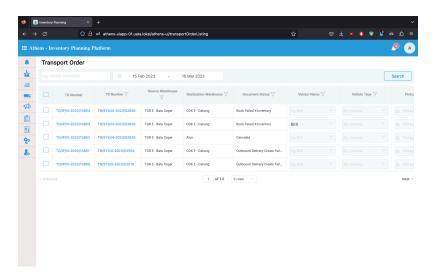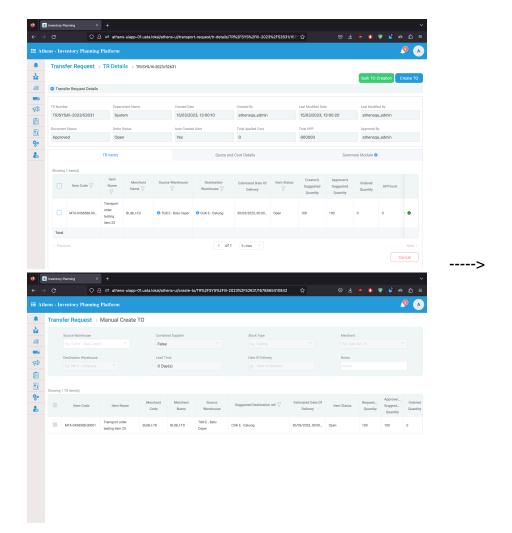# Transport Order Flow

View TOs

- Multiple APIs are called here to populate the filter menus i.e. warehouse, vehicle type etc.
- The TOs' are fetched based on the date, default is a months' duration from the current date.
- We can also filter based on estimated delivery, pickup date, TR number etc.
- Warehouses and Vehicles are fetched from `CommonDataCommandController` (CDCC)
- The TO details are fetched from `TransportOrderCommandController` (TOCC).

Create TO

----->

- The user can go through a Bulk TO Creation flow where the source & destination warehouses, estimated date of delivery and other details will be the same details as provided in the TR, they cannot be overridden, the only details under their control would be the logistics details.
- In Manual Creation of TO, the details can be overridden by the user.
- After creating the TO and filling in the logistic details, two APIs are called, the save and submit API.
- The save API validates and saves the TO into db with audit.
- The submit API updates the TO document status and publishes a message to a kafka topic for OMS to proceed with the order.
- For Bulk TO Creation, the count API is called first, this API converts all the TRs to TOs by mapping the common values of TR and TO i.e. source and destination warehouse, TR number, estimated Date of Delivery etc.

View all the Transport Orders

start

view TOs
validate user and permissions → find all TOs with filter → convert to TransportOrderResponse → convert to paged response

Create a TO after successful TR Approval

start

Create TO
bulk TO creation ? —yes→ count API
enter all required details
save TO
submit TO

count API
converts all the requests in a TR to TOs
validate markForDelete → validate document status and order status → convert to TO
postgres

convert to TO
validate all items → calculate total HPF for each request → create a new TO details response object for each request
postgres

save TOs
postgres
validate TR → validate number of items → validate item and document status → validate all item details and warehouses → existing TO ? —yes→ check if we can update TO
—no→
populate TO reference → set lead time, default = 1 → is TO number null ? —yes→ insert TO with audit
—no→ update TO with audit
postgres
mongo

submit TO
postgres
validate TO and TR → validate TRDoc status, TODoc status, order status, item status → validate item details and metrics
update all statuses → update TO with audit → publish a kafka message to DMS topic → update TR reference → update item metrics and status
create alert
postgres
mongo

order status lifecycle
initially, if ordered qty == 0
OPEN
if ordered qty == reserved qty
—no→ PARTIALLY ORDERED
—yes→ ORDERED