Enabling “Ambulance Booking” workflow via DHP

# Introduction

Ambulance booking is a fundamental use case in healthcare required for the transportation of patients and their caregivers from one location to another. Ambulances are requested during an emergency and non-emergency situations. In most cases, the destination of an ambulance journey is a hospital or a healthcare service facility.

The commonly observed workflows of ambulance services have been broken down into 4 stages defined by DHP - Discovery, Booking, Fulfillment, and Post-fulfillment.

## **Ambulance service Discovery Flow**

### **Search API - call between the BAP & DHP gateway for search broadcast**

This is the first step in the ambulance service discovery flow where a search request is shared by the BAP to the gateway. The gateway then broadcasts the search to all the relevant BPPs registered on the network.

A user can search for an ambulance service using one or more search parameters listed below -

1. **Pick up location:** My current location or other
2. **Drop Location :** hospital/clinic/etc.
3. **Ride schedule:** Schedule now/schedule later
4. **Start Date :**
5. **Start time:** current Time or schedule later
6. **Service Category:** free, paid
7. **Fulfillment Type :** emergency/non-emergency
8. **Patient Type:** Adult, Child, Senior Citizen, Critically Ill
9. **Ambulance Type**: BLS, ALS, Pediatric, ICU, Hearse

#### *Usecase 1: Search an on-demand ambulance by Start & end locations*

**Steps :**

* User opens/visits an EUA (consumer side app) for ambulance search.
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* The user selects the ride schedule as “ride now”.
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service (nic code to be confirmed)

country: "IND",

city: <use std code for the city>

action: "search",

core\_version: <latest version of DHP>

bap\_id: "https://example-bap.com/",

message ID : UUID

timestamp : <current timestamp>

**Message Parameters**

Pick up location: My current location or the location provided by the user

Drop Location : hospital/clinic/etc.

Ride schedule : Schedule now

Start time : current

Start date : current

**JSON Example : [To do]**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T10:00:40.065Z ",

},

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

}

}

}

}

}

}

#### *Usecase 2: Search by Start & end locations and ride schedule= schedule for later*

**Steps :**

* User opens/visits an EUA (consumer side app) for ambulance search.
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* The user selects the ride schedule as “schedule for later”.
* The user enters the “start date” from the calendar/manually
* The user enters the “start time” from the time slots
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "search",

core\_version:

bap\_id: "https://example-bap.com/",

message ID :

timestamp :

**Message Parameters**

Pick up location : My current location or other

Drop Location : hospital/clinic/etc.

Ride schedule : schedule for later

Start date :

Start time :

**JSON Example :**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-29T18:00:00.000Z ",

},

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

}

}

}

}

}

}

#### *Usecase 3 : Search by Start & end locations and ride schedule= schedule now and patient type*

**Steps :**

* User opens/visits a DHP network integrated app/web portal for ambulance search.
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* The user selects the ride schedule as “ride now”.
* User selects patient type: Adult, Child, Senior Citizen, Critically Ill
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain: Category code for ambulance service

country: "IND",

City:

action: "search",

core\_version:

bap\_id: "https://example-bap.com/",

message ID :

timestamp :

**Message Parameters**

Pick up location: My current location or other

Drop Location : hospital/clinic/etc.

Ride schedule : Schedule now

Start time : Current

Start date: Current

Patient type : Adult/Child/Senior Citizen/Critically Ill

**JSON Example :**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T10:00:40.065Z ",

},

"customer" : {

"person" : {

"tags" : {

"patient-type": "Adult"

}

}

}

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

}

}

}

}

}

}

#### *Usecase 4 : Search by Start & end locations and ride schedule= now and fulfillement type = non- emergency*

**Steps :**

* User opens/visits a DHP network integrated app/web portal for ambulance search.
* The user selects the fulfillment type as “non-emergency”
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* The user selects the ride schedule as “now”.
* BAP picks up the current date & time
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain: Category code for ambulance service

country: "IND",

City:

action: "search",

core\_version:

bap\_id: "https://example-bap.com/",

message ID :

timestamp :

**Message Parameters**

Pick up location: My current location or other

Drop Location : hospital/clinic/etc.

Ride schedule : now

Start date : current

Start time : current

Start time: current time

Fulfillment type: non-emergency

**JSON Example :**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"type" : "non-emergency",

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T10:00:40.065Z ",

}

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

}

}

}

}

}

}

#### *Usecase 5 : Search by Start & end locations and ride schedule= now and fulfillement type = emergency*

**Steps :**

* User opens/visits a DHP network integrated app/web portal for ambulance search.
* The user selects the fulfillment type as “emergency”
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* The user selects the ride schedule as “now”.
* BAP picks up the current date & time
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain: Category code for ambulance service

country: "IND",

City:

action: "search",

core\_version:

bap\_id: "https://example-bap.com/",

message ID :

timestamp :

**Message Parameters**

Pick up location: My current location or other

Drop Location : hospital/clinic/etc.

Ride schedule: now

Start date: Current

Start time : current

Start time : current time

Fulfillment type : emergency

**JSON Example :**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"type" : "emergency",

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T10:00:40.065Z ",

}

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

}

}

}

}

}

}

#### Usecase 6 : Search by Start & end locations and fulfillment type = rental

**Steps :**

* User opens/visits a DHP network integrated app/web portal for ambulance search.
* User selects the fulfillment type as “rental”
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* User selects the start date
* User selects start time
* User selects end date
* User selects end time
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain: Category code for ambulance service

country: "IND",

city:

action: "search",

core\_version:

bap\_id: "https://example-bap.com/",

message ID :

timestamp :

**Message Parameters**

Pick up location: My current location or other

Drop Location : hospital/clinic/etc.

Fulfillment type: rental

Start date :

End date :

Start time :

End time :

**JSON Example :**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"type" : "rental",

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T10:00:40.065Z",

}

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T15:00:40.065Z",

}

}

}

}

}

}

#### *Usecase 5: Search by Start & end locations and ride schedule= now, fulfillment type = emergency, and patient type*

**Steps :**

* User opens/visits a DHP network integrated app/web portal for ambulance search.
* The user selects the fulfillment type as “emergency”
* The user enters “pick up location” by searching a location by name or map.
* The user now enters “drop location” by searching a location by name or map.
* The user selects the ride schedule as “now”.
* BAP picks up the current date & time
* Submits the search intent by hitting a search/submit action button.
* Search intent is passed on to the DHP gateway via a “search API” call.
* On receiving the search API call from the BAP, the gateway sends back an ACK message and broadcasts the search to registered service providers in the network.

**Parameters passed :**

**Context Parameters**

Domain: Category code for ambulance service

country: "IND",

city:

action: "search",

core\_version:

bap\_id: "https://example-bap.com/",

message ID :

timestamp :

**Message Parameters**

Pick up location: My current location or other

Drop Location : hospital/clinic/etc.

Ride schedule: now

Start date : current

Start time : current

Start time : current time

Fulfillment type : emergency

Patient type : Adult/Child/Senior Citizen/Critically Ill

**JSON Example :**

{

"context": {

"domain": "nic2006:12345",

"country": "IND",

"city": "std:080",

"action": "search",

"core\_version": "0.9.2",

"bap\_id": "https://example-bap.com/",

"bap\_uri": "https://example-bap.com/beckn/",

"transaction\_id": "1234567890",

"message\_id": "123e4567-e89b-12d3-a456-426614174000",

"timestamp": "2022-08-24T10:00:40.065Z"

},

"message": {

["intent"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/intent) : {

["fulfillment"](https://developers.becknprotocol.io/docs/logistics-specification/schema-reference/fulfillment): {

"type" : "emergency",

"start" : {

"location" : {

"gps" : "12.4535445,77.9283792"

},

"time" : {

"timestamp" : "2022-08-24T10:00:40.065Z",

},

"customer" : {

"person" : {

"tags" : {

"patient-type": "Senior Citizen"

}

}

}

},

"end" : {

"location" : {

"gps" : "12.4535445,77.9283792"

}

}

}

}

}

}

## **on\_search response - response to the search intent broadcasted by the gateway**

A service provider will respond back with a catalog of ambulance services matching the search request

1. Items: one-way tri/round trip/rental/advance booking
2. Ambulance variant: Basic Ambulance/Advance Ambulance/Mortuary Ambulance/Neonatal Ambulance/Patient Transport Vehicle/Air Ambulance
3. fulfillment type:emergency/non-emergency
4. price: for each item
5. add-ons: equipment, medical attendants & types, paramedics, etc

**Steps :**

* Gateway broadcasts the search intent to the registered BPPs in the network with the BAP uri
* The BPPs send an ACK message to the gateway
* The BPPs published the service catalog as per the search intent back to the gateway
* The gateway forwards the catalogs to the requesting BAP.
* BAP sends ACK message to the gateway
* BAP aggregates the service catalogs across the BPPs and shows it to the consumer in the search page
* The search page displays results with ambulance service details with the type of ambulance, pricing, and add ons if any

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "search",

core\_version:

bap\_id: "<https://example-bap.com/>",

bpp uri:

bpp id :

message ID :

timestamp :

**Additional catalog Parameters**

bpp name

items :

item name/category : categories: Basic Ambulance/Advance Ambulance/Mortuary ambulance/Neonatal Ambulance/Patient Transport Vehicle/Air Ambulance

fulfillment type : emergency/non-emergency/rental

item price :

add ons : endo-tracheal intubations, administration of drugs or intravenous fluids, cardiac monitoring, and electrical therapy by qualified personnel, ECG monitor, defibrillator, intravenous and blood drawing tools etc

eta : max arrival time (arriving in 10 mins etc)

**JSON Example : [To Do]**

## **Booking Flows**

### **Selecting the required ambulance service, initializing the order, and confirming the service.**

Once a patient, caregiver or healthcare provider gets a search list of ambulance service providers available on the network with a brief catalog with ambulance type, price quote, next available schedule etc will go ahead and makes a selection on the service provider as per the required service, price and personal preference from the list.

### **Select : Selecting the desired ambulance service and add ons**

### Select-on\_select APIs are used to complete the selection process of the ambulance service along with all add-ons/offers from the provider’s catalog.

**Steps :**

1. User browses through the catalog and selects the type of ambulance, any add ons etc if applicable.
2. The selected item/service information is sent to the HSPA vis the “select API” call.
3. This call is made p2p between the EUA & HSPA without the gateway

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "select",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Order :

Provider : (id,name)

Items : (id, name= Advance Life Support, add ons)

Fulfillment id :

Fulfillment type : emergency

Start location :

End location :

Start time :

**JSON Example :**

### 

### **on\_select - Calculate the total order value and share a draft invoice**

On\_select api accepts the info provided by the user during select and calculates the quote of the service and returns back to the BAP.

**Steps :**

1. Once the user has selected the required ambulance type, location details, add on etc, the BPP calculates the total cost of the added item/services and sends back the quote via “on\_select”.

**Parameters passed :**

**Context Parameters**

Domain: Category code for ambulance service

country: "IND",

city:

action: "on\_select",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Order :

Provider : (id,name)

Items : (id, name= Advance Life Support, add ons)

Fulfillment : (id : , type : emergency)

Start location :

End location :

Start time :

End time :

Quote : (price, currency, value, breakup = cost, taxes, etc)

**JSON Example :**

### **init - Initialize an order by providing billing and/or shipping details**

Once the user is comfortable with the final quote received from the BPP, use the init API call to proceed with the booking by providing required billing details etc

**Steps :**

1. User clicks on book now once he/she is comfortable with the draft quote received.
2. Provides billing details confirming the pickup and drop location and time.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "init",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Order :

Provider : (id,name)

Items : (id, name= Advance Life Support, add ons)

Fulfillment : (id : , type : emergency)

Start location :

End location :

Start time :

End time :

Quote : (price, currency, value, breakup = cost, taxes, etc)

Billing : (address, email, phone)

Customer : (id, cred)

Other details may include (optional) : patient medical conditions, current condition: conscious, if booking for somebody else patient details - name, age, DOB, phone number (for coordination if different from registered) etc (mostly this info can flow from the profile created in the end-user application)

**JSON Example :**

### **on\_init - HSPA Send order object with payment details updated**

HSPA utilizes on\_init api to send the order terms including payment and settlement details for the service selected. The terms could include (not limited to) confirmation terms, fulfillment terms, update terms, cancellation terms & payment terms.

**Steps :**

1. As soon as the user shares the billing details, the user is shown the order terms sent by the HSPA along with a link to the payment page, if required.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "on\_init",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Order id :

Order state: provisionally booked

Provider : (id,name)

Items : (id, name= Advance Life Support, add ons)

Fulfillment : (id : , type : emergency)

Start location :

End location :

Start time :

End time :

Quote : (price, currency, value, breakup = cost, taxes, etc)

Billing : (address, email, phone)

Customer : (id, cred)

Other details may include (optional) : patient medical conditions, current condition: conscious, if booking for somebody else patient details - name, age, DOB, phone number (for coordination if different from registered) etc (mostly this info can flow from the profile created in the end-user application)

Payment :

Uri : COD

Type : on-fulfillment

Status : pending

**JSON Example :**

### **confirm - Confirms the order by agreeing to the fulfillment terms**

A user confirms the order by either making the payment or promising to pay. A EUA sends the confirmation via the “confirm API” call to the HSPA and Order ID for the same is generated by the HSPA.

Note : The payment construct is solely to be implemented by the application providers on the network, the confirm API only communicates the information confirming an order if a payment was made or was promised to be paid later.

**Steps :**

1. Post receiving the payment link/details from the HSPA, the user can make the payment by a mode enabled by the HSPA or EUA as per the configuration set by the service provider. For example- The service providers could configure a post-paid or pre-paid or partial pay workflow as per their requirements.
2. On payment or promise to pay the BAP sends the confirmation to the HSPA

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "confirm",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

transaction\_ID

timestamp :

**Message Parameters**

Order :

Provider : (id)

Order state : PROVISIONALLY\_BOOKED

Items : (id, quantity, name: advance life support)

fulfillment\_id : ( type : emergency)

Start location :

End location :

Start time :

End time :

Quote : (price, currency, value, breakup = cost, taxes, etc)

Billing : (address, email, phone)

Customer : (id, cred)

Other details may include (optional) : patient medical conditions, current condition: conscious, if booking for somebody else patient details - name, age, DOB, phone number (for coordination if different from registered) etc (mostly this info can flow from the profile created in the end-user application)

Payment :

Uri : COD

Type : on-fulfillment

Status : pending

**JSON Example :**

### **on\_confirm - HSPA Send confirmation of the booked service (Final active order)**

HSPA sends back booking confirmation with an Order ID, once it receives the mandatory information around payment/promise to pay and other details from the EUA against for a booked service or item. The confirmation response can include more details about the booked service. For example in the ambulance usecase the HSPA will send back an order ID,

**Possible Parameters**

* Order id
* Driver’s information: contact number, ID, Name
* Ambulance details : Type, Number, vehicle type etc
* Arrival time : ETA
* Tracking details/map : coordinates

**Steps :**

1. BPP receives the confirmation about the payment (paid or promise to pay) along with the fulfillmentID.
2. In response, the BPP provides confirmation on payment/promise to pay and generates an “order id” and transaction ID .
3. In addition, the BPP can also share further fulfillment details which can include the arrival status, ETA, Driver’s information, Ambulance Information etc.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "on\_select",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

Bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Transaction id :

Order id :

Order state: Booked

Provider : (id,name)

Items : (id, name= Advance Life Support, add ons)

Fulfillment : (id : , type : emergency)

Start location :

End location :

Start time :

End time :

Quote : (price, currency, value, breakup = cost, taxes, etc)

Billing : (customer name, address, email, phone)

Customer : (id, cred)

Other details may include (optional) : patient medical conditions, current condition: conscious, if booking for somebody else patient details - name, age, DOB, phone number (for coordination if different from registered) etc (mostly this info can flow from the profile created in the end-user application)

Payment :

Uri :payto://ban/98273982749428?amount=INR:110&ifsc=SBIN0000575&message=hello

Type : on-order

Status : paid

**JSON Example :**

## **Fulfillment flows**

### **status - Fetch the latest status of the order/booking**

A user can check the status of the ambulance service booking by using a simple function for “check status” in a BAP. The BAP calls the status API to get the latest status update for the order against the fulfillment and order ID

**Steps :**

1. The “status API “ enables a user to check the status of the ambulance booking done at any point until the post fulfillment stage. This function will be very relevant to scheduled ambulance booking where an ambulance is booked in advance and the information on the booking status is required to be checked for any cancellation, re-scheduling etc.
2. The BAP will use a status API call to fetch the current status of the booking mapped to the order id/fulfillment ID.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "status",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Order ID :

**JSON Example :**

### **on\_status - send the current service status against the order ID/ Ambulance Ride Status update to Patient**

A BPP can use the on\_status API to send a response back to the BAP for a status call or even can use it to regularly update the status of the booking/order against the order ID everytime the order state changes for an active booking.

**Note\*:** Not necessarily the BPP needs to have status API call from BAP for sending active status of the booking. The on\_status api can be used independently as well by the BPP.

**Parameters passed :**

**Context Parameters**

Domain : Category code for ambulance service

country: "IND",

city:

action: "status",

core\_version:

bap\_id: "<https://example-bap.com/>",

bap\_uri:

bpp\_id:

bpp\_uri:

message ID :

timestamp :

**Message Parameters**

Order ID :

Order state : on its way, booked, cancelled, rescheduled

**JSON Example :**

**Reference Screens**