{
  "orders": [

```

```

    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}

```

## JSONata

```

Unset
(
  $ordersCount := $count(orders);
  $ordersCount > 2 ? "Large order" : "Small order"
)

```

## Output

Unset

"Large order"

Multiply values across all items in an array and return a boolean flag based on the result

### Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  },
  "senderName": "CHAINIO",
  "customerID": "999333",
  "shipmentID": 3312412,
  "shipment type": "ASAP"
}
```

## JSONata

```
Unset
(
  $orderCosts := $map(orders, function($order) {
    $order.quantity * $order.pricePerUnit
  });
  $orderCosts ~> $sum > 10000
)
```

## Output

```
Unset
true
```

## Positional variables

Positional variables binding can be used to determine at which position in the sequence the current context item is. It can be used following any map, filter or order-by stage in the path.

The variable is available for use within subsequent stages of the path (e.g. within filter predicates or a map operation) and goes out of scope at the end of the path expression.

Populate order title based on its index within the orders array

## Source

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,

```

```

        "volume": "15"
      },
      {
        "orderDate": "2022/07/01",
        "productID": "PWE47",
        "quantity": 45,
        "pricePerUnit": 500,
        "volume": "35"
      }
    ],
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    },
    "senderName": "CHAINIO",
    "customerID": "999333",
    "shipmentID": 3312412,
    "shipment type": "ASAP"
  }
}

```

## JSONata

```

Unset
orders#$myIndex.{
  "productId": productId,
  "orderTitle": "Order #" & ($myIndex + 1)
}

```

## Output

```

Unset
[
  {
    "productId": "DEG32",
    "orderTitle": "Order #1"
  },
  {
    "productId": "OIU98",

```

```
      "orderTitle": "Order #2"  
    }  
  ]
```

## 1.8 Dates

Get the current date & time in ISO 8601 format

### JSONata

```
Unset  
$now()
```

### Output

```
Unset  
"2025-02-10T19:39:01.069Z"
```

Get the current date & time in 6-character EDI date format

### JSONata

```
Unset  
$convertDateTime($now(), $dateTime.RFC3339Millis, $dateTime.EDIDate)
```

### Output

```
Unset  
"250210"
```

Get the current date & time in 8-character EDI date format

### JSONata

Unset

```
$convertDateTime($now(), $dateTime.RFC3339Millis, $dateTime.EDIDateLong)
```

## Output

Unset

```
"20250210"
```

Convert a date from `yyyy/MM/dd` format to `DD/MM/YYYY`

## Source

Unset

```
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {
      "orderDate": "2022/07/01",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "volume": "35"
    }
  ],
  "address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
  }
}
```

```
    },
    "senderName": "CHAINIO",
    "customerID": "999333",
    "shipmentID": 3312412,
    "shipment type": "ASAP"
  }
}
```

## JSONata

```
Unset
$convertDateTime(orders[0].orderDate, "yyyy/MM/dd", "dd/MM/yyyy")
```

## Output

```
Unset
"15/03/2023"
```

Get the month given a date in `yyyy/MM/dd` format

## Source

```
Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",
      "productID": "DEG32",
      "quantity": 3,
      "pricePerUnit": 5,
      "volume": "10"
    },
    {
      "orderDate": "2022/09/27",
      "productID": "OIU98",
      "quantity": 100,
      "pricePerUnit": 1,
      "volume": "15"
    },
    {

```

```

        "orderDate": "2022/07/01",
        "productID": "PWE47",
        "quantity": 45,
        "pricePerUnit": 500,
        "volume": "35"
      }
    ],
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    },
    "senderName": "CHAINIO",
    "customerID": "999333",
    "shipmentID": 3312412,
    "shipment type": "ASAP"
  }
}

```

## JSONata

```

Unset
$convertDateTime(orders[1].orderDate, "yyyy/MM/dd", "MM")

```

## Output

```

Unset
"09"

```

Convert a date taken from an array from `yyyy/MM/dd` format to `DD-MM-YYYY`

## Source

```

Unset
{
  "orders": [
    {
      "orderDate": "2023/03/15",

```

```

        "productID": "DEG32",
        "quantity": 3,
        "pricePerUnit": 5,
        "volume": "10"
    },
    {
        "orderDate": "2022/09/27",
        "productID": "OIU98",
        "quantity": 100,
        "pricePerUnit": 1,
        "volume": "15"
    },
    {
        "orderDate": "2022/07/01",
        "productID": "PWE47",
        "quantity": 45,
        "pricePerUnit": 500,
        "volume": "35"
    }
],
"address": {
    "street": "1234 Main St.",
    "city": "Philadelphia",
    "state": "PA",
    "country name": "USA",
    "is-europe": false
},
"senderName": "CHAINIO",
"customerID": "999333",
"shipmentID": 3312412,
"shipment type": "ASAP"
}

```

## JSONata

Unset

```
$convertDateTime(orders[-1].orderDate, "yyyy/MM/dd", "dd-MM-yyyy")
```

## Output

Unset  
`"01-07-2022"`

Convert epoch date to EDI date format

### Source

Unset  
`{ "epoch": 1648812955 }`

### JSONata

Unset  
`$convertDateTime($fromMillis(epoch * 1000), $dateTime.RFC3339Millis, $dateTime.EDIDateLong)`

### Output

Unset  
`"20220401"`

Convert given UTC date to `America/New_York` timezone

### Source

Unset  
`{ "date": "2022-04-01T11:48:58.451Z" }`

### JSONata

Unset  
`$convertDateTime(date, $dateTime.RFC3339Millis, $dateTime.RFC3339Millis, "UTC", "America/New_York" )`

### Output

Unset

"2022-04-01T07:48:58.451-04:00"

## 2. JSON array-of-objects source document

The source for each example is derived from the following JSON:

Unset

```
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

### 2.1 Retrieving data

Retrieve a field from the first item of a root-level array

**Source**

Unset

```
[
  {
    "orderDate": "2021/10/12",
    "productID": "0IU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

Unset

```
$$[0].orderDate
```

## Output

Unset

```
"2021/10/12"
```

Retrieve an array of fields from a root-level array

## Source

Unset

```
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

Unset

orderDate

## Output

Unset

```
[
  "2021/10/12",
  "2021/01/07"
]
```

```
]
```

Retrieve a nested value from a root-level array

### Source

```
Unset
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

### JSONata

```
Unset
$$[1].address."is-europe"
```

### Output

Unset

true

Retrieve a value with a space in its key from a root-level array

### Source

Unset

```
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

### JSONata

Unset

`$$[0].address.`country name``

## Output

```
Unset
"USA"
```

## 2.2 Operating on data

The data operations expressions operate on the same basis as for the JSON object source documents.

## 2.3 String manipulation

The string manipulation expressions operate on the same basis as for the JSON object source documents.

## 2.4 Conditionals

The conditionals expressions operate on the same basis as for the JSON object source documents.

## 2.5 Filtering data

The data filtering expressions operate on the same basis as for the JSON object source documents.

Instead of selecting an array field, you can select the root array with the `$$` selector.

Get an array of all items based on a filter expression

## Source

```
Unset
[
  {
    "orderDate": "2021/10/12",
    "productID": "0IU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  }
]
```

```
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

```
Unset
$$[quantity > 10]
```

## Output

```
Unset
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
```

```

    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]

```

## 2.6 Counting data

The data counting expressions operate on the same basis as for the JSON object source documents.

Instead of selecting an array field, you can select the root array with the `$$` selector.

Count all items in an array based on a greater-than filter expression

### Source

```

Unset
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,

```

```
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

```
Unset
$count($$[quantity > 50])
```

## Output

```
Unset
1
```

Count all items in an array based on a substring filter expression

## Source

```
Unset
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
]
```

```

{
  "orderDate": "2021/01/07",
  "productID": "PWE47",
  "quantity": 45,
  "pricePerUnit": 500,
  "address": {
    "street": "La Malaga",
    "city": "Porto",
    "state": "Porto",
    "country name": "Portugal",
    "is-europe": true
  }
}
]

```

## JSONata

```

Unset
$count($[$substring(orderDate, 0, 4) = "2021"])

```

## Output

```

Unset
2

```

`$substring(orderDate, 0, 4)` returns first four characters of a string)

Count all items in an array based on a substring filter expressions using `~>` operator

## Source

```

Unset
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {

```

```
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    },
    {
      "orderDate": "2021/01/07",
      "productID": "PWE47",
      "quantity": 45,
      "pricePerUnit": 500,
      "address": {
        "street": "La Malaga",
        "city": "Porto",
        "state": "Porto",
        "country name": "Portugal",
        "is-europe": true
      }
    }
  ]
}
```

## JSONata

```
Unset
$$[$substring(orderDate, 0, 4) = "2021"] ~> $count
```

## Output

```
Unset
2
```

Sum up all items in an array

## Source

```
Unset
[
  {
```

```
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

```
Unset
$sum(quantity)
```

## Output

```
Unset
145
```

Sum up all items in an array based on a filter expression

## Source

Unset

```
[
  {
    "orderDate": "2021/10/12",
    "productID": "0IU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

Unset

```
$$[quantity > 10].quantity ~> $sum
```

## Output

Unset

145

Use `$map`, `$sum` and `~>` operator to add up all values in an array

## Source

Unset

```
[
  {
    "orderDate": "2021/10/12",
    "productID": "OIU98",
    "quantity": 100,
    "pricePerUnit": 1,
    "address": {
      "street": "1234 Main St.",
      "city": "Philadelphia",
      "state": "PA",
      "country name": "USA",
      "is-europe": false
    }
  },
  {
    "orderDate": "2021/01/07",
    "productID": "PWE47",
    "quantity": 45,
    "pricePerUnit": 500,
    "address": {
      "street": "La Malaga",
      "city": "Porto",
      "state": "Porto",
      "country name": "Portugal",
      "is-europe": true
    }
  }
]
```

## JSONata

Unset

```
$map(volume, $number) ~> $sum
```

## Output

Unset

```
25
```