**FreeCall/Labor Credit BulkAPI Documentation**

1. **Collect\_S4**
2. This workflow collects from a common path where SOM places its file :

*/usr/sap/interfaces/SP0/outbound/AR/thirsty/data*

And after collection the files are also archived in location:

*/usr/sap/interfaces/SP0/outbound/AR/thirsty/archive*

1. Based on the filename we route the file to different directories

**Billing:**

*Filename: INV\_Thirsty\_output\_(8 numbers combo)\_(6 numbers combo).txt*

*🡪 /usr/sap/interfaces/MZP/ServiceBilling/InvoiceToThirsty*

**AP Payment & Free Call:**

*Filename: AP\_Payment\_Date\_(8 numbers combo).txt 🡪*

*/usr/sap/interfaces/MZP/ServiceBilling/back feed*

*Filename: Free\_CallLimit\_Thirsty\_output.\* 🡪*

*/usr/sap/interfaces/MZP/ServiceBilling/back feed*

1. **FreeCall\_SplitFiles**

We collect the files from location: */usr/sap/interfaces/MZP/ServiceBilling/back feed*

**SplitFiles:**

***Consume:***

1. We first check if the record present in the file is valid or not this is done with the help of a function *isValidRecord,* we check if the records outletBP, outletACN, counterId are not null or empty ,

if so we send an appropriate message back, we then add this to an invalid list , which we later use to send out notifications.

1. If there are no issues with the record to the record we then add to a list called *list\_fc\_records(*used to make BulkAPI’s for thirsty)
2. If the list size(list\_fc\_records) has reached 5000 (i.e 5000 records are present in the list) we then create a BulkApi file for thirsty (*why? : thirsty can process on 5000 records at a time so that is why we do this*)

We first generate a header for the file and then route it out , after which we traverse the list and the records are then routed out 1 by 1.

***Drain:***

1. We create the API file for the rest of the records which are present it’s a copy of Step C in consume:

Reason:

7000 RECORDS -> 5000 1 bulk API FILE IN Consume block, rest 2000 we create 1 bulk API file in Drain Block.

11000 records -> 2x 5000 2 bulk API File in consume , 1000 1 bulk API file in Drain block.

1. For the invalid records present , we traverse the list used in Step A in consume block, and then combine all the error in a record and send them.

These files are then stored in location : */usr/sap/interfaces/MZP/ServiceBilling/BulkAPI/process*

1. **FreeCall\_BulkAPI**

**Batch:**

We collect the files from location : */usr/sap/interfaces/MZP/ServiceBilling/BulkAPI/process*

**Route\_To\_WFB:**

1. The records are all in byte array form , so when we receive a file , we combine all the byte Arrays aka records into 1 byte array.
2. We then create a wrapper where in which we store details such as the byte array we created , the filename as well as the count
3. The Wrapper is then sent to *FreeCall\_BulkAPI\_RT* through a consumeCycleUDR.

**Real Time:**

**RequestCycle:**

**WFB\_Handler:**

1. After we receive the file from batch we extract the Wrapper from the consumeCycleUDR’s data field and route to *Send\_To\_Thirsty* (Analysis Agent)

**Send\_To\_Thirsty:**

1. So after receiving the Wrapper from the WFB\_Handler Analysis Agent we create a RestCycleUDR (a rest request) for thirsty.

We create a POST Request , and assign the necessary fields for the POST Operation such as :

*Header: "Content-Type", "application/json"*

*Resource URI: /services/data/v58.0/jobs/ingest/ (Thirsty End point to be used)*

*Body: byte array from the Wrapper*

We then update the wrapper’s reqType to InitialCall and also the Rest request context field to have the wrapper. And then we route the RestCycleUDR we just created to SFDC\_Rest(Rest Client Agent aka Thirsty).