

# Response Fields

The responses returned by a GET operation can be large: they may include payloads such as Order objects or Product objects that contain over a hundred properties each. Depending on the specific data you want to retrieve, you may not always need the API to return every property for a given resource. Similarly, you may not always want to update every property on a resource when you execute a PUT operation. Furthermore, retrieving or updating only the data you need allows your operations to maintain optimal performance.

The `responseFields` argument allows you to filter which properties are returned or updated by the API, providing you with a more manageable payload to work with and reducing network latency and client SDK parse times.

Use the `responseFields` argument carefully when retrieving and then modifying a resource, as you can inadvertently delete data. For example, if you use the `responseFields` argument to retrieve a subset of a resource, make changes to that subset, and then update the resource with the new subset without using the same `responseFields` argument again, the entire resource may be overwritten with only the subset of data. In general, the `responseFields` argument is recommended for retrieving select data to display or modify on the storefront, and for selectively updating data. It is not generally recommended for retrieving API resources you intend to modify or process before updating the original object with the changes.

## HTTP Request Syntax

To use the `responseFields` argument, include it in the GET or PUT HTTP request. For example, the following line shows the placement of the argument for the `GetProduct` operation request:

```
GET api/commerce/catalog/admin/products/{productCode}?responseFields={responseFields}
```

The `responseFields` argument is optional, and if no value is provided, then the entire resource is retrieved or updated.

Not every PUT operation supports the `responseFields` argument. Consult the [API reference documentation](#) for details on a specific operation.

## Basic Example: Include Single Property

The simplest use of the `responseFields` argument is to retrieve one specific property.

### Example

#### Argument To Use

```
GET api/endpointPath/?responseFields=a1
```

#### API Object

```
{
  "a1": "value",
  "a2": "value",
  "a3": "value"
}
```

### **Payload Response**

```
{
  "a1": "value",
}
```

## Include Multiple Properties

Similar to the single property example, you can retrieve or update multiple properties at a time by separating them with a comma.

### Example

#### **Argument To Use**

```
GET api/endpointPath/?responseFields=a1,a3
```

#### **API Object**

```
{
  "a1": "value",
  "a2": "value",
  "a3": "value"
}
```

### **Payload Response**

```
{
  "a1": "value",
  "a3": "value"
}
```

## Exclude Properties

You can exclude one or more properties to retrieve or update by using the negative sign operator.

### Example

#### **Argument To Use**

```
GET api/endpointPath/?responseFields=-a2
```

#### **API Object**

```
{
  "a1": "value",
  "a2": "value",
  "a3": "value"
}
```

### **Payload Response**

```
{
  "a1": "value",
  "a3": "value"
}
```

## Include and Exclude Properties

You can specify properties to include and exclude within the same argument by combining commas and negative sign operators.

### Example

#### **Argument To Use**

GET `api/endpointPath/?responseFields=a1,-a3,a4`

#### **API Object**

```
{
  "a1": "value",
  "a2": "value",
  "a3": "value",
  "a4": "value"
}
```

### **Payload Response**

```
{
  "a1": "value",
  "a4": "value"
}
```

## Include Wildcards

You can use an asterisk to include properties that are otherwise not explicitly included or excluded in the argument.

### Example

#### **Argument To Use**

The following three arguments all return the same payload.

```
GET api/endpointPath/?responseFields=a1,-a2,*
```

```
GET api/endpointPath/?responseFields=-a2,a3,*
```

```
GET api/endpointPath/?responseFields=*,a2
```

### API Object

```
{  
  "a1": "value",  
  "a2": "value",  
  "a3": "value"  
}
```

### Payload Response

```
{  
  "a1": "value",  
  "a3": "value"  
}
```

## Arrays

The `responseField` argument works the same for arrays as it does for simple structures. You can include and exclude arrays, as well as use the asterisk symbol to denote wildcards.

### Example

#### Argument To Use

```
GET api/endpointPath/?responseFields=a1,-a3,*
```

### API Object

```
{  
  "a1": [ "value1", "value2" ],  
  "a2": [ "value1", "value2" ],  
  "a3": [ "value1", "value2" ]  
}
```

### Payload Response

```
{  
  "a1": [ "value1", "value2" ],  
  "a2": [ "value1", "value2" ]  
}
```

## Include Specific Subresources

The majority of API objects are composed of nested subresources. You can use a forward slash character in the `responseFields` argument to include only certain subresources.

## Example

### Argument To Use

GET `api/endpointPath/?responseFields=a1/b2`

### API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

### Payload Response

```
{
  "a1": {
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Include Subresources at any Depth

You can limit inclusion of subresources down to any depth necessary by stringing forward slashes together.

## Example

### Argument To Use

GET `api/endpointPath/?responseFields=a1/b2/c1`

### API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Payload Response

```
{
  "a1": {
    "b2": {
      "c1": "value1"
    }
  }
}
```

## Exclude Subresources

You can exclude subresources at any depth using the negative sign operator.

### Example

#### Argument To Use

GET `api/endpointPath/?responseFields=a1/b2/-c1`

#### API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Payload Response

```
{
  "a1": {
    "b2": {
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Include a Subset of Subresources

You can use parentheses to include a subset of subresources. Only the fields between the parentheses are retrieved or updated. Using parentheses instead of forward slashes may be more convenient depending on your use case.

### Example

## Argument To Use

GET `api/endpointPath/?responseFields=a1(b1,b2(c2))`

## API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Payload Response

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c2": "value2"
    }
  }
}
```

## Syntax Combinations

The `responseFields` argument lets you combine the syntax operators in the aforementioned sections to achieve powerful and flexible inclusions and exclusions. The following examples in this section demonstrate several example scenarios to give you an idea of how flexible the `responseFields` argument is:

### Combination: Asterisks and Parentheses

#### Argument To Use

GET `api/endpointPath/?responseFields=a1(*,b2(c1))`

#### API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Payload Response

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1"
    }
  }
}
```

Combination: Forward Slashes, and Parentheses

## Argument To Use

GET `api/endpointPath/?responseFields=a1/b2(c1,c2)`

## API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

## Payload Response

```
{
  "a1": {
    "b2": {
      "c1": "value1",
      "c2": "value2"
    }
  }
}
```

Combination: Asterisks, Forward Slashes, and Parentheses

## Argument To Use

GET `api/endpointPath/?responseFields=a1(*,b2/c1)`

## API Object

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

### **Payload Response**

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1"
    }
  }
}
```

Combination: Negation, Asterisks, Forward Slashes, and Parentheses

### **Argument To Use**

GET `api/endpointPath/?responseFields=a1(*,b2(-c2,-c3))`

### **API Object**

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1",
      "c2": "value2",
      "c3": "value3"
    }
  }
}
```

### **Payload Response**

```
{
  "a1": {
    "b1": "value",
    "b2": {
      "c1": "value1"
    }
  }
}
```

Combination: Arrays

### **Argument To Use**

GET `api/endpointPath/?responseFields=a1(b1/c1,*)`

### API Object

```
{
  "a1" : [
    {
      "b1" : {
        "c1" : "value",
        "c2" : "value",
        "c3" : "value"
      },
      "b2" : "value"
    },
    {
      "b1" : {
        "c1" : "value1",
        "c2" : "value2",
        "c3" : "value3"
      },
      "b2" : "value"
    }
  ]
}
```

### Payload Response

```
{
  "a1" : [
    {
      "b1" : {
        "c1" : "value"
      },
      "b2" : "value"
    },
    {
      "b1" : {
        "c1" : "value1"
      },
      "b2" : "value"
    }
  ]
}
```