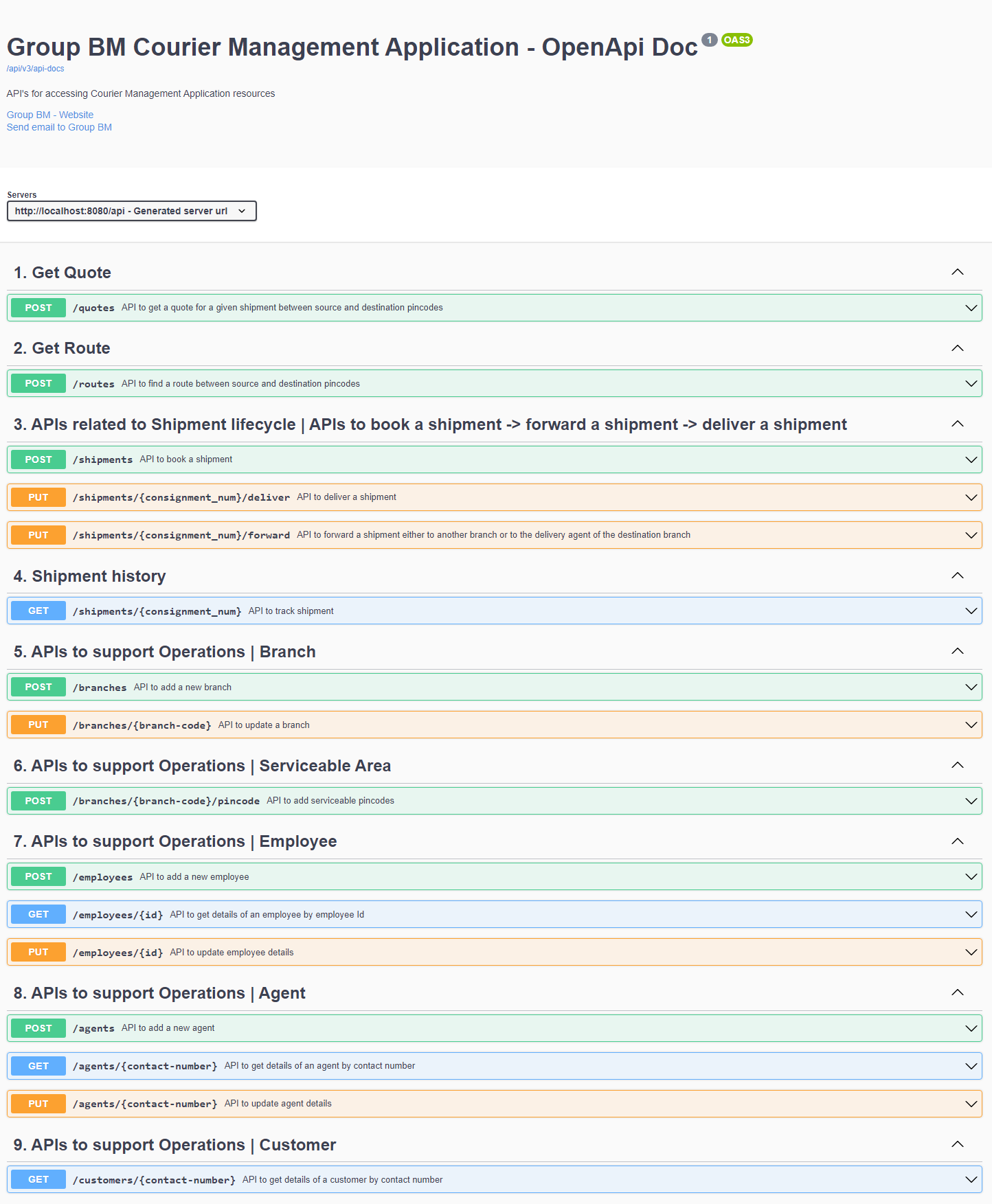
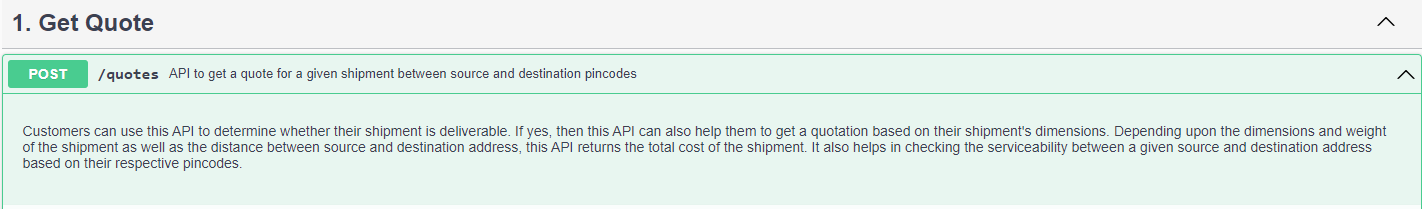
# Operations supported:

List of APIs which support functionalities as mentioned in the Problem Statement:

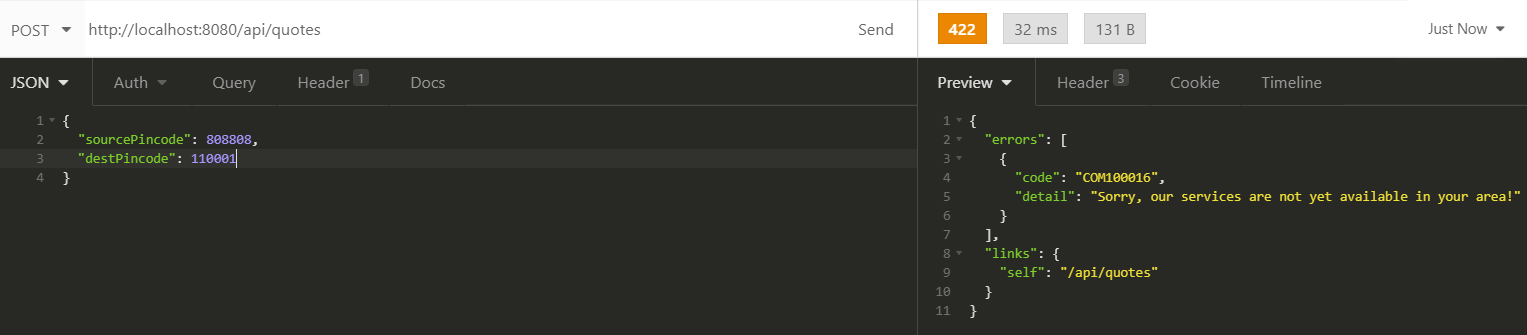


# USE CASE 1 # Courier Services not available at either source or destination pincode.

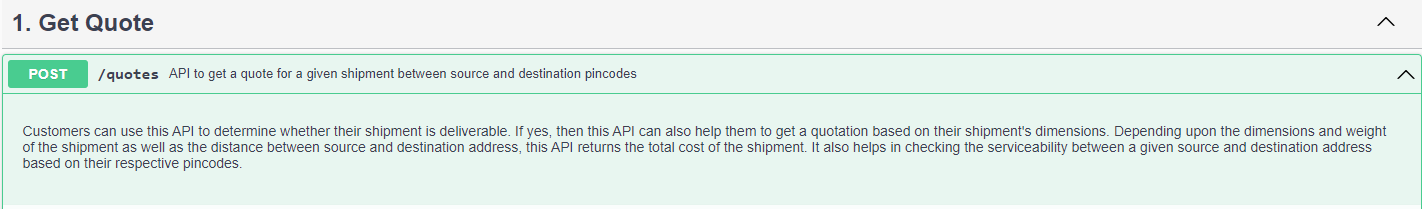
1. Customer residing at some area with pincode 808808, which is NOT serviceable by the application, uses get-quote API to check whether he can book a shipment and get it delivered to a serviceable area with pincode 110001.



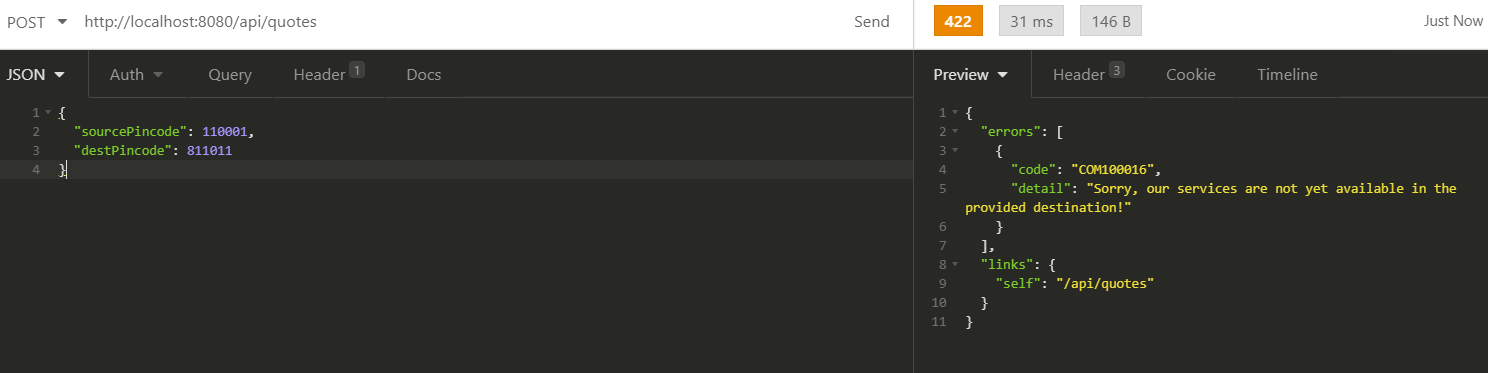
**Request & Response:**



1. Customer residing at some area with pincode 110001, which is serviceable by the application, uses get-quote API to check whether he can book a shipment and get it delivered to a location with pincode 811011 which is NOT yet served by the application.

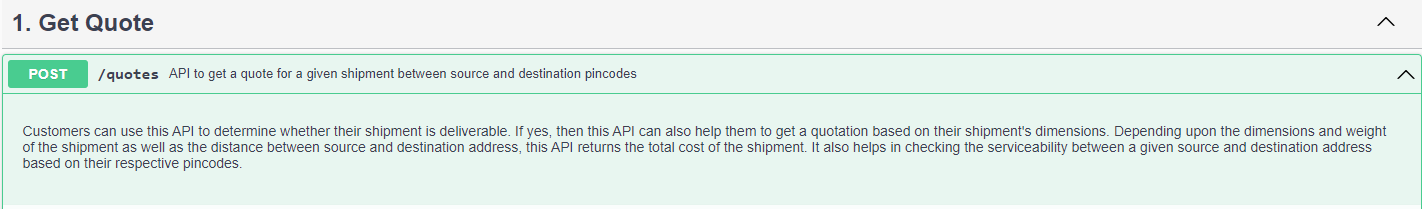


**Request And Response:**

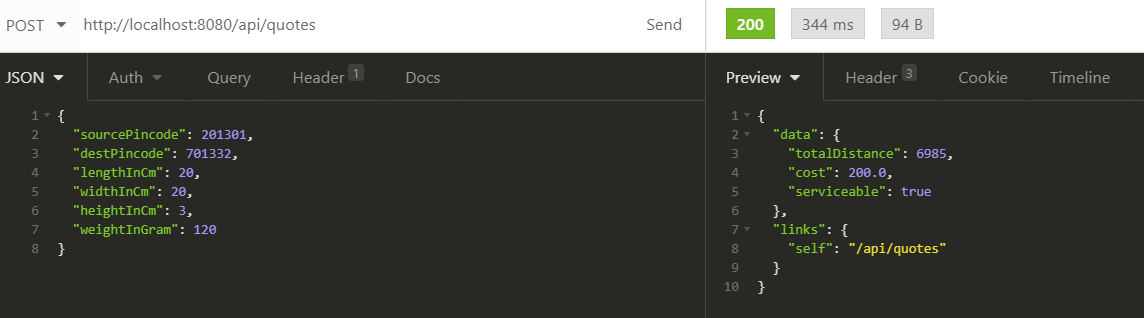


# USE CASE 2 # Courier Services are available at both source and destination address.

1. Customer residing at serviceable area with pincode 201301 is looking to get his consignment delivered at the location having pincode 701332 which is also served by the application. He uses get-quote API to get the quotations. His consignment dimensions are 20cm X 20cm X 3cm; weight = 120g

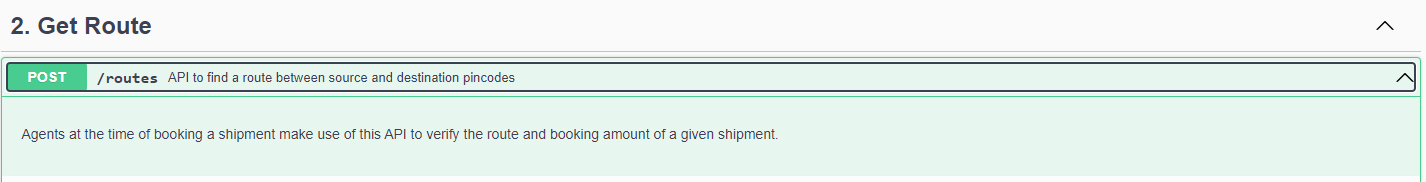


**Request & Response:**

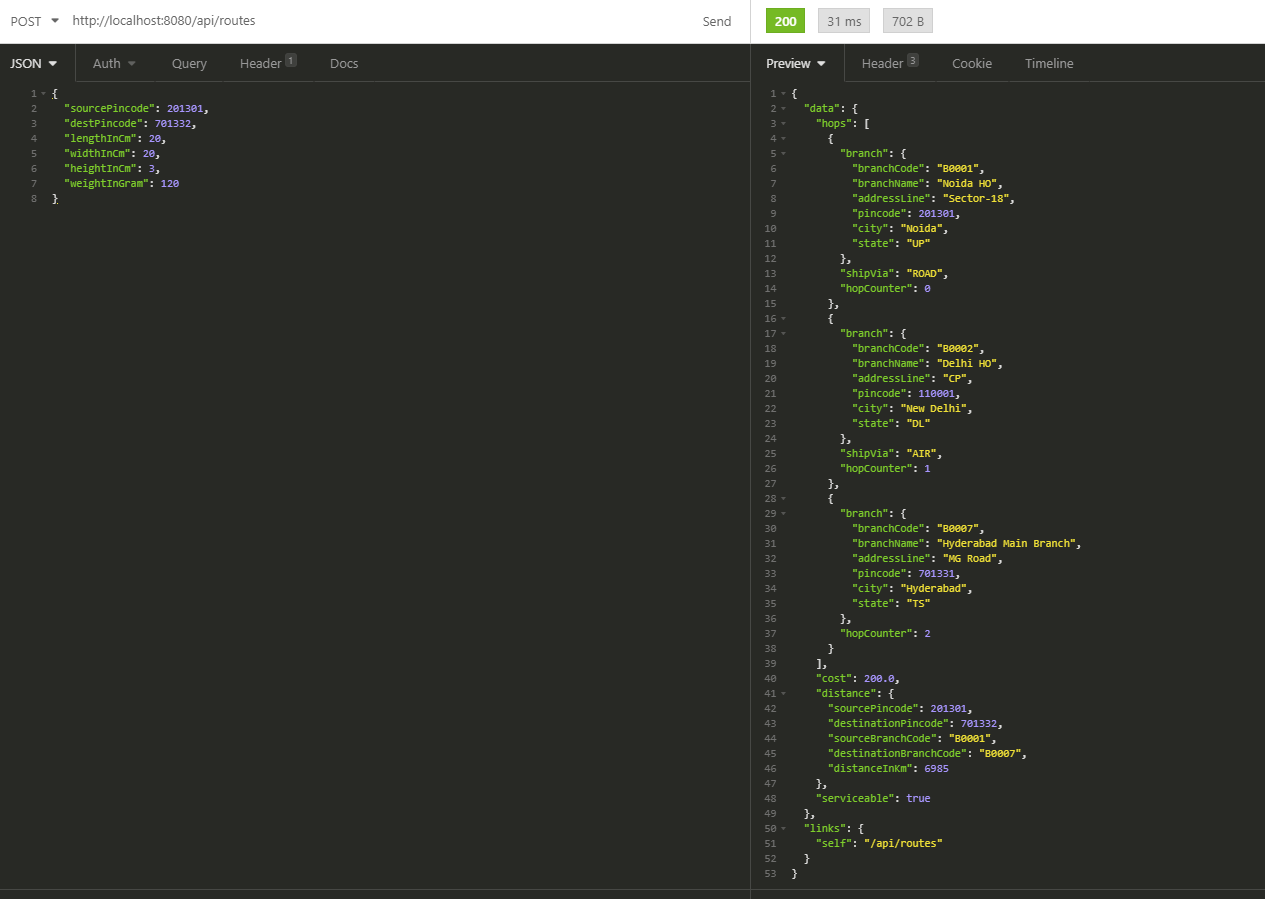


**Note: Application is using a random algorithm to evaluate distance between two pincodes. And based on this distance, it is referring to the rate\_card table and calculating the total cost.**

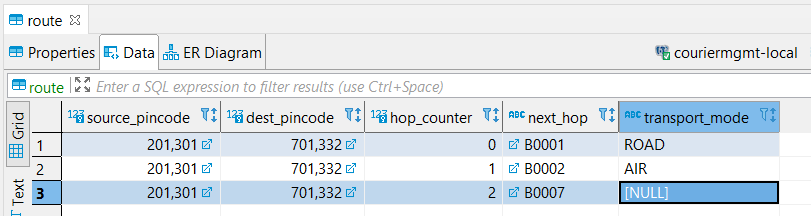
1. Customer then goes to the nearest booking agent, where agent verifies the booking amount and routing details using Get-Route API



**Request & Response:**



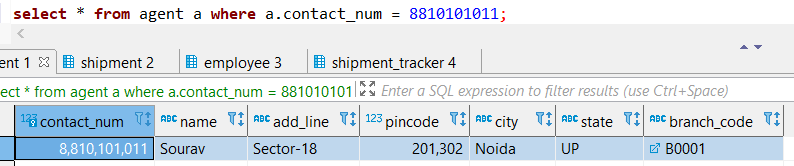
**Note: Application is using a random algorithm to generate route between two pincodes. This is how the data looks like in the route table depicting route between 201301 and 701332**

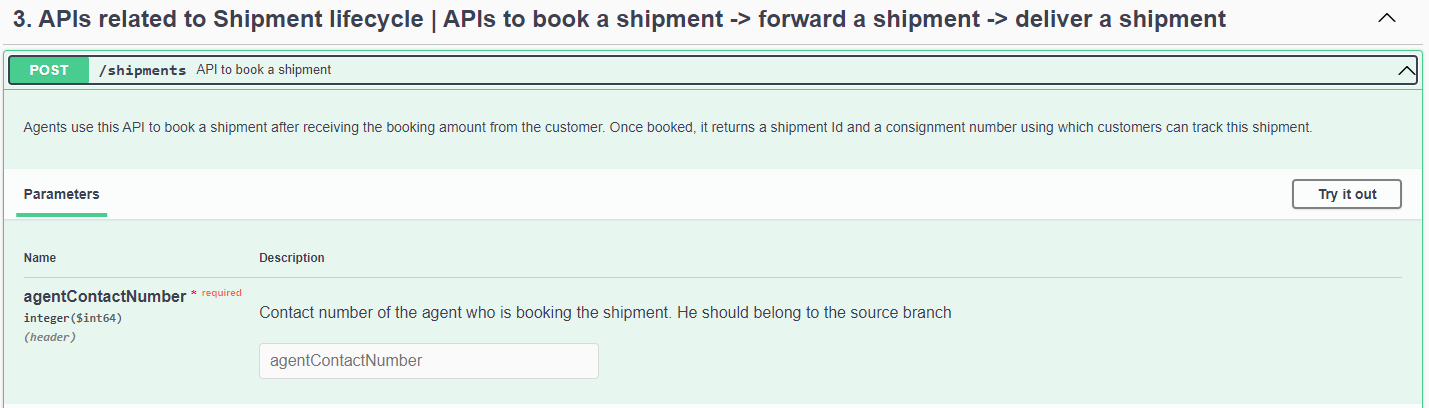


1. This agent (named Sourav having contact no. 8810101011) then collects the booking amount from the customer and books the shipment using **Book-Shipment API** with some optional remarks.

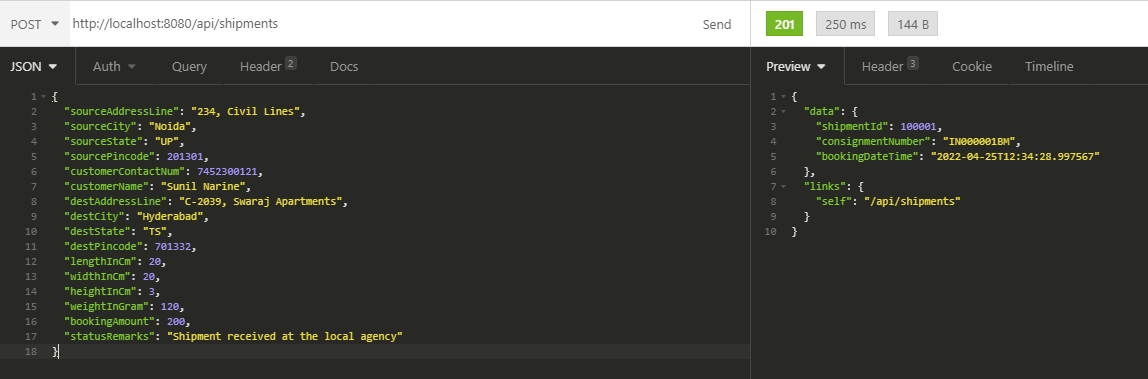
**Note: This API also takes care of adding the customer details into the system (customer table) if not already exists.**

**Verify that agent Sourav having contact no. 8810101011 belongs to B0001-Noida HO branch:**



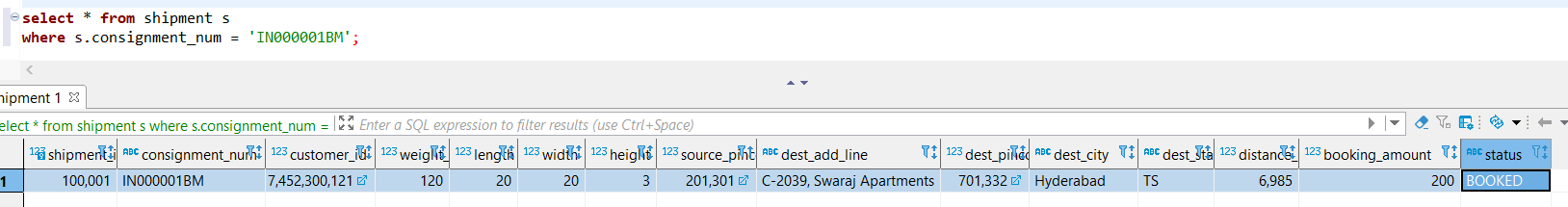


**Request & Response:**

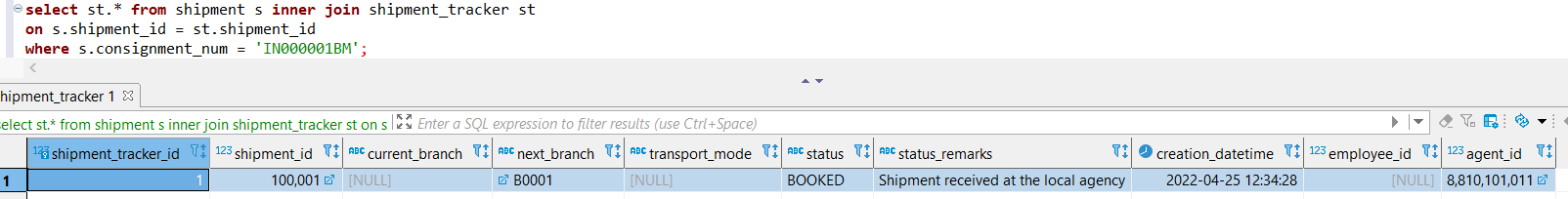


**Verify Table Snapshots:**

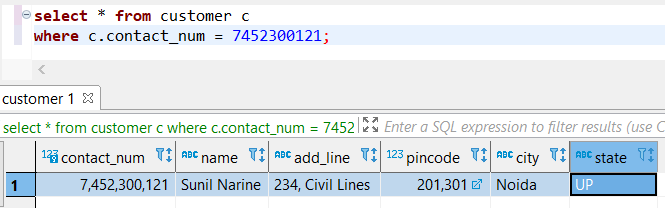
Shipment:



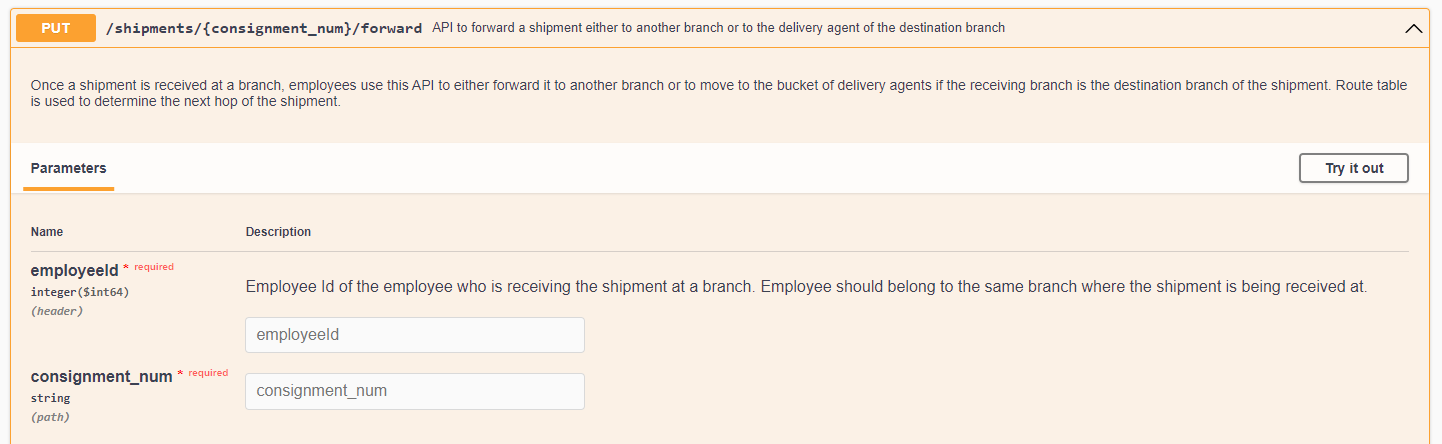
Shipment\_Tracker:

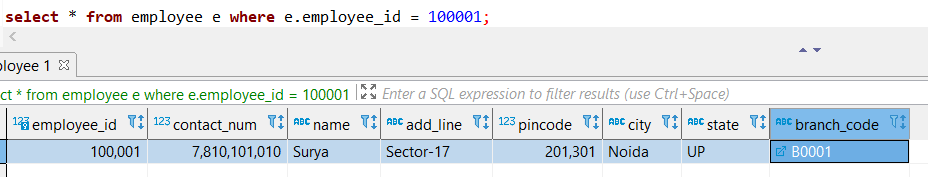


Customer:

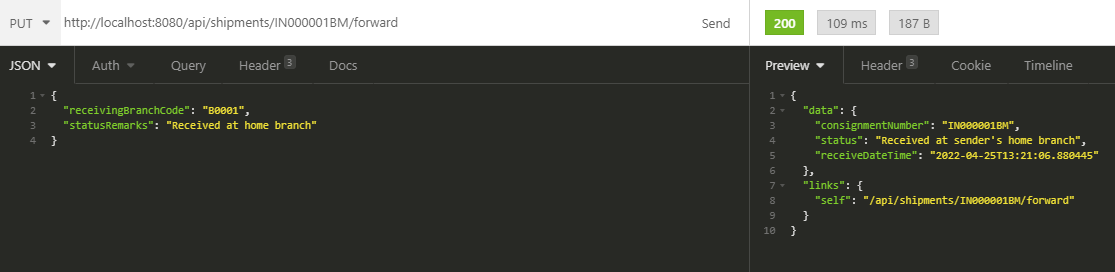


1. From the local agencies, shipments are moved to their respective local branches at regular interval. Thus, this shipment with consignment# **IN000001BM** is received at its local branch “**Noida HO**”. Employee at this branch having Employee ID# 100001, then uses **Forward-Shipment API** to receive it and updates its status to “Received at Source Branch”.



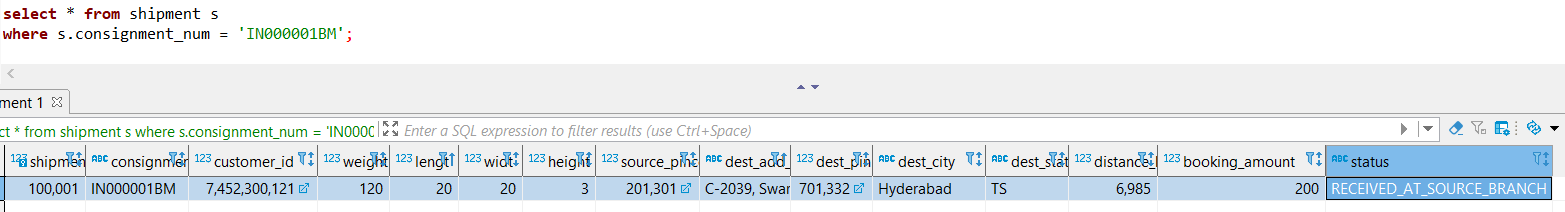
Verify that employee having employee ID 100001 belongs to B0001 (Noida HO):

**Request & Response:**

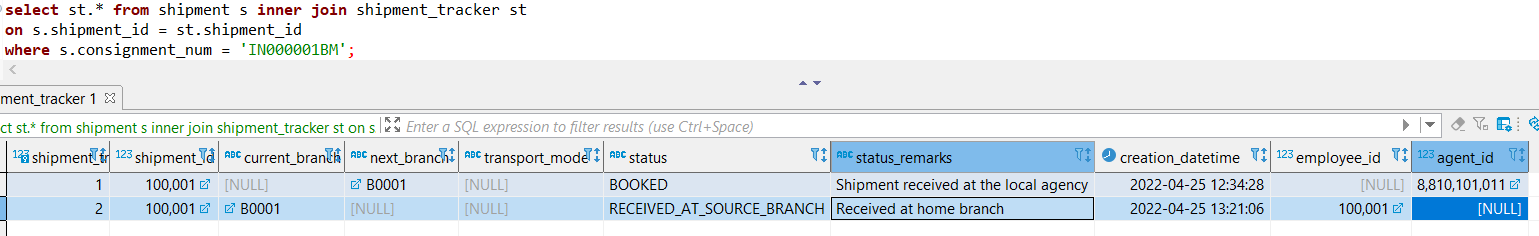


**Verify Table Snapshots:**

**Shipment:**

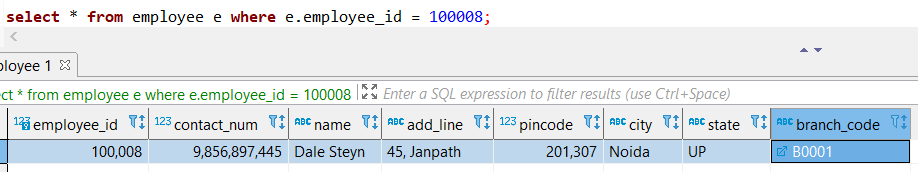


**Shipment Tracker:**

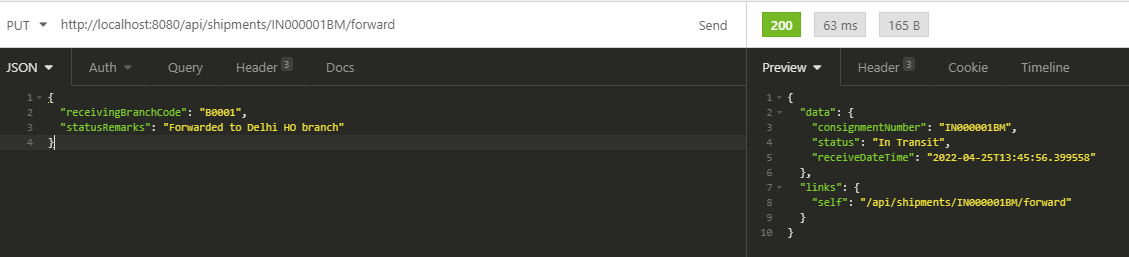


1. Shipments in status “Received at source branch” are then processed further. Another employee with employee ID #**100008** of the same branch again uses the **Forward-Shipment** API and forwards the shipment to the next branch (which is “Delhi HO”) based upon the pre-determined route as returned by Get-Route API in point#2. Status is updated to “In Transit”.

**Verify that employee having employee ID 100008 belongs to B0001 (Noida HO):**

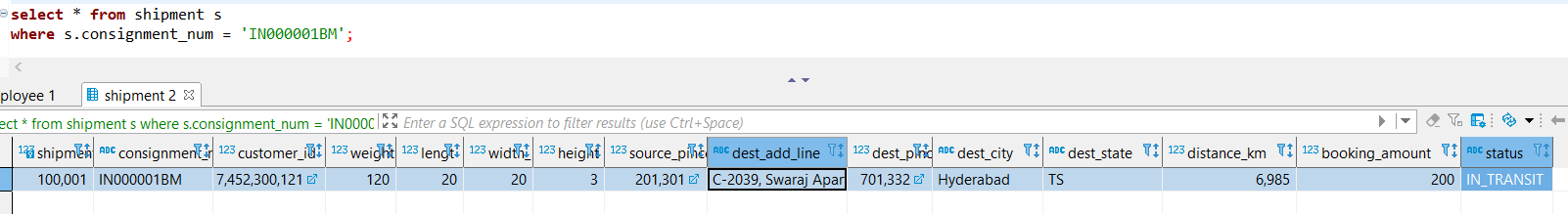


**Request & Response:**

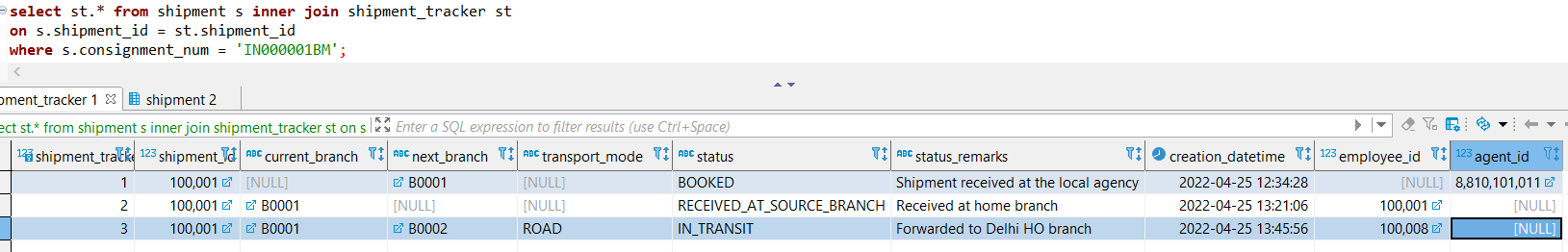


**Verify Table Snapshots:**

**Shipment:**

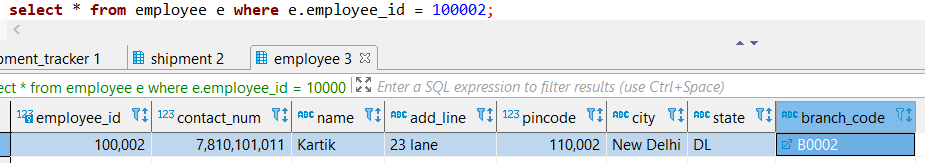


**Shipment\_Tracker:**

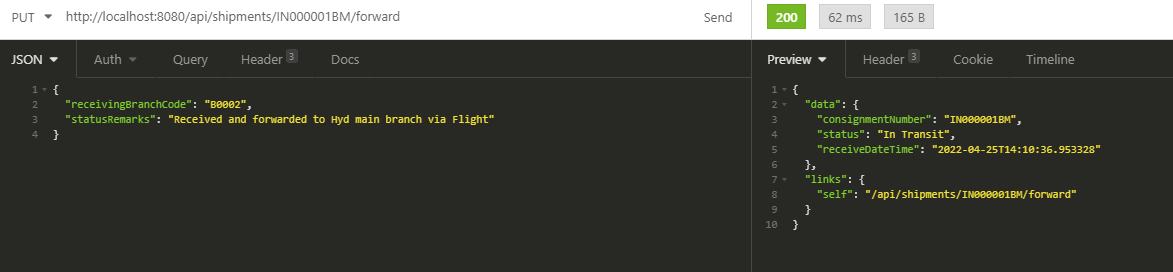


1. Shipment with consignment# **IN000001BM** is received at Delhi HO branch. Employee with id # 100002 processes it further using the same **Forward-Shipment** API.

Verify that employee having employee ID 100002 belongs to B0002 (Delhi HO):

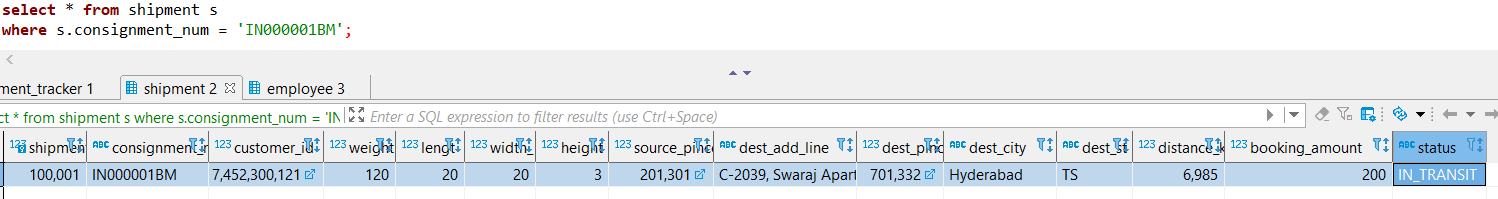


**Request & Response:**

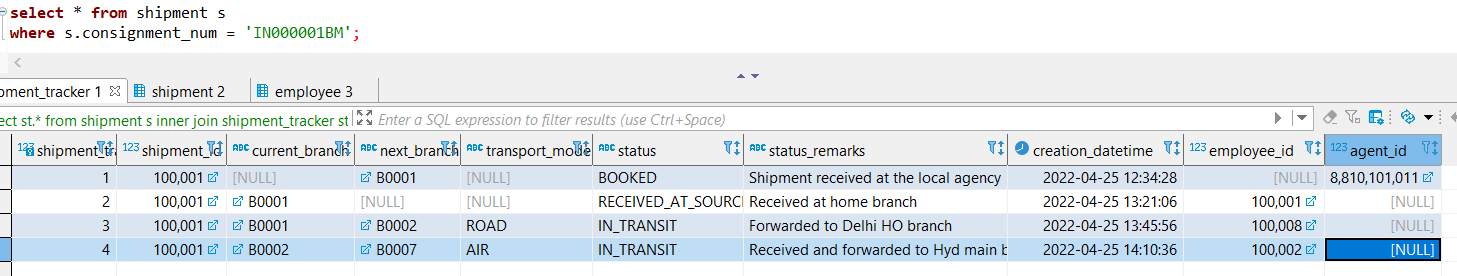


**Verify Table Snapshots:**

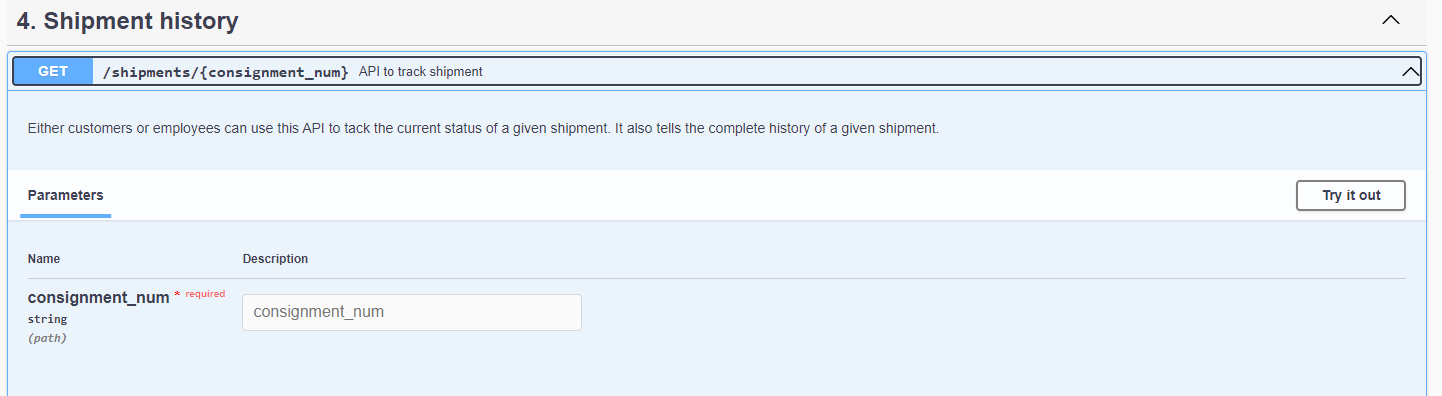
**Shipment:**



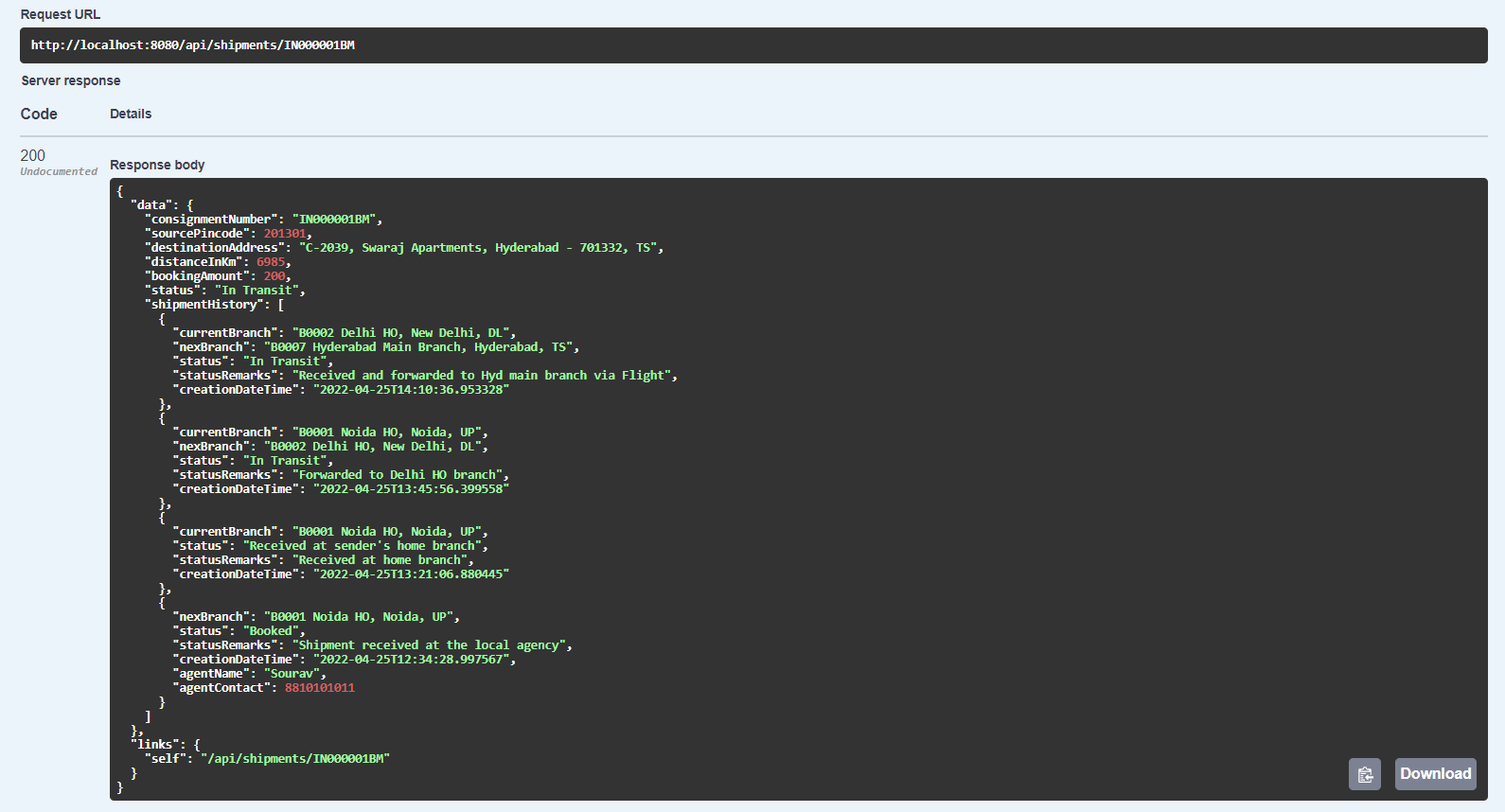
**Shipment\_Tracker:**



1. Customer wants to know the current status of his shipment **IN000001BM**. Application makes use of **Shipment-History API** which retrieves data from the shipment\_tracker table and present the response to the customer.



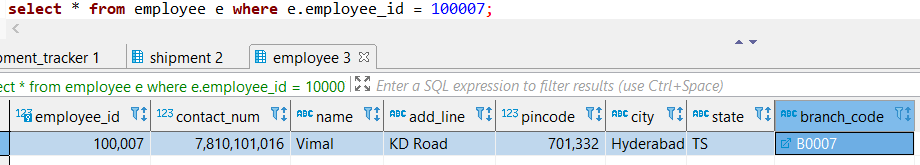
**Request & Response:**



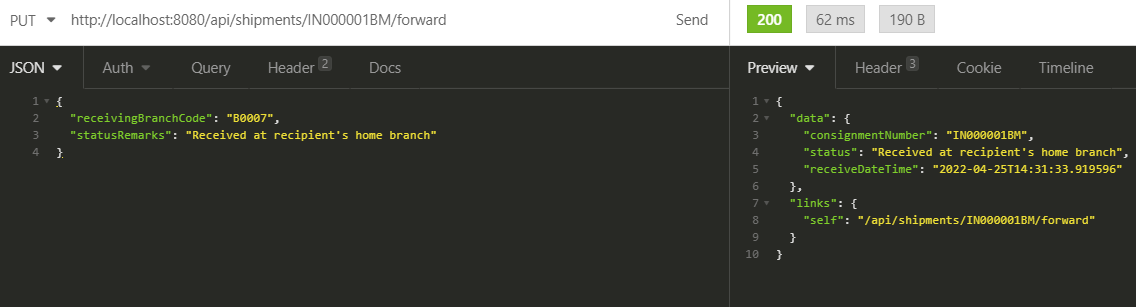
**Note: The response is sorted by record creation date (desc) i.e., first record in the “shipmentHistory” list is the latest or current status of the shipment IN000001BM.**

1. Shipment with consignment# **IN000001BM** is received at Hyderabad Main branch. Employee with id # 100007 processes it further using the same **Forward-Shipment** API. This API evaluates that the current receiving branch (B0007-Hyderabad Main branch) is the recipient’s home branch, thus the status of the shipment is updated to “Received at Destination Branch”. By doing this, employee 100007 moves this shipment to the delivery agents’ buckets.

Verify that employee having employee ID 100007 belongs to B0007 (Hyderabad Main branch):

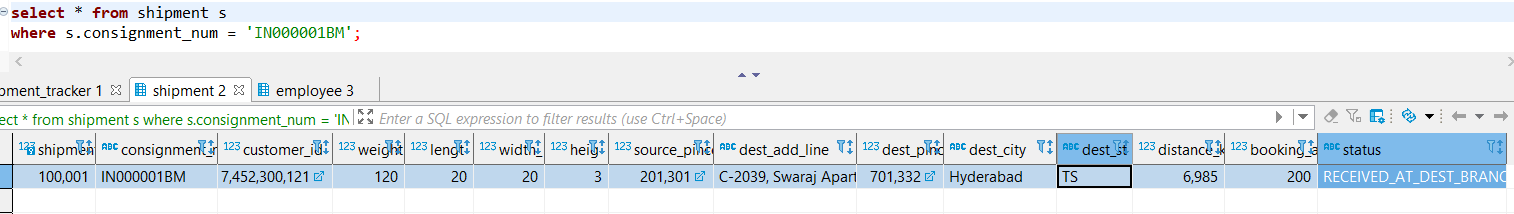


**Request & Response:**

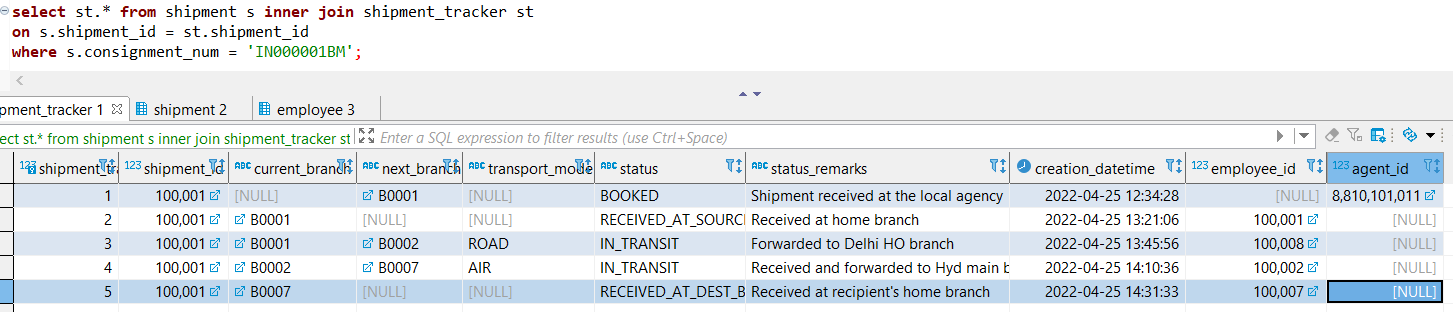


**Verify Table Snapshots:**

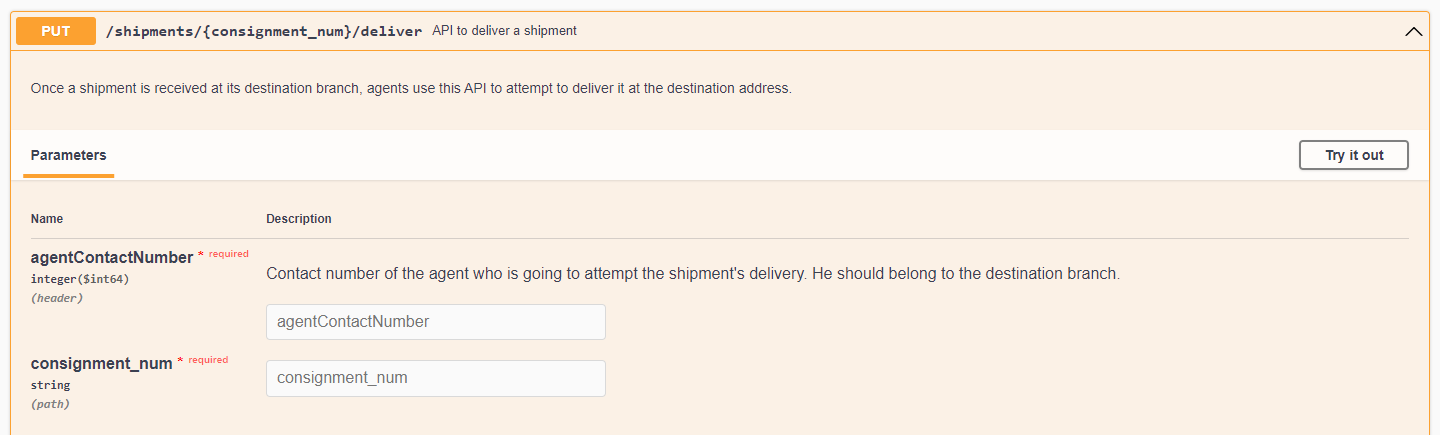
**Shipment:**



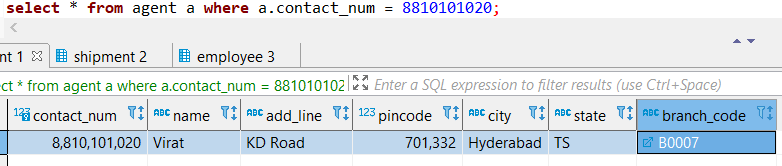
**Shipment\_Tracker:**



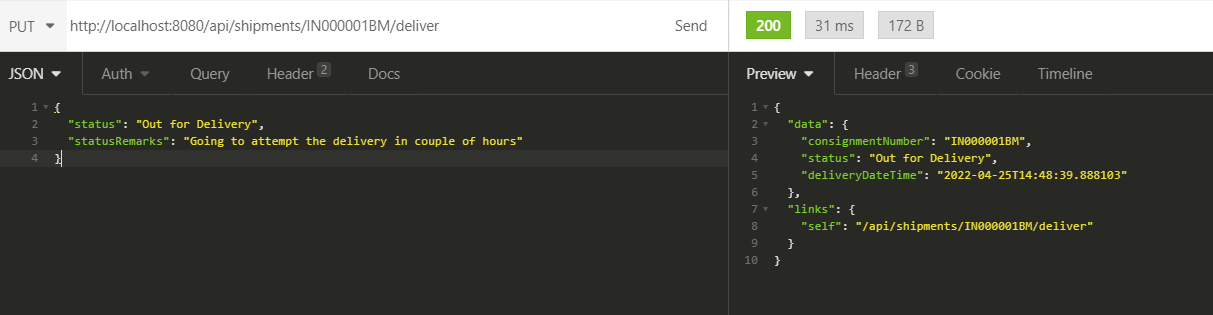
1. Agents, at their respective branches, timely check their buckets for any new or undelivered shipments. Agent Virat from Hyderabad Branch, having contact number 8810101020 picks up the shipment with consignment# **IN000001BM** and updates its status to “Out for Delivery”.



Verify that agent Virat having contact number 8810101020 belongs to B0007 (Hyderabad Main branch):

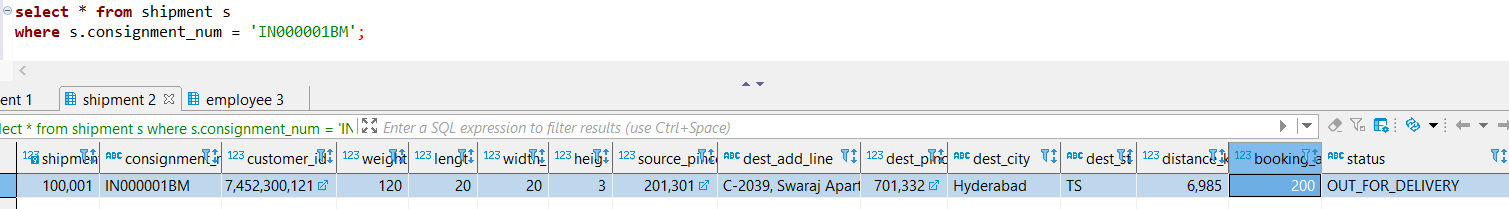


**Request & Response:**

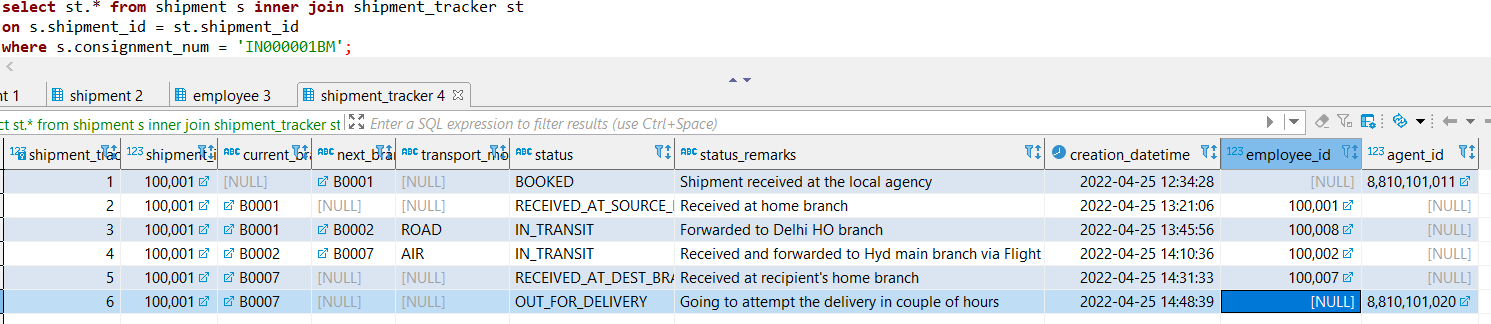


**Verify Table Snapshots**

**Shipment:**

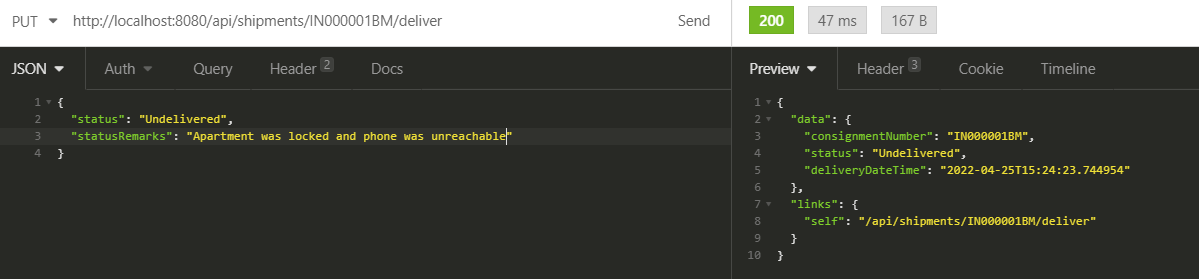


**Shipment\_Tracker:**



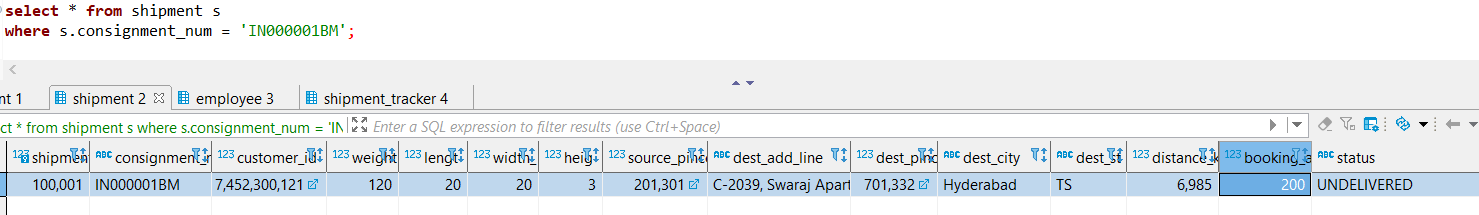
1. Agent Virat attempts the delivery of the Shipment having consignment# **IN000001BM** but since no one was available at the destination address, it gets undelivered. He marks the status as Undelivered using the same **Deliver-Shipment API**. He or some other agent can grab it later to attempt another delivery.

**Request & Response:**

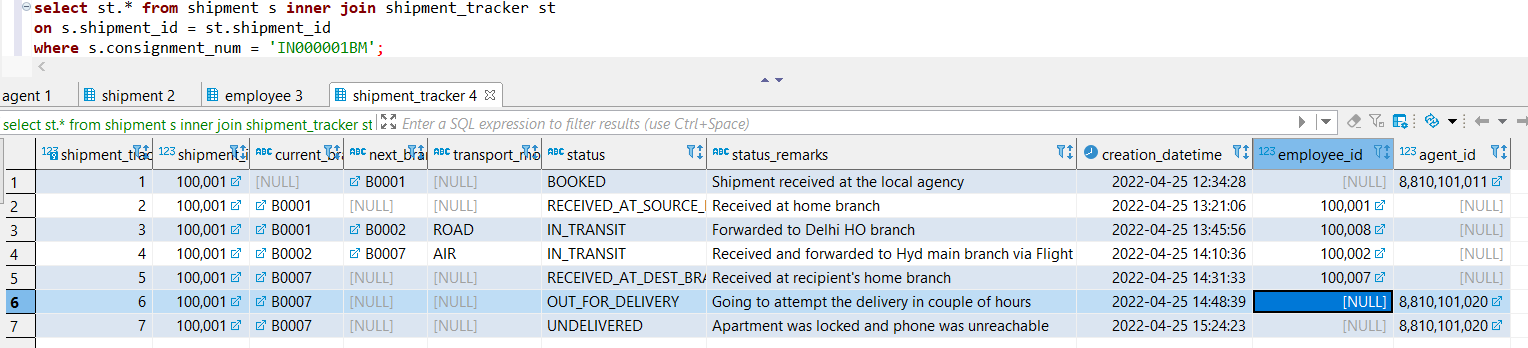


**Verify Table Snapshots:**

**Shipment:**

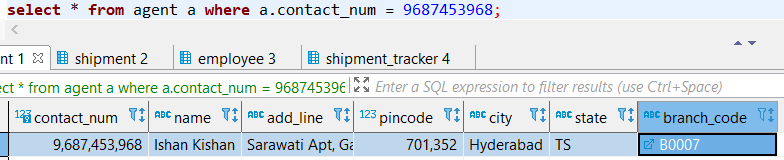


**Shipment\_Tracker:**

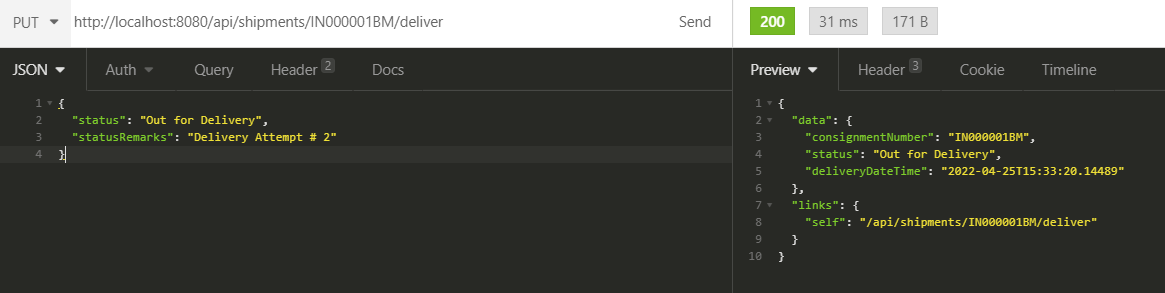


1. Another agent named Ishan from the same Hyderabad Branch, having contact number 9687453968 attempts another delivery of the undelivered Shipment (consignment# **IN000001BM)** using the same **Deliver-Shipment API**.

**Verify that agent Ishan having contact number 9687453968 belongs to B0007 – Hyderabad Main Branch:**

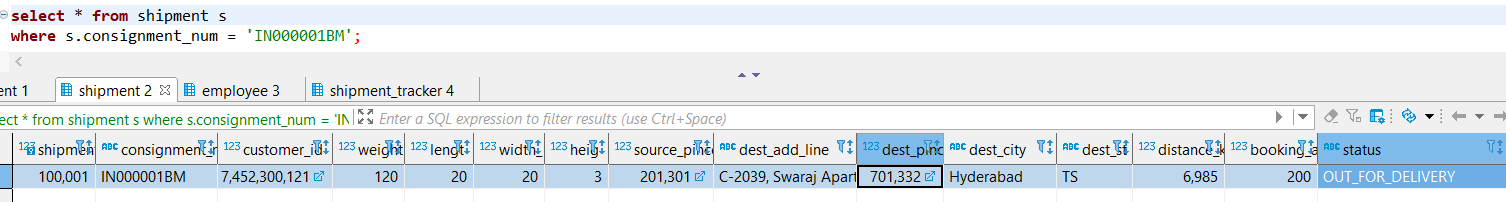


**Request & Response:**

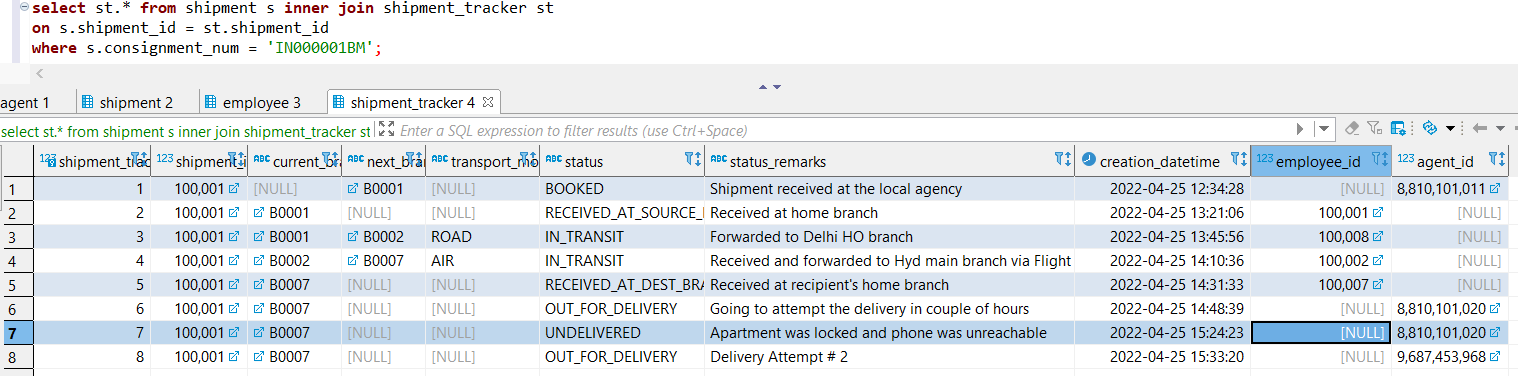


**Verify Table Snapshots:**

**Shipment:**

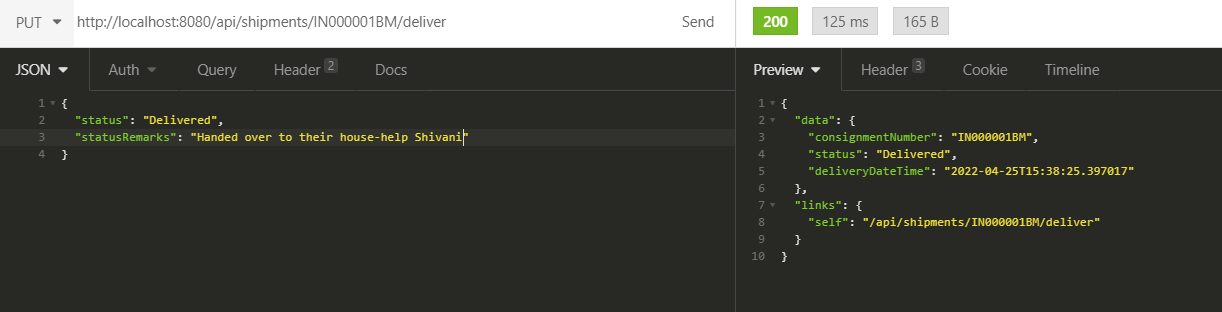


**Shipment\_Tracker:**



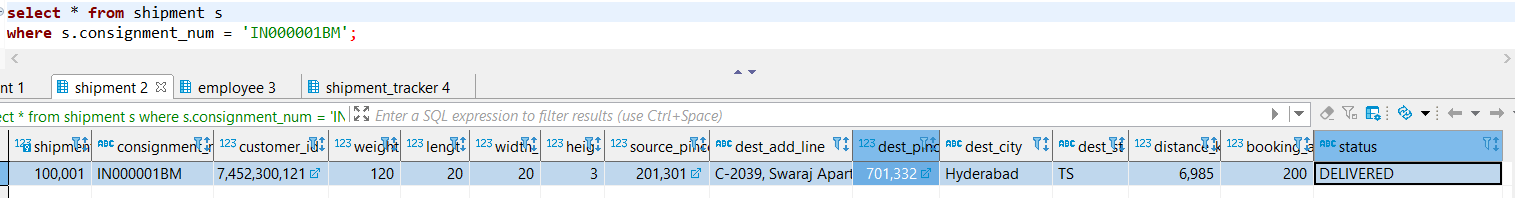
1. Agent Ishan attempts another delivery of the Shipment having consignment# **IN000001BM** and this time it gets delivered. He marks the status as Delivered using the same **Deliver-Shipment API**.

**Request & Response:**

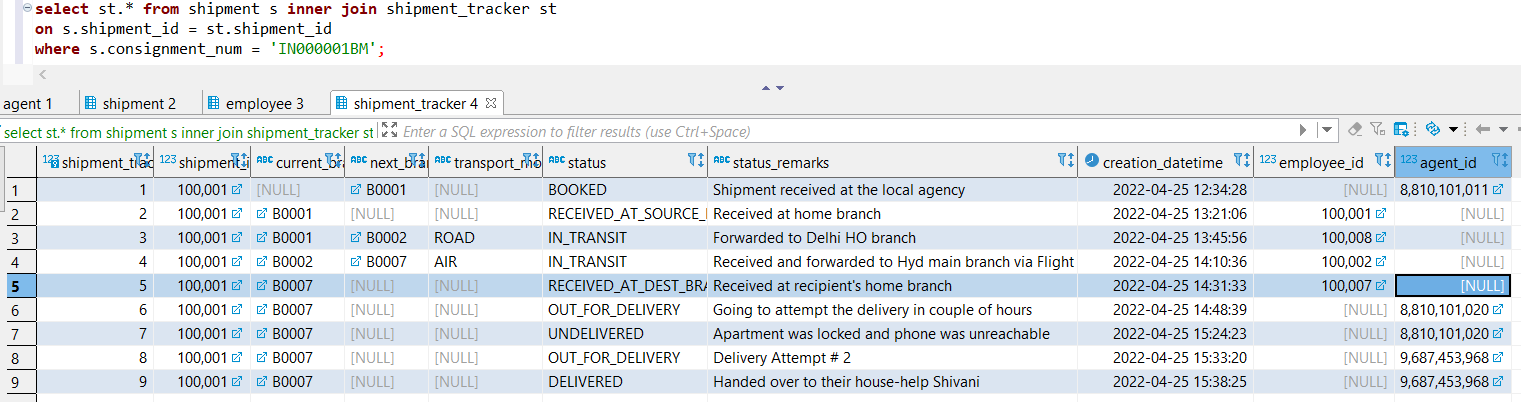


**Verify Table Snapshots:**

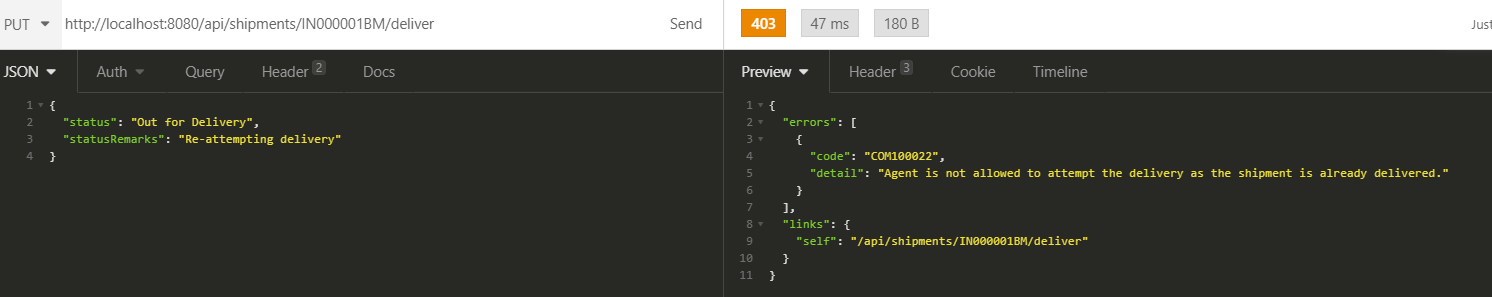
**Shipment:**



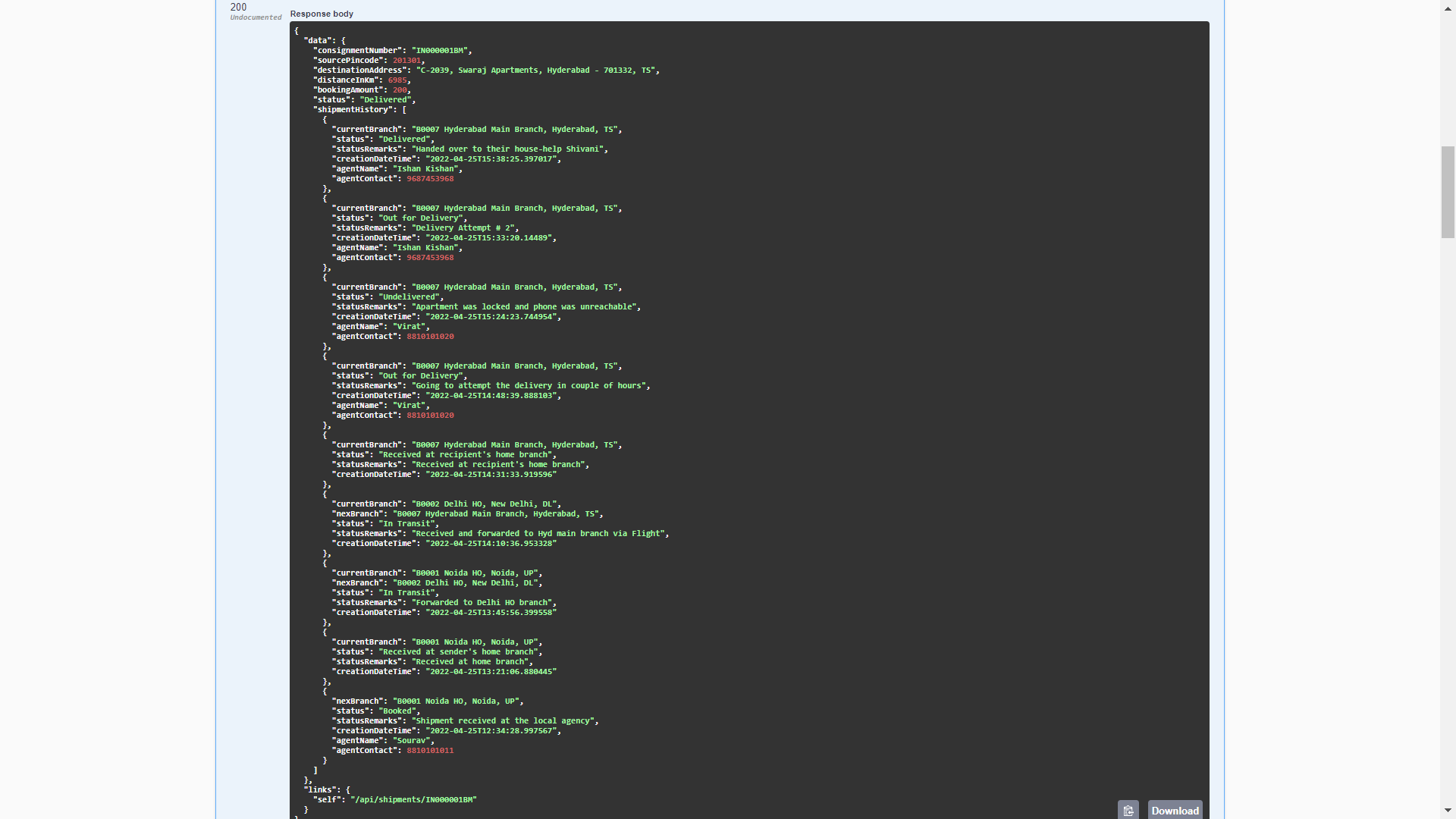
**Shipment\_Tracker:**



1. Various validations are also in place. As an example:
   * Once a shipment is Delivered, its status can’t be changed. Below is the sample request-response of Deliver-Shipment API:



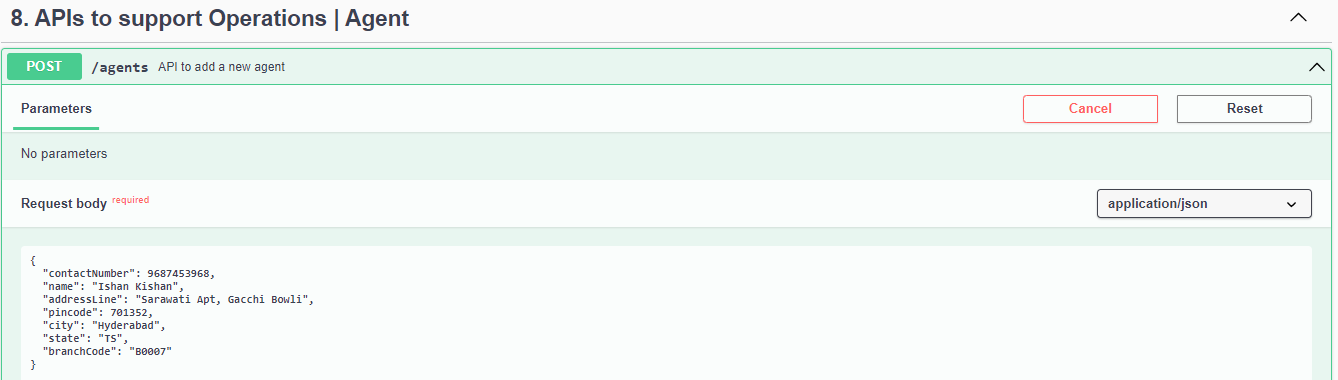
1. Response from the **Shipment Tracker API** depicting the complete journey of Shipment having consignment# **IN000001BM** right from being Booked to get Delivered.



# USE CASE 3 # Usage of APIs for managing various Application entities like Agents, Employees, Customers, Branches, Service Pincodes

1. Add Agent using **Add-Agent API**

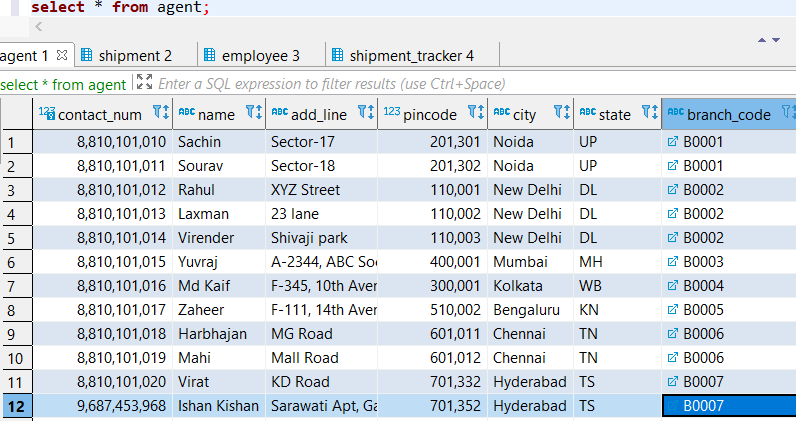
Request:



**Response:**

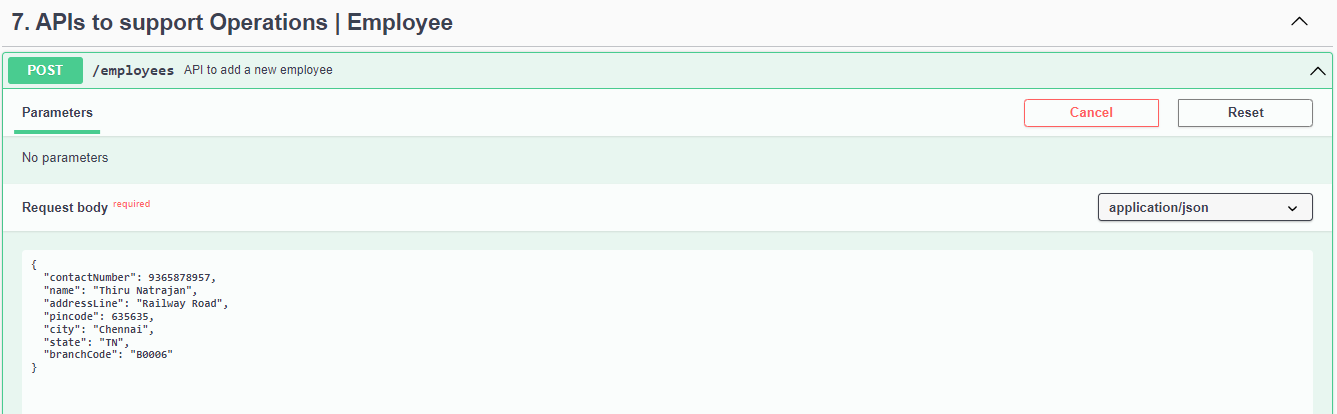


**Verify Table Agent: Record is inserted with PK 9687453968**

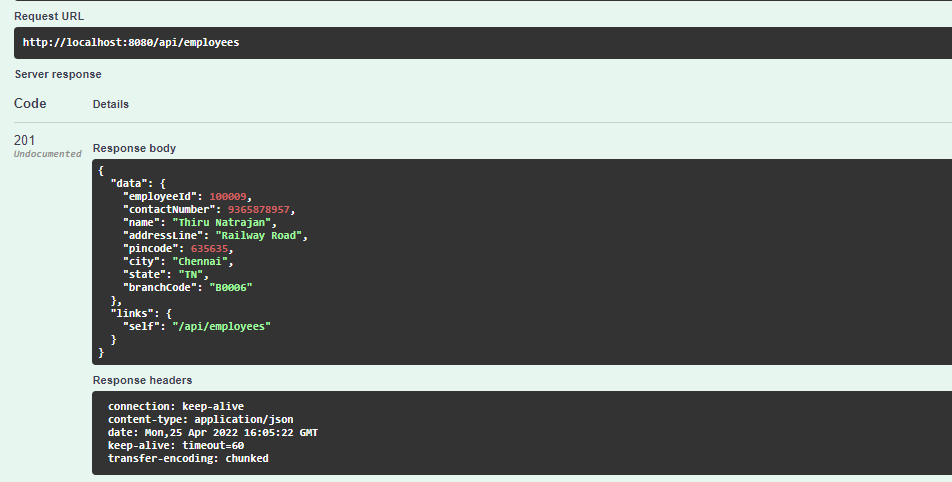


1. Add Employee using **Add-Employee API:**

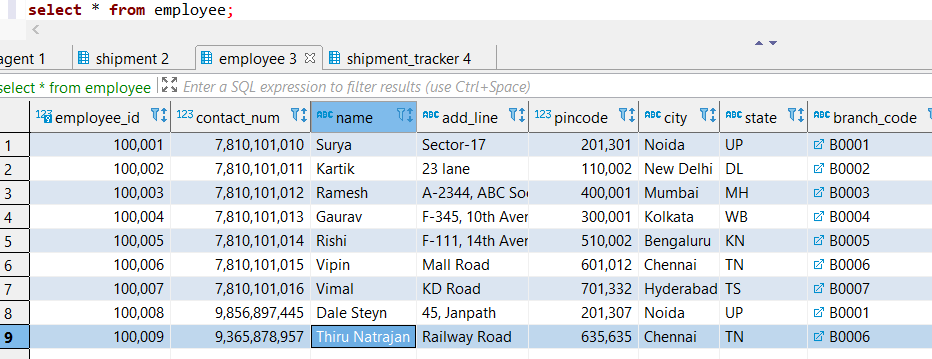
Request:



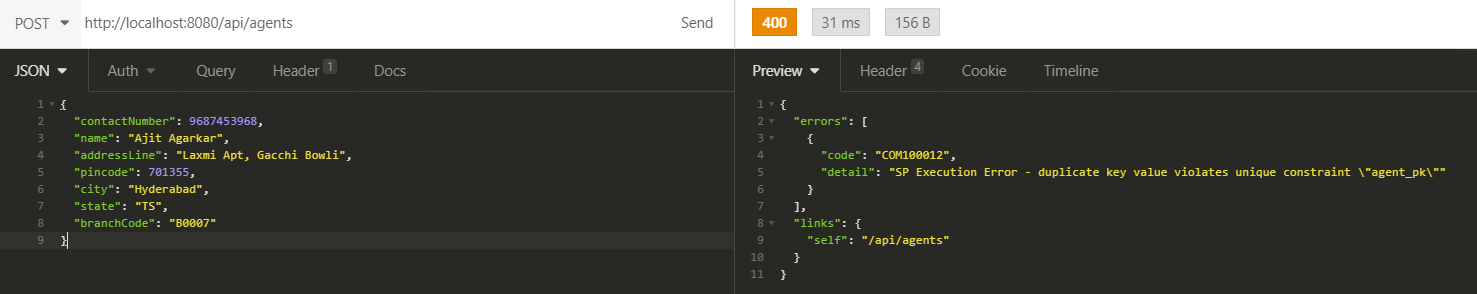
Response:



**Verify Employee Table (Record is inserted with PK 100009):**

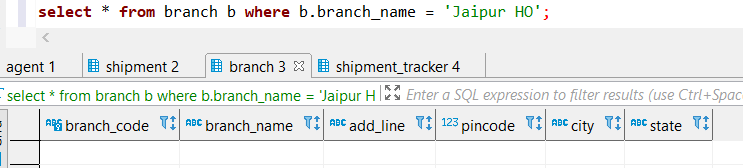


1. Adding Agent with same PK **9687453968, returns PK violation error**

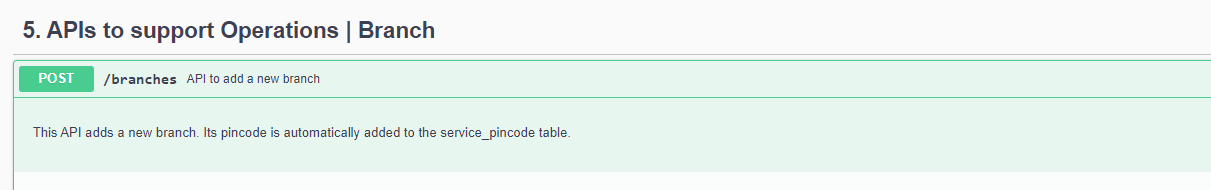


1. Add Branch using **Add-Branch** API

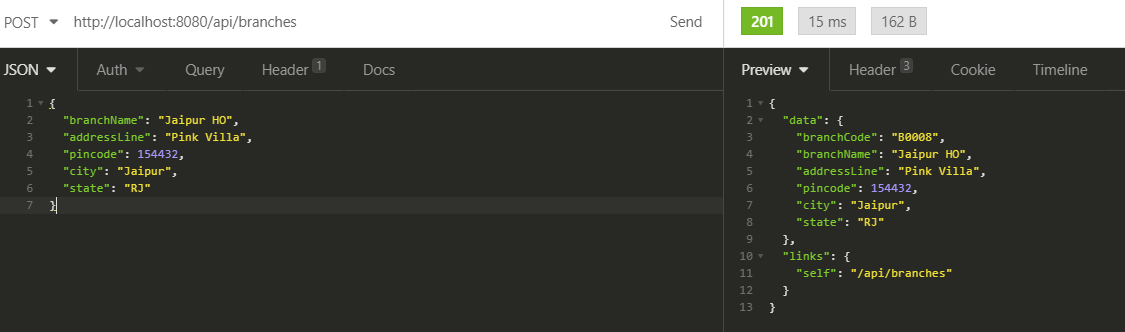
Jaipur branch didn’t exist before.



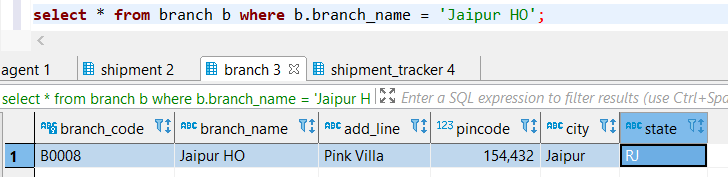
Adding it using Add-Branch API

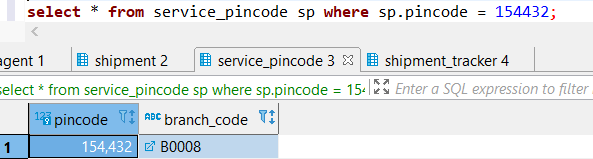


**Request & Response:**



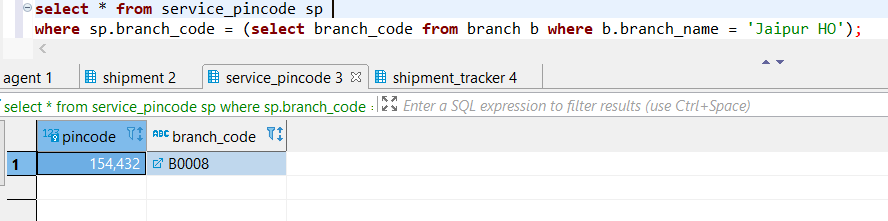
**Verify Branch table and Service\_Pincode table:**



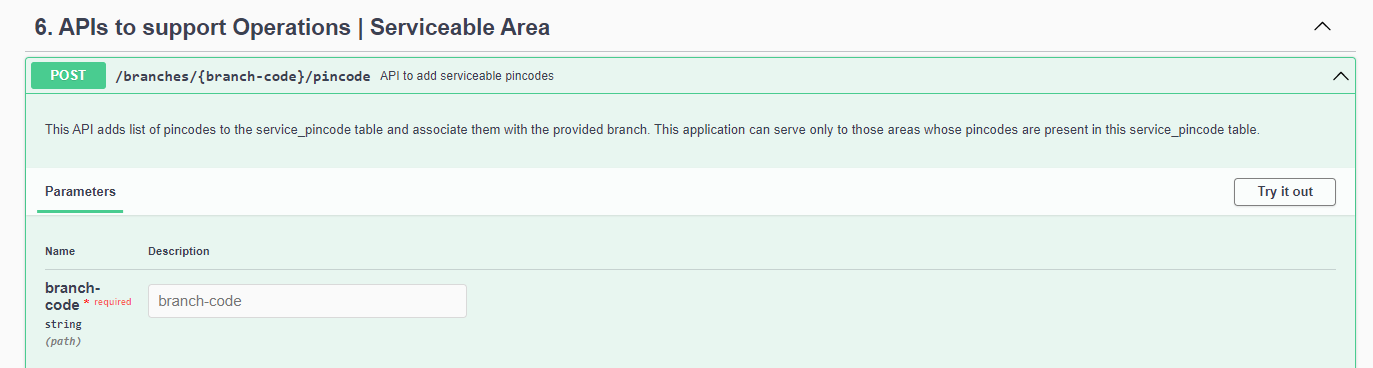


1. **Add list of serviceable pincodes by a branch using Add-Pincode API**

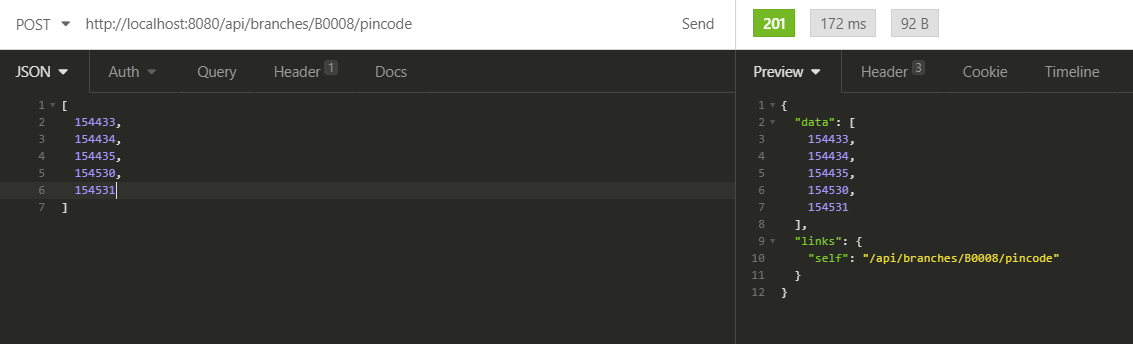
List of pincodes being served by B0008 - Jaipur HO branch before hitting this API



Five more pincodes will be added using Add-Pincode API which can be served by Jaipur HO branch



**Request & Response:**



**List of pincodes being served by B0008 - Jaipur HO branch after hitting this API:**

