
Place Order REST API Developer Guide

Version 60.0, Spring '24



CONTENTS

Place Order REST API Developer Guide	1
Requirements and Limitations	1
Authorization Through Connected Apps and OAuth 2.0	2
Understanding Place Order REST API Resources	3
Add a Contract and Orders to an Existing Account	3
Add an Order to an Existing Account	5
Add Orders to an Existing Contract	6
Add Order Products to an Existing Order	9
Get Details About a Contract	10
Get Details About an Order	12
Filter Details About a Contract	13
Filter Details About an Order	14
Place Order REST API Resource Reference	15
Create Contract-based Orders	15
Contract-based Orders	18
Create Order	21
Order	23

PLACE ORDER REST API DEVELOPER GUIDE

Access your organization's order and contract data programmatically with the Salesforce Place Order REST API.

The Salesforce Place Order REST API is a composite API that gives programmatic access to contract, order, and order product data, as well as child custom object data of contracts and orders in Salesforce. With this composite API, you can create contract, order, order product, and custom object records in a single call. Any organization that has orders and API enabled can use the Place Order REST API.

Use the Place Order REST API, a REST-based composite application programming interface, to:

- Add orders to a new or existing contract, and add order products to those orders.
- Add order products to a new or existing order.
- Add custom objects to a new or existing contract or order.
- Retrieve order records under a given contract, plus those orders' custom objects and order products.
- Retrieve order product records under a given order, plus custom object records under the order and its order products.
- Retrieve a filtered list of orders under a given contract or order products under a given order.

IN THIS SECTION:

[Requirements and Limitations](#)

[Authorization Through Connected Apps and OAuth 2.0](#)

For a client application to access REST API resources, it must be authorized as a safe visitor. To implement this authorization, use a connected app and an OAuth 2.0 authorization flow.

[Understanding Place Order REST API Resources](#)

Integrate your order creation system with Salesforce by using the Place Order REST API.

[Place Order REST API Resource Reference](#)

Each Place Order REST API resource is a URI used with an HTTP method (such as GET).

Requirements and Limitations

To access the Place Order REST API, you must establish a secure OAuth session ID.

Consider these limitations and general limits when using the Place Order REST API.

Limits and Limitations

- 2000 records per request or the API maximum limit for your organization—whichever is lower.
- Responses and requests are in JSON.

When using Place Order REST API resources that require a request or response body, use Content-Type: application/json.

- Each call can only affect one top-level entity.

For orders under a contract, you need one call for each new or existing contract you're adding orders, order products, or custom objects to. For orders not under a contract, you need one call for each new or existing order you're adding order products or custom objects to.

- In each resource, you can create custom objects at a depth of one level below the top-level entity.

- `/services/data/latest API version/commerce/sale` supports custom object records on contracts and orders.
- `/services/data/latest API version/commerce/sale/order` supports custom object records on orders and order products.
- Custom objects are not supported as children of other custom objects.
- To filter GET results, query parameters must be a fully qualified field name. The parent entity must be lower-cased (such as `order`), and the field must match the defined relationship name (such as `orders.StatusCode`).
For example, to get a list of all orders with a draft status under a given contract, you must use `contract.orders.StatusCode='Draft'`.
- When you create a new order, the `Status Code` must be Draft and the `Status` must be any value that corresponds to a `Status Code` of Draft.
- You can't update existing records.

Authorization Through Connected Apps and OAuth 2.0

For a client application to access REST API resources, it must be authorized as a safe visitor. To implement this authorization, use a connected app and an OAuth 2.0 authorization flow.

Configure a Connected App

A connected app requests access to REST API resources on behalf of the client application. For a connected app to request access, it must be integrated with your org's REST API using the OAuth 2.0 protocol. OAuth 2.0 is an open protocol that authorizes secure data sharing between applications through the exchange of tokens.

For instructions to configure a connected app, see [Create a Connected App](#) in *Salesforce Help*. Specifically, follow the steps in [Enable OAuth Settings for API Integration](#).

Apply an OAuth Authorization Flow

OAuth authorization flows grant a client app restricted access to REST API resources on a resource server. Each OAuth flow offers a different process for approving access to a client app, but in general the flows consist of three main steps.

1. To initiate an authorization flow, a connected app on behalf of a client app requests access to a REST API resource.
2. In response, an authorizing server grants access tokens to the connected app.
3. A resource server validates these access tokens and approves access to the protected REST API resource.

After reviewing and selecting an OAuth authorization flow, apply it to your connected app. For details about each supported flow, see [OAuth Authorization Flows](#) in *Salesforce Help*.

More Resources

Salesforce offers the following resources to help you navigate connected apps and OAuth:

- *Salesforce Help*: [Connected Apps](#)
- *Salesforce Help*: [Authorize Apps with OAuth](#)
- *Salesforce Help*: [OpenID Connect Token Introspection](#)

- [Trailhead: Build Integrations Using Connected Apps](#)

Understanding Place Order REST API Resources

Integrate your order creation system with Salesforce by using the Place Order REST API.

Use this API to:

IN THIS SECTION:

- [Add a Contract and Orders to an Existing Account](#)
- [Add an Order to an Existing Account](#)
- [Add Orders to an Existing Contract](#)
- [Add Order Products to an Existing Order](#)
- [Get Details About a Contract](#)
- [Get Details About an Order](#)
- [Filter Details About a Contract](#)
- [Filter Details About an Order](#)

Add a Contract and Orders to an Existing Account

Here's an example of a POST request using the [Create Contract-based Orders](#) resource to create a contract—with child orders, order products, and custom objects—to an existing account.

Example usage

```
/services/data/v30.0/commerce/sale
```

Example request body

```
{
  "contract": [
    {
      "attributes": {
        "type": "Contract"
      },
      "AccountId": "001D000000JRDAv",
      "StartDate": "2013-08-01",
      "Status": "Draft",
      "ContractTerm": "1",
      "Test_Contract1__r": {
        "records": [
          {
            "attributes": {
              "type": "Test_Contract1__c"
            },
            "Name": "Contract Custom Object"
          }
        ]
      }
    }
  ],
}
```

```

    "Orders": {
      "records": [
        {
          "attributes": {
            "type": "Order"
          },
          "EffectiveDate": "2013-08-11",
          "Status": "Draft",
          "billingCity": "SFO-Inside-OrderEntity-1",
          "Pricebook2Id": "01sD0000000G2NjIAK",
          "OrderItems": {
            "records": [
              {
                "attributes": {
                  "type": "OrderItem"
                },
                "PricebookEntryId": "01uD00000001c5toIAA",
                "quantity": "1",
                "UnitPrice": "10"
              }
            ]
          }
        }
      ]
    }
  ]
}

```

Example JSON response body

```

{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Contract",
      "url" : "/services/data/v30.0/subjects/Contract/800D0000000PIcMIAW"
    },
    "Id" : "800D0000000PIcMIAW",
    "Orders" : {
      "totalSize" : 1,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "Order",
          "url" : "/services/data/v30.0/subjects/Order/801D0000000G0ylIAC"
        },
        "Id" : "801D0000000G0ylIAC",
        "OrderItems" : {
          "totalSize" : 1,
          "done" : true,
          "records" : [ {
            "attributes" : {
              "type" : "OrderItem",

```



```

        "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CKoyIAG"
      },
      "Id" : "802D0000000CKoyIAG"
    } ]
  }
} ]
},
"Test_Contract1__r" : {
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Test_Contract1__c",
      "url" : "/services/data/v30.0/subjects/Test_Contract1__c/a02D00000006YKZIA4"
    },
    "Id" : "a02D00000006YKZIA4"
  } ]
}
} ]
}
}
}

```

Add an Order to an Existing Account

Here's an example of a POST request using the [Create Order](#) resource to create an order with order products for an existing account.

Example usage

```
/services/data/v30.0/commerce/sale/order
```

Example request body

```

{
  "order": [
    {
      "attributes": {
        "type": "Order"
      },
      "EffectiveDate": "2013-07-11",
      "Status": "Draft",
      "billingCity": "SFO-Inside-OrderEntity-1",
      "accountId": "001D0000000JRDAv",
      "Pricebook2Id": "01sD0000000G2NjIAK",
      "OrderItems": {
        "records": [
          {
            "attributes": {
              "type": "OrderItem"
            },
            "PricebookEntryId": "01uD00000001c5toIAA",
            "quantity": "1",
            "UnitPrice": "15.99"
          }
        ]
      }
    }
  ]
}

```

```

    ]
  }

```

Example JSON response body

```

{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Order",
      "url" : "/services/data/v30.0/subjects/Order/801D0000000G0ySIAS"
    },
    "Id" : "801D0000000G0ySIAS",
    "OrderItems" : {
      "totalSize" : 1,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "OrderItem",
          "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CKp8IAG"
        },
        "Id" : "802D0000000CKp8IAG"
      } ]
    } ]
  } ]
}

```

Add Orders to an Existing Contract

Here's an example of a PATCH request using the [Contract-based Orders](#) resource to add new orders and order products to an existing contract.

Example usage

```

/services/data/v30.0/commerce/sale/800D0000000PFL8IAO

```

Example request body

```

{
  "contract": [
    {
      "attributes": {
        "type": "Contract"
      },
      "Id" : "800D0000000PFL8IAO",
      "Orders": {
        "records": [
          {
            "attributes": {
              "type": "Order"
            },
            "EffectiveDate": "2013-08-11",
            "Status": "Draft",
            "billingCity": "SFO-Inside-OrderEntity-1",

```

```

"contractId": "800D0000000PFL8IAO",
"pricebook2Id": "01sD0000000G2JbIAK",
"OrderItems": {
  "records": [
    {
      "attributes": {
        "type": "OrderItem"
      },
      "PricebookEntryId": "01uD0000001c5tjIAA",
      "quantity": "12",
      "UnitPrice": "10"
    },
    {
      "attributes": {
        "type": "OrderItem"
      },
      "PricebookEntryId": "01uD0000001cAkMIAU",
      "quantity": "2",
      "UnitPrice": "20"
    }
  ]
},
{
  "attributes": {
    "type": "Order"
  },
  "EffectiveDate": "2013-10-11",
  "Status": "Draft",
  "billingCity": "SFO-Inside-OrderEntity-1",
  "contractId": "800D0000000PFL8IAO",
  "pricebook2Id": "01sD0000000G2JbIAK",
  "OrderItems": {
    "records": [
      {
        "attributes": {
          "type": "OrderItem"
        },
        "PricebookEntryId": "01uD0000001cAkRIAU",
        "quantity": "11",
        "UnitPrice": "10"
      },
      {
        "attributes": {
          "type": "OrderItem"
        },
        "PricebookEntryId": "01uD0000001cAkWIAU",
        "quantity": "2",
        "UnitPrice": "20"
      },
      {
        "attributes": {
          "type": "OrderItem"
        }
      }
    ]
  }
}

```

```

        "PricebookEntryId": "01uD0000001cAkgIAE",
        "quantity": "14",
        "UnitPrice": "10"
      }
    ]
  }
}
]
}

```

Example JSON response body

```

{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Contract"
    },
    "Orders" : {
      "totalSize" : 2,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "Order",
          "url" : "/services/data/v29.0/subjects/Order/801D0000000G0xsIAC"
        },
        "Id" : "801D0000000G0xsIAC",
        "OrderItems" : {
          "totalSize" : 2,
          "done" : true,
          "records" : [ {
            "attributes" : {
              "type" : "OrderItem",
              "url" : "/services/data/v29.0/subjects/OrderItem/802D0000000CKoPIAW"
            },
            "Id" : "802D0000000CKoPIAW"
          }, {
            "attributes" : {
              "type" : "OrderItem",
              "url" : "/services/data/v29.0/subjects/OrderItem/802D0000000CKoQIAW"
            },
            "Id" : "802D0000000CKoQIAW"
          } ]
        }
      } ],
      "Id" : "801D0000000G0xtIAC",
      "OrderItems" : {

```

```
    "totalSize" : 3,
    "done" : true,
    "records" : [ {
      "attributes" : {
        "type" : "OrderItem",
        "url" : "/services/data/v29.0/subjects/OrderItem/802D0000000CKoRIAW"
      },
      "Id" : "802D0000000CKoRIAW"
    }, {
      "attributes" : {
        "type" : "OrderItem",
        "url" : "/services/data/v29.0/subjects/OrderItem/802D0000000CKoSIAW"
      },
      "Id" : "802D0000000CKoSIAW"
    }, {
      "attributes" : {
        "type" : "OrderItem",
        "url" : "/services/data/v29.0/subjects/OrderItem/802D0000000CKoTIAW"
      },
      "Id" : "802D0000000CKoTIAW"
    } ]
  } ]
} ]
}
```

Add Order Products to an Existing Order

Here's an example of a PATCH request using the [Order](#) resource to add order products to an existing order.

Example usage

```
/services/data/v30.0/commerce/sale/order/801D0000000Frh8
```

Example request body

```
{
  "order": [
    {
      "attributes": {
        "type": "Order"
      },
      "Id": "801D0000000Frh8",
      "OrderItems": {
        "records": [
          {
            "attributes": {
              "type": "OrderItem"
            },
            "PricebookEntryId": "01uD00000001cCd1",
            "quantity": "1",
            "UnitPrice": "100",
            "orderId": "801D0000000Frh8"
          },

```

```

    {
      "attributes": {
        "type": "OrderItem"
      },
      "PricebookEntryId": "01uD0000001cCd1",
      "quantity": "2",
      "UnitPrice": "200",
      "orderId": "801D00000000Frh8"
    }
  ]
}

```

Example JSON response body

```

{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Order"
    },
    "OrderItems" : {
      "totalSize" : 2,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "OrderItem",
          "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CJtMIAW"
        },
        "Id" : "802D0000000CJtMIAW"
      }, {
        "attributes" : {
          "type" : "OrderItem",
          "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CJtNIAW"
        },
        "Id" : "802D0000000CJtNIAW"
      } ]
    } ]
}

```

Get Details About a Contract

Here's an example of a GET request using the [Contract-based Orders](#) resource to query details about a contract and its child orders, order products, and custom objects.

Example usage

```
/services/data/v30.0/commerce/sale/800D0000000PFHp
```

Example request body

None

Example JSON response body

```

{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Contract",
      "url" : "/services/data/v30.0/subjects/Contract/800D000000PFHpIAO"
    },
    "Id" : "800D000000PFHpIAO",
    "Orders" : {
      "totalSize" : 4,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "Order",
          "url" : "/services/data/v30.0/subjects/Order/801D000000FqzsIAC"
        },
        "Id" : "801D000000FqzsIAC",
        "OrderItems" : {
          "totalSize" : 3,
          "done" : true,
          "records" : [ {
            "attributes" : {
              "type" : "OrderItem",
              "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CJX0IAO"
            },
            "Id" : "802D0000000CJX0IAO"
          }, {
            "attributes" : {
              "type" : "OrderItem",
              "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CJYDIA4"
            },
            "Id" : "802D0000000CJYDIA4"
          }, {
            "attributes" : {
              "type" : "OrderItem",
              "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CKFCIA4"
            },
            "Id" : "802D0000000CKFCIA4"
          } ]
        },
        "Custom_Objects__r" : null
      }, {
        "attributes" : {
          "type" : "Order",
          "url" : "/services/data/v30.0/subjects/Order/801D0000000FpNEIAO"
        },
        "Id" : "801D0000000FpNEIAO",
        "OrderItems" : {
          "totalSize" : 3,
          "done" : true,
          "records" : [ {

```

```

    "attributes" : {
      "type" : "OrderItem",
      "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CIWBIA4"
    },
    "Id" : "802D0000000CIWBIA4"
  }, {
    "attributes" : {
      "type" : "OrderItem",
      "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CIWCIA4"
    },
    "Id" : "802D0000000CIWCIA4"
  }, {
    "attributes" : {
      "type" : "OrderItem",
      "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CIWDIA4"
    },
    "Id" : "802D0000000CIWDIA4"
  } ]
},
"Custom_Objects__r" : null
}, {
  "attributes" : {
    "type" : "Order",
    "url" : "/services/data/v30.0/subjects/Order/801D0000000FqztIAC"
  },
  "Id" : "801D0000000FqztIAC",
  "OrderItems" : null,
  "Custom_Objects__r" : null
}, {
  "attributes" : {
    "type" : "Order",
    "url" : "/services/data/v30.0/subjects/Order/801D0000000FpkNIAS"
  },
  "Id" : "801D0000000FpkNIAS",
  "OrderItems" : null,
  "Custom_Objects__r" : null
} ]
},
"Test_Contract1__r" : null
} ]
}

```

Get Details About an Order

Here's an example of a GET request using the [Order](#) resource to query details about an order and its order products and custom object records.

Example usage

```
/services/data/v30.0/commerce/sale/order/801D0000000FzsM
```

Example request body

None

Example JSON response body

```
{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Order",
      "url" : "/services/data/v30.0/subjects/Order/801D0000000FzsMIAS"
    },
    "Id" : "801D0000000FzsMIAS",
    "OrderItems" : {
      "totalSize" : 2,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "OrderItem",
          "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CKIHIA4"
        },
        "Id" : "802D0000000CKIHIA4"
      }, {
        "attributes" : {
          "type" : "OrderItem",
          "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CKIGIA4"
        },
        "Id" : "802D0000000CKIGIA4"
      } ]
    },
    "Custom_Objects__r" : null
  } ]
}
```

Filter Details About a Contract

Here's an example of a GET request using the [Contract-based Orders](#) resource to query a given contract's activated orders.

Example usage

```
/services/data/v30.0/commerce/sale/800D0000000PFL8?contract.orders.StatusCode='Activated'
```

Example request body

None

Example JSON response body

```
{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "Contract",
      "url" : "/services/data/v30.0/subjects/Contract/800D0000000PFHpIAO"
    },
    "Id" : "800D0000000PFHpIAO",
    "Orders" : {
```

```

    "totalSize" : 1,
    "done" : true,
    "records" : [ {
      "attributes" : {
        "type" : "Order",
        "url" : "/services/data/v30.0/subjects/Order/801D0000000FpNEIA0"
      },
      "Id" : "801D0000000FpNEIA0",
      "OrderItems" : {
        "totalSize" : 3,
        "done" : true,
        "records" : [ {
          "attributes" : {
            "type" : "OrderItem",
            "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CIWBIA4"
          },
          "Id" : "802D0000000CIWBIA4"
        }, {
          "attributes" : {
            "type" : "OrderItem",
            "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CIWCIA4"
          },
          "Id" : "802D0000000CIWCIA4"
        }, {
          "attributes" : {
            "type" : "OrderItem",
            "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CIWDIA4"
          },
          "Id" : "802D0000000CIWDIA4"
        } ]
      },
      "Custom_Objects__r" : null
    } ]
  },
  "Test_Contract1__r" : null
} ]
}

```

Filter Details About an Order

Here's an example of a GET request using the [Order](#) resource to query details for order products with a certain start date for a given order.

Example usage

```
/services/data/v30.0/commerce/sale/order/801D0000000FzsM?order.orderItems.effectiveDate=2013-08-05
```

Example request body

None

Example JSON response body

```

{
  "totalSize" : 1,
  "done" : true,
  "records" : [ {

```

```

    "attributes" : {
      "type" : "Order",
      "url" : "/services/data/v30.0/subjects/Order/801D0000000FzsMIAS"
    },
    "Id" : "801D0000000FzsMIAS",
    "OrderItems" : {
      "totalSize" : 1,
      "done" : true,
      "records" : [ {
        "attributes" : {
          "type" : "OrderItem",
          "url" : "/services/data/v30.0/subjects/OrderItem/802D0000000CKIHIA4"
        },
        "Id" : "802D0000000CKIHIA4"
      } ]
    },
    "Custom_Objects__r" : null
  } ]
}

```

Place Order REST API Resource Reference

Each Place Order REST API resource is a URI used with an HTTP method (such as GET).

Resources for the Place Order REST API are:

Resource	Supported HTTP Method	Description
<code>/services/data/latest API version/commerce/sale</code>	POST	Add new orders, order products, and custom objects to a new contract.
<code>/services/data/latest API version/commerce/sale/contract ID</code>	PATCH, GET	Add new orders, order products, and custom objects to an existing contract. Retrieve a contract's child orders, order products, and custom objects.
<code>/services/data/latest API version/commerce/sale/order</code>	POST	Add new order products and custom objects to a new order.
<code>/services/data/latest API version/commerce/sale/order/order ID</code>	PATCH, GET	Add new order products and custom objects to an existing order. Retrieve an order's child order products and custom objects.

Create Contract-based Orders

With this resource, you can create a new contract with orders and order products, as well as custom object records on the contract or order level.

Syntax

URI

```
/services/data/latest API version/commerce/sale
```

Available since release

30.0

Formats

JSON

HTTP methods

POST

Request body

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the new contract.	30.0
AccountId	String	Unique record identifier for the parent account.	30.0
Status	String	Status of the contract.	30.0
Orders	Orders[]	Child orders of the new contract.	30.0
<i>CustomObject__r</i>	Custom Objects[]	Child custom object records of the new contract.	30.0

Attributes

Property	Type	Description	Since Version
type	String	Format of the resource.	30.0

Orders

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the new order record.	30.0
Status	String	Status of the order.	30.0
Pricebook2Id	String	Unique record identifier for the associated price book.	30.0
OrderItems	Order Products[]	Child order products of the new order.	30.0
<i>CustomObject__r</i>	Custom Objects[]	Child custom object records of the new order.	30.0

Custom Object Records

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the custom object record.	30.0
Id	String	Unique record identifier.	30.0

Order Products

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the order product record.	30.0
PricebookEntryId	String	Unique record identifier for the associated price book entry.	30.0
quantity	Number	Number of units of the order product.	30.0
UnitPrice	Currency	The unit price for the order product.	30.0

Request parameters

None

Response body

Property	Type	Description
totalSize	Number	Total number of records added.
done	Boolean	When <code>true</code> , the operation was successful.
records	Records[]	Attributes and IDs of records.

Records

Property	Type	Description
attributes	Attributes	Type and URL for the new record.
Id	String	Unique identifier for the new record.

Attributes

Property	Type	Description
type	String	Format of the resource.
url	String	Resource URL.

Example

See [Add a Contract and Orders to an Existing Account](#) on page 3.

Usage

You can only create one contract at a time. You can't POST new orders onto an existing contract. For that, see the [Contract-based Orders](#) resource.

Contract-based Orders

With this resource, you can add one or more new orders to an existing contract, as well as order products and custom object records for each order, or you can retrieve data for a specific contract.

If available, GET method retrieves the contract's child orders and order products, as well as custom objects under the contract and orders.

Syntax

URI

```
/services/data/latest API version/commerce/sale/contractId
```

For retrieving filtered data:

```
/services/data/latest API version/commerce/sale/contract ID?contract.orders.field name=value
```

Available since release

30.0

Formats

JSON

HTTP methods

PATCH, GET

Request parameters

- You can use parameters for all standard and custom fields on contracts, orders, order products, and any custom objects directly related to these objects.
- The parameters must be fully qualified. For example: **objectname.relationshipname.fieldname=value**.
 - Object name must be all lower-case.
 - Relationship name must match what is defined on the object and is case-sensitive.
 - Field name isn't case sensitive.
 - If the value is a string, it must be encased in single quotation marks. If the value is a number, it must not be encased. If the value is a date, it should be in the YYYY-MM-DD format.
- You can use multiple parameter fields, separated by "&", to make more detailed filters. For example:

```
/services/data/v30.0/commerce/sale/{contractId?contract.status='Activated'  
&contract.Orders.status='Draft'&contract.Orders.OrderItems.unitprice=300
```

The following aren't supported:

- Arrays of values. For example: `contract.orders.Status='Activated','Draft'`.
- Operators: `>`, `>=`, `<` and `<=`
- The OR condition

Parameters	Description	Since Version
<code>contract</code>	The object name of the record being filtered. In this resource, this is always <code>contract</code> .	30.0
<code>orders</code>	The relationship name of the field that the order's data will be filtered by. In this resource, this is always <code>orders</code> .	30.0
field name	The field whose value to filter by. For example, if you want to only retrieve orders with a status category, the field name is <code>StatusCode</code> .	30.0
value	The value to filter by. For example, if you want to only retrieve orders with a status category of <code>Activated</code> , the value is <code>Activated</code> .	30.0

Request body

Property	Type	Description	Since Version
<code>attributes</code>	Attributes	Type of the contract.	30.0
<code>Id</code>	String	Unique contract identifier.	30.0
<code>Orders</code>	Orders[]	Child orders of the contract.	30.0

Attributes

Property	Type	Description	Since Version
<code>type</code>	String	Format of the resource.	30.0

Order Records

Property	Type	Description	Since Version
<code>attributes</code>	Attributes	Type of the order record.	30.0
<code>Status</code>	String	Status of the order.	30.0
<code>contractId</code>	String	Unique record identifier for the parent contract.	30.0
<code>pricebook2Id</code>	String	Unique record identifier for the associated price book.	30.0
<code>OrderItems</code>	OrderProducts[]	Child order products of the order.	30.0

Property	Type	Description	Since Version
<i>CustomObject__r</i>	Custom Objects[]	Child custom object records of the order.	30.0

Order Products

Property	Type	Description	Since Version
<i>attributes</i>	Attributes	Type and URL of the order product.	30.0
<i>PricebookEntryId</i>	String	Unique record identifier for the associated price book entry.	30.0
<i>quantity</i>	Number	Number of units of the order product.	30.0
<i>UnitPrice</i>	Currency	The unit price for the order product.	30.0

Custom Objects

Property	Type	Description	Since Version
<i>attributes</i>	Attributes	Type and URL of the custom object.	30.0
<i>Id</i>	String	Unique custom object identifier.	30.0

Response body

Property	Type	Description
<i>totalSize</i>	Number	Total number of records retrieved.
<i>done</i>	Boolean	When <code>true</code> , the operation was successful.
<i>records</i>	Records[]	Attributes and ID of contract record.

Records

Property	Type	Description
<i>attributes</i>	Attributes	Type and URL of the record.
<i>Id</i>	String	Unique contract identifier.

Attributes

Property	Type	Description
type	String	Format of the resource.
url	String	Resource URL.

Examples

- [Add Orders to an Existing Contract](#) on page 6
- [Get Details About a Contract](#) on page 10
- [Filter Details About a Contract](#) on page 13

Create Order

With this resource, you can create a new order with order products and custom objects.

If you don't want to add the order to a contract, you can add it directly to an account. You can only create one new order per call. The request body must have either an account or a contract as its parent record, and it must have a reference to a price book.

Syntax**URI**

```
/services/data/latest API version/commerce/sale/order
```

Available since release

30.0

Formats

JSON

HTTP methods

POST

Request body

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the new order.	30.0
orderItems	Order Products []	Child order products of the new order.	30.0
<i>CustomObject__r</i>	Custom Objects []	Child custom object records of the new order.	30.0

Attributes

Property	Type	Description	Since Version
<code>type</code>	String	Format of the resource.	30.0
<code>url</code>	String	Resource URL.	30.0

Order Products

Property	Type	Description	Since Version
<code>attributes</code>	Attributes	Type and URL of the order product.	30.0
<code>PricebookEntryId</code>	String	Unique record identifier for the associated price book entry.	30.0
<code>quantity</code>	Number	Number of units of the order product.	30.0
<code>UnitPrice</code>	Currency	The unit price for the order product.	30.0
<code>Id</code>	String	Unique order product identifier.	30.0

Custom Objects

Property	Type	Description	Since Version
<code>attributes</code>	Attributes	Type and URL of the custom object.	30.0
<code>Id</code>	String	Unique custom object identifier.	30.0

Request parameters

None

Response body

Property	Type	Description
<code>totalSize</code>	Number	Total number of records retrieved.
<code>done</code>	Boolean	When <code>true</code> , the operation was successful.
<code>records</code>	Records[]	Attributes and ID of contract record.

Records

Property	Type	Description
attributes	Attributes on page 23	Type and URL of the record.
Id	String	Unique contract identifier.

Attributes

Property	Type	Description
type	String	Format of the resource.
url	String	Resource URL.

Example

See [Add an Order to an Existing Account](#) on page 5.

Order

Use this resource to add one or more new order products and custom object records to an existing order or to retrieve data for a specific order.

You can only PATCH one order at a time.

If available, GET method retrieves the orders' child order products and custom objects under the order or order products.

Syntax

URI

```
/services/data/latest API version/commerce/sale/order/order ID
```

For retrieving filtered data:

```
/services/data/latest API version/commerce/sale/order/order ID?order.orderItems.field name=value
```

Available since release

30.0

Formats

JSON

HTTP methods

POST

Request body

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the order.	30.0
OrderItems	Order Products	Child order products of the new order.	30.0
<i>CustomObject__r</i>	Custom Object	Child custom object records of the new order.	30.0

Attributes

Property	Type	Description	Since Version
type	String	Format of the resource.	30.0

Order Products

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the order product.	30.0
PricebookEntryId	String	Unique record identifier for the associated price book entry.	30.0
quantity	Number	Number of units of the order product.	30.0
UnitPrice	Currency	The unit price for the order product.	30.0
orderId	String	Unique record identifier for the parent order.	30.0
Id	String	Unique order product identifier.	30.0

Custom Objects

Property	Type	Description	Since Version
attributes	Attributes	Type and URL of the custom object.	30.0
Id	String	Unique custom object identifier.	30.0

Request parameters

- You can use parameters for all standard and custom fields on contracts, orders, order products, and any custom objects directly related to these objects.
- The parameters must be fully qualified. For example: ***objectname.relationshipname.fieldname=value***.

- Object name must be all lower-case.
 - Relationship name must match what is defined on the object and is case-sensitive.
 - Field name isn't case sensitive.
 - If the value is a string, it must be encased in single quotation marks. If the value is a number, it must not be encased. If the value is a date, it should be in the YYYY-MM-DD format.
- You can use multiple parameter fields, separated by "&", to make more detailed filters. For example:

```
/services/data/v30.0/commerce/sale/{contractId}?contract.status='Activated'
&contract.Orders.status='Draft'&contract.Orders.OrderItems.unitprice=300
```

The following aren't supported.

- Arrays of values. For example: `order.orderItems.effectiveDate=2013-01-01,2013-01-02`.
- Operators: `>`, `>=`, `<` and `<=`
- The OR condition

Parameters	Description
<code>order</code>	The object name of the record being filtered. In this resource, this is always <code>order</code> .
<code>orderItems</code>	The relationship name of the field that the order's data will be filtered by. In this resource, this is always <code>orderItems</code> .
field name	The field whose value to filter by. For example, if you want to only retrieve order products with a certain start date, the field name is <code>effectivedate</code> .
value	The value to filter by. For example, if you want to only retrieve order products that started on January 1, 2013, the value is <code>2013-01-01</code> .

Response body

Property	Type	Description
<code>totalSize</code>	Number	Total number of records listed.
<code>records</code>	Records[]	Attributes and IDs of the new records.

Records

Property	Type	Description
<code>attributes</code>	Attributes	Type and URL for the record.
<code>Id</code>	String	Unique record identifier.

Attributes

Property	Type	Description
type	String	Format of the resource.
url	URI	Resource URL.

Examples

- [Add Order Products to an Existing Order](#) on page 9
- [Get Details About an Order](#) on page 12
- [Filter Details About an Order](#) on page 14