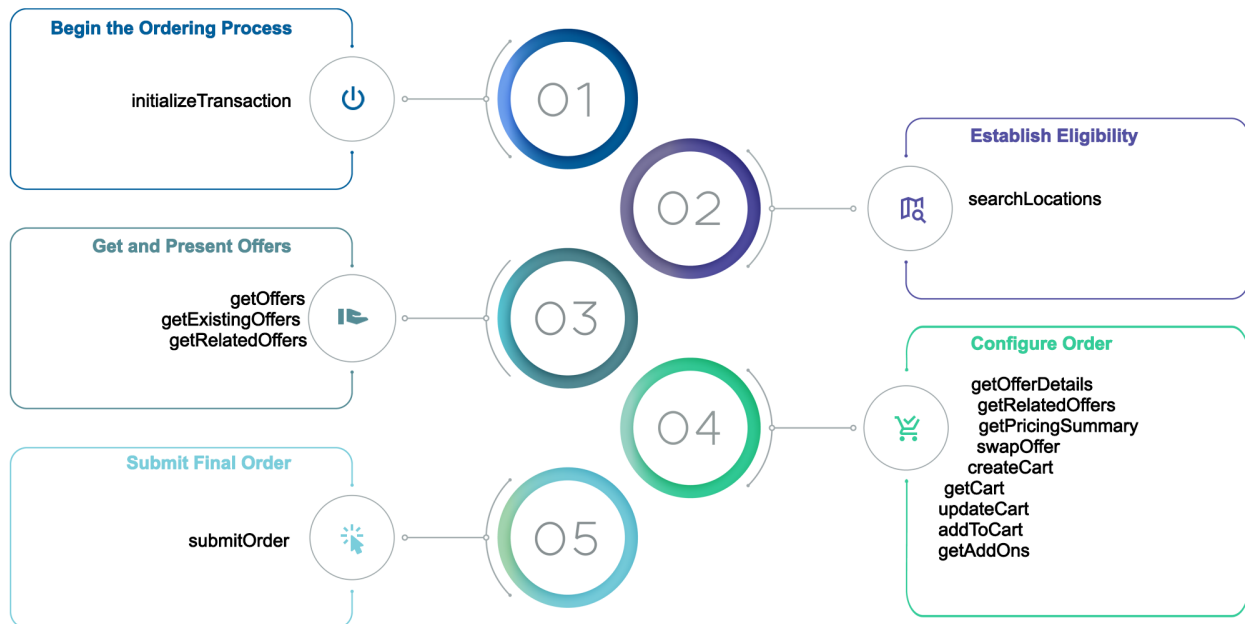# ✷ ORDERING PROCESS

In this section, we'll take a step-by-step look at the ordering process. You'll learn how to use the OCBO API to verify eligibility of a customer's address, get and present an offer, configure or reconfigure the offer, order services, arrange installation, and troubleshoot any issues that might arise. You'll also see the expected response from a successful API call and the error messages returned when something goes awry. The sections that follow are numbered to correspond with the steps shown in the following diagram.

Begin the Ordering Process

initializeTransaction

O1

Establish Eligibility

searchLocations

O2

Get and Present Offers

getOffers
getExistingOffers
getRelatedOffers

O3

Configure Order

getOfferDetails
getRelatedOffers
getPricingSummary
swapOffer
createCart
getCart
updateCart
addToCart
getAddOns

O4

Submit Final Order

submitOrder

O5

🗗 Begin the Ordering Process

Once the client has an access token, the first step in the ordering process is to establish a session and initialize a transaction. This step requires only a single endpoint.

POST /session

The ordering process begins when the client initializes a new purchasing transaction ID (called the session-id) and returns it in a response header. The session-id expedites logging and reporting by creating an association among all calls made during a given transaction.

/session | initializeTransaction

Function: The `/session` endpoint must be called before any other endpoint is consumed. During onboarding, the client generates the credentials necessary to request an OAuth2 access token. OAuth2 access tokens allow varying levels of access, or scopes, for different groups of users.

Typical use: The initializeTransaction operation is the first step in the ordering process.

Good to know: The ordering process can't proceed without a valid session-id.

Details: The purpose of the `/session` endpoint is to create a session-id (for example, "758fc320-af96-4faf-8af7-536e7cac5775") and corresponding state object. In the remaining service calls in the transaction, the client-id and session-id are passed in the header and validated.

Resource

Name	Method	Description
/session	POST	Validates client and initializes a purchasing transaction.

operationId	Description
initializeTransaction	Initializes a new purchasing transaction ID (session-id), validates the client-id, and returns an OAuth2 access token.

Header Parameters*

Object	Type	Description	Example	Required?
authorization	string	An OAuth2 token generated by the authorization server. This token is passed to all remaining API calls in the ordering session.	eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJMb2NhbERvbWFpblxcRW50RXpZUFQSU0tanlZdWYxaDc1eFJGGdTFscml0UEdYeHNIOGlVY09hQXhCOE9VTnBXYSIsInNjb3BlIjoiblszdGVyaW5lLmFsbCIsmlzcyI6IkVOVEFQSVBAiLCJHcmFudElEIjoiZnRyZWlqY2M2ZTRwIiwiZXhwIjoxNTk1MjExSTA0LCJpYXQiOjE1OTMyNzE0MDQsImNsaWVudF9pZCI6IkVudGVycHJpc2VBUEINLWp5WXVmMWg3NXhSRnUxbHJpdFBBSWHhzSDhpVWNPYUF4QjhPVU5wV2EifQ.Rj4aDOdCU1YFfxyYHpSW-7cEU75v7lCw0UKFFA–Y6o	Yes
client-id	string	An ID tied to the client application with which the request originated. This ID is passed downstream.	EnterpriseAPIM-RPuj1Q5nWk6bDC9fP3VpsjRJTmnLshfqo7cbq437	Yes

Only these two header parameters are required. Subsequent calls require all six headers shown in .