**ROI-PARCEL SERVICE**

RateService:

-----------------

- Couple of things we found interesting while working on the RateService.

FedEx-

------------

1.) In Fedex rate SOAP request we have two options. either we give option as "List" to give rate for all Service or we can give request for only 1 specific Service.In our case we are giving option as List and taking out the non requested service at the time of agregation and preparing the final rateResponse.

2.) There are categories in FedEx services called as "US Express","US Express Freight","Fedex Ground","International Express Freight" as per their Fedex-Service\_Guide\_2016.pdf.

- For our Rate Service perspective when we say WAY="Air" in that case in Fedex we are considering all services comming under "US Express".

3.) In FedEx we can send request with tag <ReturnTransitAndCommit>true</ReturnTransitAndCommit> to get the commitment date and time. But we will not get commitment date and time for any of the Ground Services in reference with the confirmation with FedEx technical support team. However Ground Services Gives in Days <TransitTime>ONE\_DAY<TransitTime> while all the "US Express" will give time in UTC <DeliveryTimeStamp>2016-03-17T16:30:00<DeliveryTimeStamp>.

Note:- Handling of <TransitTime>ONE\_DAY<TransitTime> is a challenge in our code. After a bunch of trials on the Ground service, as a solution we took currentdate and added one day to it for the sorting purpose.

4.) For sorting based on the amount we are considering <TotalNetChargeWithDutiesAndTax>, as it sums up all the detailed charges that is for all the packages.

UPS-

----------

1.) In Ups Soap Rate request the service responds always for all the services even if we request any specific service. Hence we are maintaining the list of services as it responds for WAY,Carier and removing unwanted services from the reponse at the stage of aggregation and sorting in the code.

2.) A point to note is none of the services in UPS gives the CommitmentDateAndTime it always gives number of days, and in UPS Air for applicable service it gives days and time as response.

<rate:GuaranteedDelivery>

<rate:BusinessDaysInTransit>1</rate:BusinessDaysInTransit>

<rate:DeliveryByTime>10:30 A.M.</rate:DeliveryByTime>

</rate:GuaranteedDelivery>

To get the GuranteedDelivery we have to set <RequestAction>Shop</RequestAction> instead of <RequestAction>Rate</RequestAction>

3.) For UPS Ground we will not get the GuranteedDelivery days or time ,so to get the delivery date and time we need to use the UPS TimeInTransit webservice API. this will give response as below

<tnt:EstimatedArrival>

<tnt:Arrival>

<tnt:Date>20160322</tnt:Date>

<tnt:Time>103000</tnt:Time>

</tnt:Arrival>

<tnt:BusinessDaysInTransit>1</tnt:BusinessDaysInTransit>

</tnt:EstimatedArrival>

In our poc currently we are not hitting the UPS TimeInTransit webservice to get commitedDelivery date and time, however, managed in code to solve this issue, by adding the number of days with the current UTF-Time, and is considered for further process in aggregation, filtering.

4.) UPS has different webservice for Air Freight Shipments called "Gemini API". Any package above 150 lbs has to be done through the Gemini API. Researched as a result of one of the tests done with Default ShipmentApi for Shipment Weight above 150lbs resulted in a message as "package weight limit exceeded"

In our POC currently we are testing with weight less than 150lbs we are not using the Gemini API as it is applicable for 150lbs above packages.

5.) In UPS for all the different Services they have different Service Code like 03=UPS GROUND,11=UPS Standard and some varies from country to country.

- When UPS is returning the rate response they are giving response with service code,we are converting them in to the respective service names at the time of preparing the GenericRateResponse.

6.) UPS accepts the units of weight in LBS is different from FedEx, however we are handling it in the request, as well as in the response by replacing the LB with LBS to make it similar.

PermaStore Details:

---------------------------

- ParcelShipperAddress

-----------------------------

In ParcelShipperAddress of permastore we have ShipperID and its corresponding details, such as "CompanyName", "PhoneNumber", "StreetLine1", "City", "StateOrProvinceCode", "PostalCode" and "CountryCode".

Note: ShipperId is one of the element from the input/request.

- FedExCredentials

--------------------------

In FedExCredentials perma store has values of "Key", "Password", "AccountNumber" and "MeterNumber"

- FedExProperties

------------------------

In FedEx Properties perma store has values of "ServiceId\_Rate", "Major\_Rate", "ShipTimestamp\_Void", "ServiceId\_Void", "Major\_Void", "DeletionControl\_Void", "PaymentType\_ship", "DropoffType\_ship", "PackagingType\_ship", "LabelFormatType\_ship".

- UpsCredentials

-----------------------

In UPS Credentials perma store has values of "Username", "Password" and "AccessLicenseNumber".

- UpsProperties

----------------------

In UPS Properties, perma store has values of "RequestOption\_Rate", "XpciVersion\_Rate", "PickupTypeCode\_Rate", "CustomerClassificationCode\_Rate",

"PackagingTypeCode\_Rate", "XpciVersion\_ship", "LabelStockSizeHeight\_ship", "LabelStockSizeWidth\_ship"

Flow is as Follows of RateService of FedEx And UPS

-----------------------------------------------------------------

BaseRoute

---------------

1.) Converts the xml/soap request to the JSON Request

RateRequest-ExecutionRoute

--------------------------------------

1.) Converts json data into the rateRequest xml string

2.) Validates the xml with the xml schema for required Tags and few fixed data like the tag <way></way> can have this 3 values only "Ground", "Air" and "Freight". anything apart from this will roll back the process and throws error.

3.) The rateRequest xml string is sent to multiple route such as "fedExFilter route" and "upsFilter route" using multicast, and this fedExFilter route and upsFilter route will filter the carrier type and if it matches then it will sent the rateRequest to the respective implementation route.

FedEx-Impl route

-----------------------

1.) The rateRequest xml string from RateRequest-ExecutionRoute is transformed to a proper structured xml using xslt.

2.) Few elements like Credentials, shipperDetails are added to the rateRequest xml of FedEx using Bean which appends the required Credential, shipperDetails tags by taking values from perma store.

3.) The rateRequest is transformed to soap envelope and send to the cxf endpoint of FedEx rate request as soap envelope request.

4.) The SOAP Reponse from FedEx is converted back to the GenericResponse xml using xslt.

5.) The weight from the SOAP response comes in LB form which is again converted to LBS

UPS-Impl route

----------------------

1.) The rateRequest xml from RateRequest-ExecutionRoute is transformed to a proper structured xml using xslt.

2.) Few tags like Credentials, shipperDetails are added to the rateRequest xml of UPS using Bean which appends the required Credential, shipperDetails tags by taking values from perma store.

3.) The rateRequest xml is transformed to soap envelope and send to the cxf endpoint of UPS as soap envelope request.

4.) The SOAP Reponse of UPS has Service code which is convert to its respective service type using perma store data.

5.) Again the soap reponse is converted back to the GenericResponse xml using xslt.

RateRequest-ExecutionRoute

-------------------------------------

1.) The GenericeResponse xml of UPS and FedEx is sent back to ExecutionRoute.[Even for single carrier reponse , the aggregation will sort the rates]

2.) ExectionRoute will aggregate the reponse using rate and send the response back to ui.

Note: Aggregation of Rate is done by taking <TotalNetChargeWithDutiesAndTax> tag's amount.

ShipmentService:

-----------------------

- Couple of things we found while working on the ShipmentService.

FedEx:

-----------

1.) In Shipment FedEx to generate the label we have <LabelSpecification> <ImageType>PDF</ImageType></LabelSpecification> which takes ImageType as "PDF","ZPLII" ,"PNG" and so on.

- For our request we have only taken ZPLII

UPS:

---------

2.) In Shipment of UPS to generate the label we have <LabelImageFormat> <Code>ZPL<Code> </LabelImageFormat> and we have to mention the size of the label as

<LabelStockSize>

<Height>6</Height>

<Width>4</Width>

</LabelStockSize>

- For our request we have only taken ZPL

The LabelImageFormat Code has "GIF", "ZPL", "EPL" and "SPL".

Note: Here the LabelStockSize is fixed[6,4] if the Height and Width is increased or decreased we get an Exception as "Missing or invalid label specification label stock size width"

Flow is as Follows of ShipmentService of FedEx And UPS

-------------------------------------------------------------------------

BaseRoute

---------------

1.) Converts the xml/soap request to the JSON Request

CreateShipRequest-ExecutionRoute

-----------------------------------------------

1.) Converts json data into the createShipRequest xml

2.) Validates the xml with the xml schema for required Tags and data if not matched then request will roll back the process and throws error in json.

3.) The createShipRequest xml is sent to multiple route such as "fedExFilter route" and "upsFilter route" using multicast, and this fedExFilter route and upsFilter route will filter the carrier type and if it matches then it will sent the createShipRequest xml to corresponding implementation route.

FedEx-Impl route

------------------------

1.) The createShipRequest xml from CreateShipRequest-ExecutionRoute is formed to a proper structured xml using xslt.

2.) Few tags of Credentials are added to the createShipRequest xml of FedEx using Bean which appends the required Credential tags and takes values from perma store.

3.) The createShipRequest xml is transformed to soap envelope and send to the cxf endpoint of FedEx Create Shipment request as soap envelope request.

4.) The SOAP Reponse from FedEx is converted back to the GenericResponse xml using xslt.

UPS-Impl route

----------------------

1.) The createShipRequest xml from createShipRequest-ExecutionRoute is formed to a proper structured xml using xslt.

2.) Few tags of Credentials are added to the createShipRequest xml of UPS using Bean which appends the required Credential tags and takes values from perma store.

3.) The createShipRequest xml is transformed to soap envelope and send to the cxf endpoint of UPS as soap envelope request.

4.) The response contains ShipmentIdentificationNumber and TrackingNumber which is same id for both used to void the shipment

5.) Again the soap reponse is converted back to the GenericResponse xml using xslt.

VoidShipmentService:

-----------------------------

- Couple of things we found while working on the VoidShipmentService.

FedEx:

-----------

1.) In Void or Delete Shipment for FedEx, we need to have the TrackingIdType, TrackingNumber and DeletionControl.

- TrackingIdType has "EXPRESS", "FREIGHT", "GROUND" and "FEDEX"

- DeletionControl has verious option of deleting a shipment like "DELETE\_ALL\_PACKAGES", "DELETE\_ENTIRE\_CONSOLIDATION", "DELETE\_ONE\_PACKAGE" and "LEGACY".

- We have use EXPRESS and "DELETE\_ALL\_PACKAGES" we have used

UPS:

--------

2.) In Void Shipment of UPS, we need of ShipmentIdentificationNumber and TrackingNumber.

- Both ShipmentIdentificationNumber and TrackingNumber is same.

NOTE: The Void of UPS cant be deleted util "12" 0r "24" hours from the time of shipment.

Flow is as Follows of ShipmentService of FedEx And UPS

-------------------------------------------------------------------------

BaseRoute

---------------

1.) Converts the xml/soap request to the JSON Request

voidShipRequest-ExecutionRoute

-------------------------------------------

1.) Converts json data into the voidShipRequest xml

2.) Validates the xml with the xml schema for required Tags and data if not matched then request will roll back the process and throws "{ERROR NAME}" error.

3.) The voidShipRequest xml is sent to multiple route such as "fedExFilter route" and "upsFilter route" using multicast, and this fedExFilter route and upsFilter route will filter the carrier type and if it matches then it will sent the voidShipRequest xml to corresponding implementation route.

FedEx-Impl route

-----------------------

1.) Few tags of Credentials are added to the voidShipRequest xml of FedEx using Bean which appends the required Credential tags and takes values from perma store.

2.) The voidShipRequest xml is transformed to soap envelope and send to the cxf endpoint of FedEx void Shipment request as soap envelope request.

3.) The SOAP Reponse from FedEx is converted back to the GenericResponse xml using xslt.

UPS-Impl route

---------------------

1.) Few tags of Credentials are added to the voidShipRequest xml of UPS using Bean which appends the required Credential tags and takes values from perma store.

2.) The voidShipRequest xml is transformed to soap envelope and send to the cxf endpoint of UPS as soap envelope request.

3.) Again the soap reponse is converted back to the GenericResponse xml using xslt.