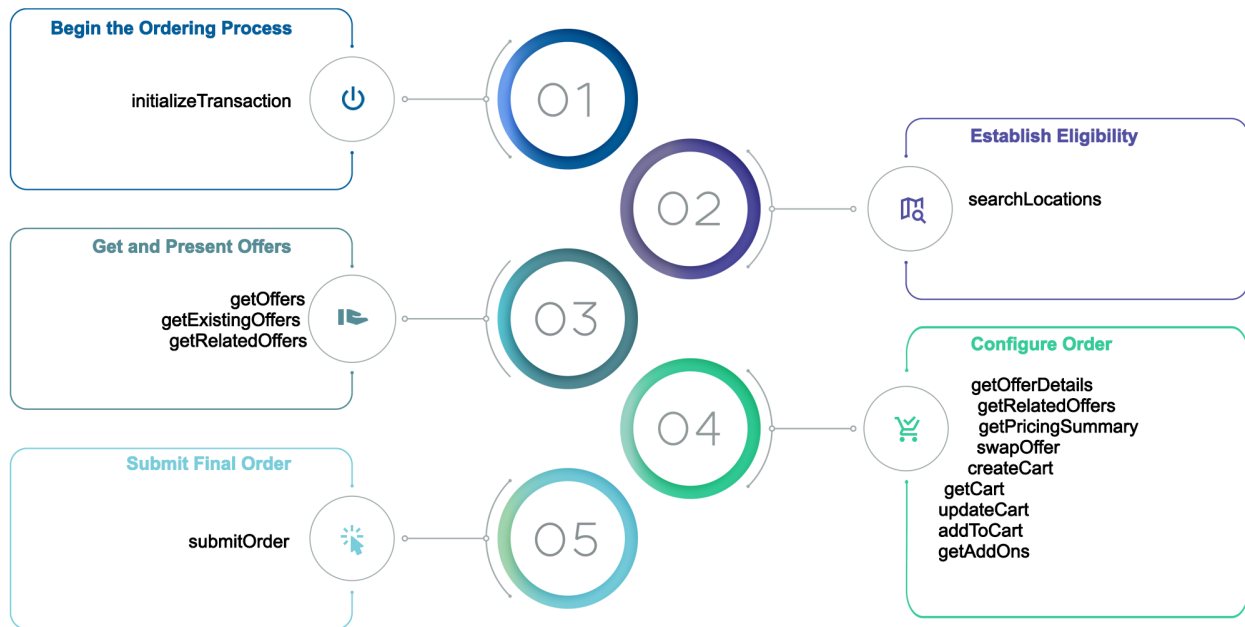


ORDERING PROCESS

In this section, we'll take a step-by-step look at the ordering process. You'll learn how to use the Omni-Channel Back Office API (OCBO) 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

Once the client has an [access token](#), the first step in the ordering process is establishing a session and initializing a transaction. This step requires only a single endpoint and [method](#):

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 the calls made during a transaction.

[/session](#) | [initializeTransaction](#)



Function: This first endpoint must be called before any other OCBO endpoint can be consumed. During onboarding, Spectrum will work with the client to establish a client identifier and generate the credentials used to request an [OAuth](#) access token.

Typical use: The initializeTransaction operation is used to start the ordering process.

Good to know: Without first establishing the session, the client would not have valid session-id and subsequent operations in the ordering process would fail.



Details: The purpose of the /session endpoint is to create the session-id 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.

OAuth access tokens can allow different levels of access for different groups of users. The access granted to a given user group is called a [scope](#). OCBO, however, has only one scope: all clients have full access.

Resource

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

operationId	Description
initializeTransaction	Validates the client-id contained in a JWT token signed by the client. Initializes a new purchasing transaction ID (session-id). Returns both the client-id and session-id in a signed JSON web token. The session-id ties together all of the different calls made during a given transaction. The client-id and session-id must be passed into subsequent service calls using their respective header parameters. Both IDs will be validated in the remaining calls based on the credentials established in this call.

Header Parameters*

Object	Type	Description	Example	Required?
authorization	string	An OAuth2 token generated by the authorization server. This token is passed to all other API calls in this ordering session.	eyJhbGciOiJIUzI1NiJ9.eyJzdWUiOiJMb2NhbERvbWFpblxcRW50ZXJwcmVzZUFQSU0tanlZdWYxaDc1eFJGdTfscml0UEdYeHNIOGIVY09hQXhCOE9VTnBXYSIsbnNjb3BljoibGlzdGVyaW5lImFsYXNjaXNlbnNpdjE1OTUyNzE0MDQsImNsaWVudF9pZCI6IkVudGVycHJpc2VBUEINLWp5WXVmMWg3NXhSRnUxbHJpdFBHWHhzSDhpVWNPYUF4QjhPVU5wV2EifQ.Rj4aDOdCU1YFfyYHpSW-7cEU75v7lCw0UKFFA-Y6o	Yes
client-id	string	An ID identifying the client application originating the request. This ID is passed downstream to other internal API calls.	EnterpriseAPIM-CHTR-azrLz7unmLRCsl	Yes

*Only these two header parameters are required. Subsequent calls require all six headers shown in [Table 1](#).

Request Body

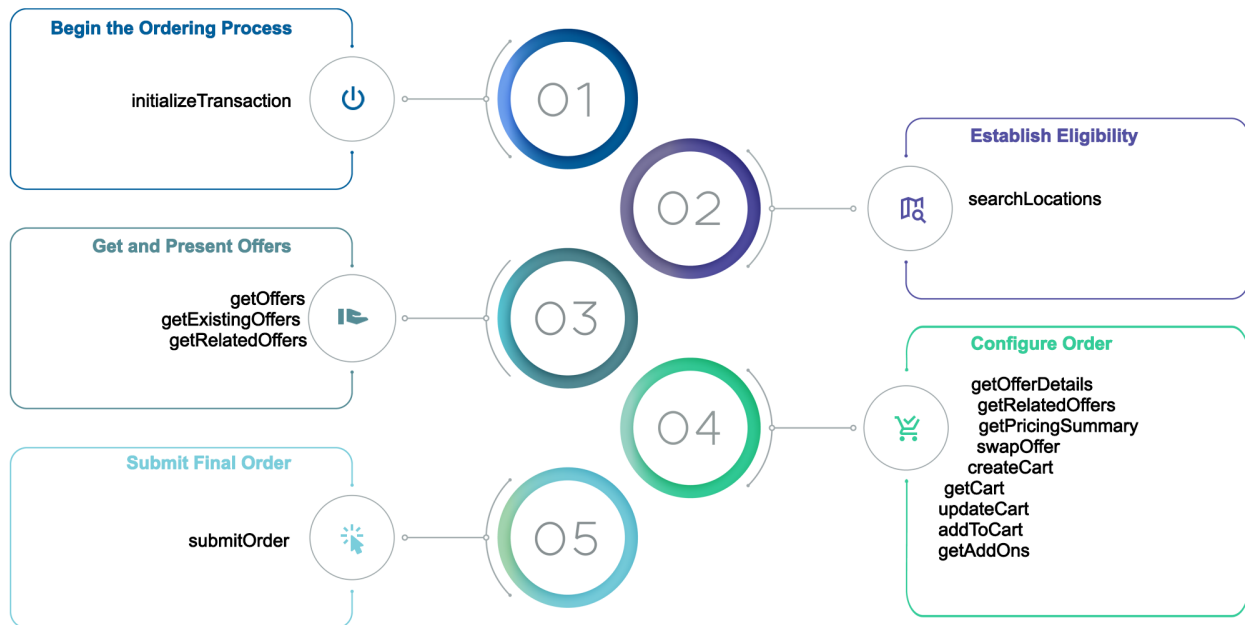
None—header parameters only.

Response Details: GET /session

HTTP Code	Message	Error Object		Details
		Client Code	Tech Code	
200				
	OK	NA	NA	<ul style="list-style-type: none">Returns a JWT access token containing the client-id passed in the request as well as a new session-id.The service returns 200 OK.
400				
	Bad Request	CHTR-4504	TECHMSG-0001	<ul style="list-style-type: none">Returned when client access check and validation pass but findLocation service reports only partial content.The service returns 400 Bad Request.
403				
	Forbidden	CHTR-1018	TECHMSG-0006	<ul style="list-style-type: none">The OCBO client permission check finds that the client lacks authorization to call this service.The service returns 403 Forbidden.

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.



1 Begin the Ordering Process

Once the client has an [access token](#), the first step in the ordering process is establishing a session and initializing a transaction. This step requires only a single endpoint and [method](#):

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 the calls made during a transaction.

[/session](#) | [initializeTransaction](#)



Function: This first endpoint must be called before any other OCBO endpoint can be consumed. During onboarding, Spectrum will work with the client to establish a client identifier and generate the credentials used to request an [OAuth](#) access token.

Typical use: The initializeTransaction operation is used to start the ordering process.

Good to know: Without first establishing the session, the client would not have valid session-id and subsequent operations in the ordering process would fail.



Details: The purpose of the [/session](#) endpoint is to create the session-id 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.

OAuth access tokens can allow different levels of access for different groups of users. The access granted to a given user group is called a [scope](#). OCBO, however, has only one scope: all clients have full access.

Resource

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

operationId	Description
initializeTransaction	Validates the client-id contained in a JWT token signed by the client. Initializes a new purchasing transaction ID (session-id). Returns both the client-id and session-id in a JWT token signed by Spectrum. The session-id is used for logging and reporting purposes to tie together all of the different calls made during a given transaction. The client-id and session-id must be passed into subsequent service calls using their respective header parameters. Both IDs will be validated in the remaining calls based on the credentials established in this call.

Header Parameters*

Object	Type	Description	Example	Required?
authorization	string	An OAuth2 token generated by the authorization server. This token will be passed to all other API calls in this ordering session.	eyJhbGciOiJIUzI1NiJ9.eyJzdWUiOiJMb2NhbERvbWVpblxcRW50ZXJwcmlzZUFQSU0tanlZdWYxaDc1eFJGdTFscml0UEdYeHNIOGIVY09hQXhCOE9VTnBXYSIsInNjb3BlIjoibGlzdGVyaW5lImFsbCIsImIzcyI6IkVOVEFQSVAiLCJHcmFudEIEljoiznRyZWlqY2M2ZTRwliwiZXhwIjoxNTk1MjcxNTA0LCJpYXQiOiJ1OTUyNzE0MDQsImNsaWVudF9pZCI6IkVudGVycHJpc2VBUEINLWp5WXVmMWg3NXhSRnUxbHJpdFBHWHhzSDhpVWNPYUF4QjhPVU5wV2EifQ.Rj4aDOdCU1YffxyYHpSW-7cEU75v7ICw0UKFFA-Y6o	Yes
client-id	string	An ID identifying the originating client application of the request. This ID will be passed downstream to other internal API calls unchanged.	EnterpriseAPIM-ABunchOfAlphaNumericCharacters	Yes

*Only these two header parameters are required. Subsequent calls require all six headers shown in [Table 1](#).

Request Body

None—header parameter only.

Response Details: GET /session

HTTP Code	Message	Error Object		Details
		Client Code	Tech Code	
200	OK	NA	NA	<ul style="list-style-type: none">Returns a JWT access token containing the client-id passed in the request, as well as a new session-id.The service returns 200 OK.
400	Bad Request	CHTR-4504	TECHMSG-0001	<ul style="list-style-type: none">Returned when client access check and validation pass but the findLocation service reports only partial content.The service returns 400 Bad Request.
403	Forbidden	CHTR-1018	TECHMSG-0006	<ul style="list-style-type: none">The OCBO client permission check finds that the client lacks authorization to call this service.The service returns 403 Forbidden.