

# Automatic Subscription creation

Frontend hits **cart** express checkout API with (where order ID is created) and then it hits **PG** with 1rs and upon PG success(status=2) , cart hits automatic system with **notify** request (**v1/order/notify**)

There are basic pre validation checks on the notify request served .Upon received valid notify request we further process the request otherwise NACK (Negative Acknowledgement)the request . While **nacking** the request we **update fulfilment with status 6 i.e Failure** .

1.check if **fulfilment service id is 17** .If service id is not 17 then nack the request .

2.Check if **item status is 2** i.e PG success .Read from item[0] position.

3.Check if **payment object**(payments) exists in the request .otherwise nack the request .

4.if payments object exists then check if **payment id i.e payments.SUBS\_ID** exists otherwise nack the request .

5.Check if **already subscription exists for the customer ID** .

a.**select from subscription where recharge number =x and productid=<items[0].product\_id> and status=<to be decided based on editable flag in meta data>**

b.if **editable flag under body.items.meta\_data** field reads **true** then it is considered as an **Edit request** and there is no check on status of the subscription.

b.1 : if in the select query (without status check ) ,count is 0 then create the request otherwise update the subscription.

c.if **editable** flag under body.items.meta\_data field reads **false** or not present then it is considered as an **Create request** and there is check on status if that lies in **Active(1)** or **Cancelled(0)** state.

c.1: if query count with status check is >0 then nack the order stating that Already subscription exists for this .

c.2: If no data exists against the select query then proceed further .

6.Get **product ID** from the request body and get cvr data against the PID i.e operator,service,paytype,circle and put into a request object .

7.Now proceed to **create the subscription**. verify mandatory checks for subscription creation

a.merchant id, product id ,customer id(req body.customer\_id),customer email,operator name,phone number,payment id (subs\_id),created at, order id , subscriptionInfo(interval and expiry date). If any of the mandatory param is missing then nack the request .

b.Read days\_before\_renewal from meta data .By default the value is -1 .Based on this value fulfilment time is calculated i.e due date- days before renewal value.

c.Notify at time is fulfilment time-1

d.Initialize all date fulfilment time,notify at ,bill fetch date, due date as 0000. Set next bill fetch date as now+ (days based on the operator selection).By default the value for next bill fetch date is now + 20 days .

e.verify mandatory check on recharge number and phone number .if not available then nack the request .

f .Now finally insert the prepared data in **subscription** table with fulfilment id=0 .If there is any error in inserting then nack the request .

8.Create **bill record in reminder**

a. hit **getBill** api against the customer id , recharge number, product id and operator combination. If no data is received then get data from **ES** using PID and recharge number combination with time range till current time stamp from validation index(Currently we store values upto 6 months).

b. if **no data is received from ES** then prepare the data and call **createBill** API (next bill fetch date=now+5 days) and then hit getBill api for the same.

c.if data is found from ES then set the **gateway** and next bill fetch date=now+5 days. Hit createBill api and then get bill api for the same.

9.Now we hit fulfilment for **fulfilment id creation** and the same is updated in subscription table.