CHECKOUT

The checkout process is a series of steps a customer follows to purchase the items on an ecommerce store. We define its beginning when a customer creates a shopping cart using any frontend platform, and we define its end when the customer confirms the shipping.

Fields

Field	Description
ID	Spring boot JPA identifier
Cart	Container object of Line-item objects It has the next fields: - Id - Items - Instructions - Discount - Price
Customer	Reference to persisted customer object, it is populated when checkout process has ended, meanwhile it will be stored on customer session.
Origin	Location object to store customer origin information.
Destination	Location object to store customer shipping address.
Store-variant	Persisted object that stores information about the restaurant. It is a mandatory object even if we are on the initially steps of a checkout.
Payment	Customer payment related information.
Created at	Instant based object which has the current date time when customer started a checkout process.
Price	CheckoutPrice object that stores pricing information of the checkout object.
Finished at	Instant based object that’s populated when customer ends the checkout process.
Cancelled at	Instant based object that’s populated when a customer cancels the checkout process.

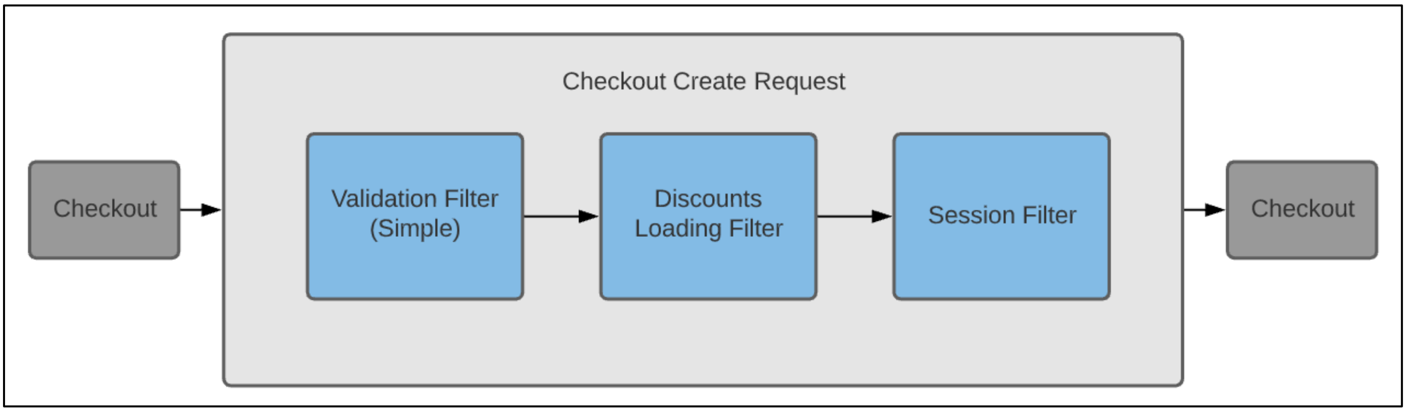
Source device	Handler that Tongue uses to distinct if a customer has come from a web or mobile platform.
Currency	Customer's country currency.

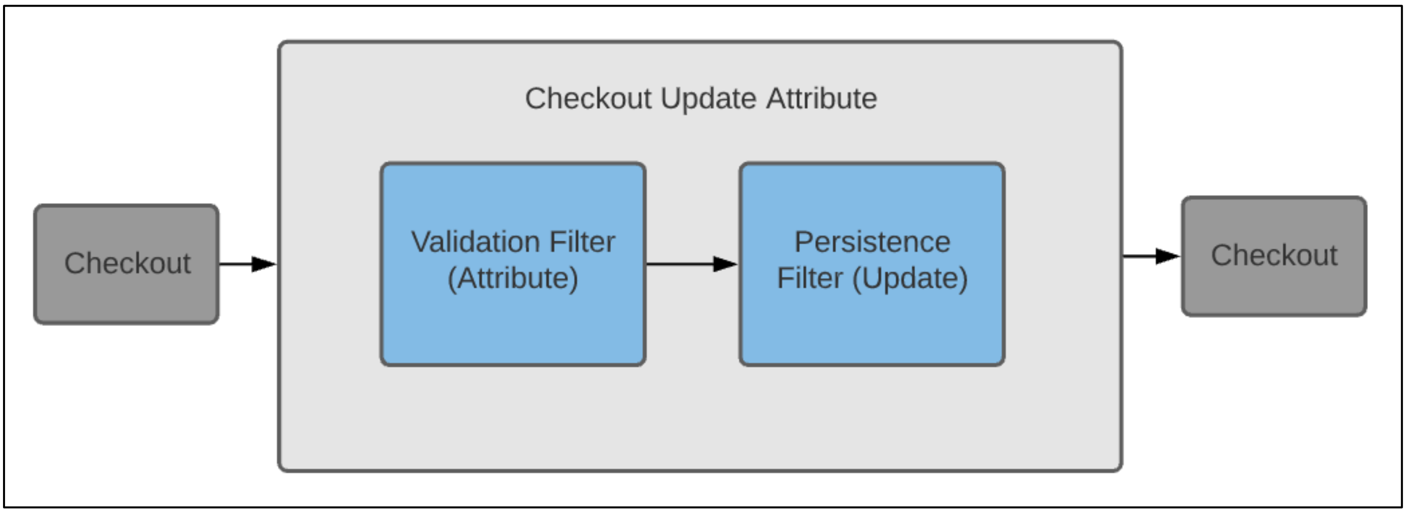
End points

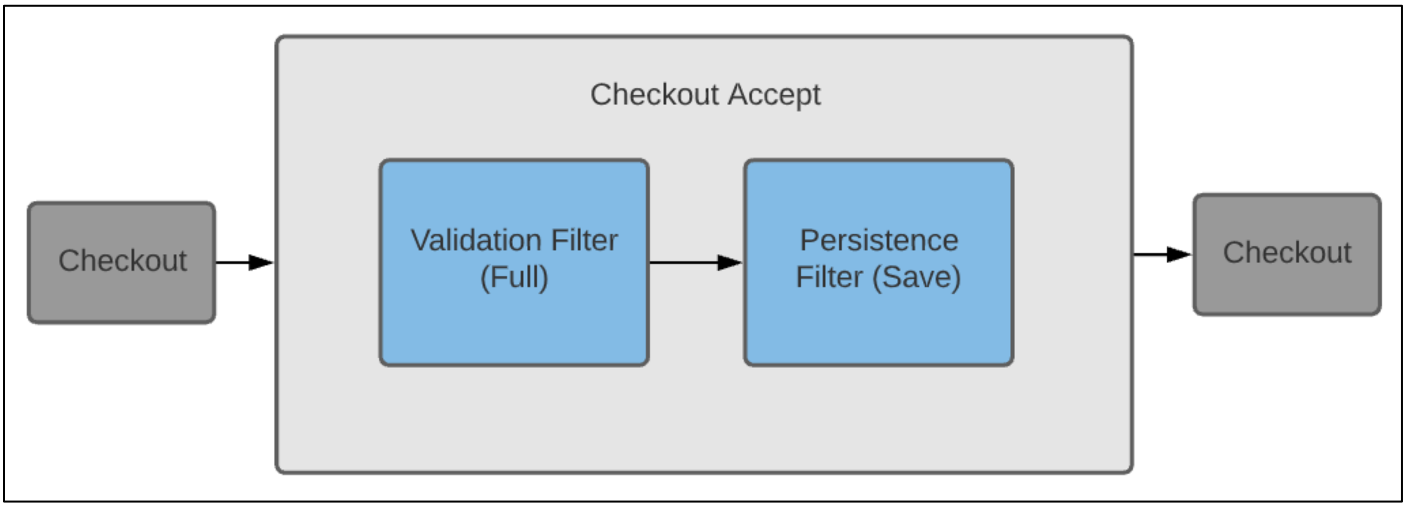
Endpoint	Parameters	Description
checkout/create	Customer must send a Checkout object in the request body, with the following fields populated: - storeVariant - origin - cart (Populated with at least one line-item)	When a customer accesses this endpoint, it must send a checkout object with the quoted objects populated at least by the store variant id for the store variant object, origin google place id for the origin object and a non-empty cart object. After the server receives this object, it calls the internal Checkout Create Request process.
checkout/update	Customer must send a CheckoutAttribute object in the request body. Supported attribute objects are: - cart - destination - payment - origin Customer must populate CheckoutAttributeName filed with any of these constants: CART, DESTINATION, PAYMENT, ORIGIN. And the corresponding attribute object.	This endpoint is accessed when a customer needs to update the current Checkout object stored on his session. He should send only the attribute with the corresponding Checkout Attribute Name constant. Server will validate mainly by id's validation and internal object rules such as discount expiring time. Once the server receives this object, it will update the current Checkout on session and after that it will return the updated checkout object to user. Checkout update process is showed below.
checkout/cancel	Nothing. The checkout object deleted is the one stored on the current session.	This endpoint is called when customer wants to end the current checkout, either because he wants to add products from a different store or simply wants to clean the current shopping cart list.

checkout/confirm	Nothing. The checkout object stored is the one on session.	This endpoint is called only one time by checkout object when a customer wants to start a shipping process. It returns an Order object to the client and asks any of the nearest drivers to deliver it. Corresponding process is showed below.

Checkout filters




Checkout Update Attribute
Checkout
Validation Filter (Attribute)
Persistence Filter (Update)
Checkout



Written by: alexanderommel