For all v2 Api and few v1 endpoints which are Address, Billings, Account Details and Orders(Order History) we need to pass bearer token in Authorization

To generate token in postman we have an api in Firebase folder in which v2-AuthToken Validation , we have to pass valid email , password , returnSecureToken which will return token and store globally in postman and applied to all the required APIs

**Search, On\_Search APIs need to be checked for postman for, searching for an item by delivery location:**

* The BIAB BAP Hackathon collection must have set the ‘base\_url’ under ‘variables’ and initial value and current value must be set as <https://qa.api.box.beckn.org/bap>
* For this api user Authentication is not mandatory
* The search by item and location in Search API needs to be tested at the postman for this step, the command {{base\_url}}/client/v1/search needs to be run under POST in Postman by clicking ‘send’
* There will be search string and delivery location values in the body of the Search API
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message id must be entered in the ‘Params’ (parameters) part of the On\_Search API
* GET On-Search API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* The context containing domain country etc is seen again, here it is observed that the message\_id is same as the message\_id of copied from result of running POST Search API
* Transaction\_id, message\_id are some values seen in ‘context’. The ‘message’ in the result contains the providers with their respective id, descriptor, price and other parameters.

**Add item to cart and On Get Quote API:**

The Add item to cart under Client-v2 Quote needs to be tested at the postman to add items to cart, the command {{base\_url}}/client/v2/get\_quote and add Bearer-Token in Authorization in needs to be run under POST in Postman by clicking ‘send’

* There will be array object which has context,message, cart, items, bpp provider seen in the body of the Add item to cart API
* For this api user Authentication is mandatory for which we need to pass Bearer token in Authorization tab
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a array object which has message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message id from objects and must be entered in the ‘Params’ (messageIds) with comma separated as a part of the On Get Quote API
* GET On Get Quote API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* The context containing domain country etc is seen again, here it is observed that the message\_id is same as the message\_id of copied from result of running POST Add item to cart API
* Transaction\_id, message\_id are some values seen in ‘context’. The ‘message’ in the result contains the quote, provider id, descriptor, provider location and other parameters.

**Search by Category: Test By Category and On\_Search APIs**

* The search string with delivery location, category id category name are seen in Client-v2 By Category API under Search folder, the command {{base\_url}}/client/v1/ needs to be run under POST in Postman by clicking ‘send’
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message id must be entered in the ‘Params’ (parameters) part of the By Category API
* GET On\_Search API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* The context containing domain country etc is seen again, here it is observed that the message\_id is same as the message\_id of copied from result of running POST By Category API
* Transaction\_id, message\_id are some values seen in ‘context’. The ‘message’ in the result contains the types of catalogs.

**Billing and Shipping details : Test Initialize and On Initialize API:**

The Initialize Order API under Client-v2 Initialize folder needs to be tested at the postman to initialize an order, the command {{base\_url}}/client/v2/initialize\_order and and add Bearer-Token than needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the Initialize Order API to know more about the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a array of object which has message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message ids must be entered in the ‘Params’ (messageIds) with comma separated and also add Bearer-Token to the On Initialize API
* GET On Initialize API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* The context containing domain country etc is seen again, here it is observed that the message\_id is same as the message\_id of copied from result of running POST Initialize Order API
* Transaction\_id, message\_id are some values seen in ‘context’. The ‘message’ in the result contains the order, order provider,provider location, billing and delivery information

**Proceed to pay: Test Client confirm\_order and client onconfirm\_order API:**

The Client Confirm\_Order API under Client-v2 confirm folder needs to be tested at the postman to initialize an order, the command {{base\_url}}/client/v2/confirm\_order and also add Bearer-Token which then needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the confirm order API to know the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a array of object which has message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message id must be entered in the ‘Params’ (messageIds) with comma separated and also add Bearer-Token to the on\_confirm API
* GET on\_confirm API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* Transaction\_id, message\_id are some values seen in ‘context’. The context containing domain country etc is seen again, here it is observed that the message\_id is same as the message\_id of copied from result of running POST confirm API

**Order status: Test Client status\_order and client onstatus\_order API:**

The Client order\_status API under Client-v2 Status folder needs to be tested at the postman to initialize an order, the command {{base\_url}}/client/v2/order\_status and also add Bearer-Token needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the order\_status API to know the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a array of object which has message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message id must be entered in the ‘Params’ (messageIds) with comma separated and also add Bearer-Token to the on\_order\_status API
* GET on\_order\_status API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* Transaction\_id, message\_id are some values seen in ‘context’. The context containing domain country etc is seen again, here it is observed that the message\_id is the same as the message\_id copied from the result of running POST order\_status API. In ‘message’ order, provider, billing and fulfillment, item, payment values would be seen

**Track order: Test Client track and client ontrack\_order API**

The Client track\_orderAPI under Client-v2 Track folder needs to be tested at the postman to initialize an order, the command {{base\_url}}/client/v2/track and also add Bearer-Token which needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the track API to know the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be a array of object which has message acknowledgement status along with the context (containing domain, country, bap\_id, bap\_url) with transaction\_id and message\_id
* Copy the message\_id from here
* This copied message id must be entered in the ‘Params’ (messageIds) with comma separated and also add Bearer-Token to the on\_track API
* GET on\_track API needs to be run by clicking ‘send’, result is observed in bottom half of the page
* Transaction\_id, message\_id are some values seen in ‘context’. The context containing domain country etc is seen again, here it is observed that the message\_id is the same as the message\_id copied from the result of running POST track API. In ‘message’ status, url will be seen

**Order History: Test Order History API**

The Client orders history API under Client-v2 Orders folder needs to be tested at the postman to get order history information, the command {{base\_url}}/client/v1/order and also add Bearer-Token which needs to be run under GET in Postman by clicking ‘send’

* Look at the params of the order API to know the fields in it
* We can either pass orderId param to see the order specific history
* Or We can pass parentOrderId to see all the individual orders under parent Order
* We can handle pagination using skip and limit
* The result of sending of the above mentioned GET command can be seen at the bottom half of the page
* Response has array object in which each object has provider, items array, billing , fulfillment, quote, payment, id, state, transation\_id,message\_id, parent\_order\_id

**Delivery Address: Test Delivery Address API**

The Client delivery address API under Client-v2 Accounts/Delivery Address folder needs to be tested at the postman to add delivery address, the command {{base\_url}}/client/v1/delivery\_address and also add Bearer-Token which needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the delivery\_address API to know the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be object which has id, descriptor, gps, default\_address, address fields
* GET delivery\_address API needs to be run with only Bearer-Token to get all the delivery address by clicking ‘send’, result is observed in bottom half of the page

**Add Billings: Test Billings API**

The Client billing address API under Client-v2 Accounts/Billings folder needs to be tested at the postman to add delivery address, the command {{base\_url}}/client/v1/billing\_details and also add Bearer-Token which needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the billing\_details API to know the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be object which has id, name, phone, organization, address, email, tax\_number, location\_id fields
* GET billing\_details API needs to be run with only Bearer-Token to get all the billing details by clicking ‘send’, result is observed in bottom half of the page

**Update Account Info: Test Account Info API**

The Client Account Info API under Client-v2 Accounts/AccountDetails folder needs to be tested at the postman to update user account information, the command {{base\_url}}/client/v1/account\_details and also add Bearer-Token which needs to be run under POST in Postman by clicking ‘send’

* Look at the body of the account\_details API to know the fields in it
* The result of sending of the above mentioned POST command can be seen at the bottom half of the page
* There will be object which has user\_phone, user\_email, user\_name
* GET account\_details API needs to be run with only Bearer-Token to get all the account information by clicking ‘send’, result is observed in bottom half of the page