TAAS - Feature #497

Booking System Dashboard Chart API

02/03/2025 02:34 PM - Arathy PS

Status:	Coding Done	**Start date:**	02/03/2025
Priority:	Normal	**Due date:**	02/08/2025
Assignee:	Amal Ck	**% Done:**	60%
Category:		**Estimated time:**	0:00 hour
Target version:	22.12	**Spent time:**	0:00 hour
Owner(Agency):	Travvise	**Tested By:**	
Time Taken(HH):		**Code Reviewed By:**	
Module:	TAAS-General		
Description			

History

#1 - 02/03/2025 02:48 PM - Arathy PS

- File Top Performers.pdf added

- File Flights Target Acheivement.pdf added

- File Profitability Analysis _ Inventory (1).pdf added

- File Profitability Analysis _ Inventory.pdf added

- File Supplier Sales.pdf added

#2 - 02/04/2025 03:44 PM - Anil KV

```
Chart:-
=========
API Name:- SaleRefundTargetSummaryTaasIntegration

=> Excel Format and Columns
----------------------------
Columns:-
# | Key | Default Value | Mandatory | Values | Remarks
1)Input Filter
1.1 | strServiceGroupingType | SERVICE | No | SERVICE; FIXED_PACKAGE; INVENTORY | Service Grouping Type. Empty
 means ALL, 'SERVICE; FIXED_PACKAGE' Allowed
---
2)Input Grouping:-
---

=>Input Filter:-
----------------
->Period(From and To Date) -> Mandatory -> datFromDate, datToDate
->Period Type -> BOOKING/ISSUE/CANCEL_DATE, DOCUMENT_DATE -> Empty means BOOKING/ISSUE/CANCEL/DOCUMENT_DATE Da
te
-> Branch -> lstBranchCode
-> Service Grouping Type(Sys) -> lstSysServiceGroupingType ->  SERVICE; FIXED_PACKAGE; INVENTORY -> service.tb
l_service_grouping
-> Service Group -> lstServiceGroup -> Service Group Master Inputs -> service.tbl_service->fk_service_grouping
_id
???-> Service Type(Sys) -> lstSysServiceType ->  AIR_TICKET, HOTEL, TRANSFER, VISA, OTHER_SERVICE, CARGO, COUR
IER, ANCILLARY, INSURANCE, DRIVING_LICENSE, CRUISE, PACKAGE_SERVICE, PACKAGE_FIXED, INVENTORY_UNIT/QUANTITY, I
NVENTORY_SEQUENCE ->service.tbl_service->sin_service_sys_type/sin_package_sys_type/sin_inventory_sys_type
*If need then add to Set Target also like Service Grouping Type
-> Service Sales Category(Sys) -> lstSysServiceSalesMapping -> TICKET, HOTEL, TRANSFER, VISA, OTHER_SERVICE, P
ACKAGE, INVENTORY -> service.tbl_supp_doc_no->sin_supp_doc_category
-> Service Master Category -> From Master Grouping GUI
-> Service Master Group -> From Master Grouping GUI
-> Service Master Type -> From Master Grouping GUI
-> Service Master Family -> From Master Grouping GUI
-> Service -> lstServiceCode, lstServiceErpCode -> From Service GUI -> service.tbl_supp_doc_no->fk_service_id
-> Service Provider -> lstServiceProviderCode, lstServiceProviderErpCode -> From Service Provider Master GUI -
```

```
> service.tbl_service_provider->vhr_service_provider_code
-> Supplier -> lstSupplierCode, lstSupplierErpCode -> From Supplier Master GUI -> service.tbl_supp_doc_no->fk_
supplier_id/service.tbl_supp_doc_no->jsn_related_data->SUPPLIER_CODE
-> Customer -> lstCustomerCode, lstCustomerErpCode -> From Customer Master GUI, Consider Sub Customers(Cash sa
les Case) -> service.tbl_supp_doc_no->fk_customer_ac_id or fk_actual_customer_ac_id
-> Commission Agent -> lstCommissionAgentCode, lstCommissionAgentErpCode -> From Commission Agent Master GUI.
-> service.tbl_supp_doc_no->fk_commission_agent_ac_id
-> Booking Counter Staff -> lstBookingCounterStaffCode, lstBookingCounterStaffErpCode -> From Counter Staff Ma
ster GUI, -> service.tbl_supp_doc_no->jsn_related_data->ISSUING_STAFF_CODE
-> Issuing Counter Staff -> lstIssuingCounterStaffCode, lstIssuingCounterStaffErpCode -> From Counter Staff Ma
ster GUI, -> service.tbl_supp_doc_no->jsn_related_data->BOOKING_STAFF_CODE
-> Airline -> lstAirlineChrCode, lstAirlineNumCode -> From Airline Master GUI -> service.tbl_supp_doc_no->jsn_
related_data->AIR_NUM_CODE/AIR_CHR_CODE
-> Hotel Chain -> lstHotelChain -> From Hotel Chain Master GUI -> hotel.tbl_hotel_chain->vhr_hotel_chain
-> Hotel -> lstHotelCode, lstHotelErpCode -> From Hotel Master GUI -> service.tbl_supp_doc_no->jsn_related_dat
a->HOTEL_CODE/HOTEL_ERP_CODE
-> GDS -> lstGds -> AMADEUS, TRAVELPORT, SABRE, GALILEO, WORLDSPAN, FARELOGIX  -> service.tbl_supp_doc_no->jsn
_related_data->GDS
-> User -> lstLoginName -> From User Master
-> Currency -> strCurrencyCode -> If Base1 then NOT SET -> Empty means Base1(Agency) Currency
* ERP CODE and CODE Input option need
->Input -> lstExclude
-AGAINST_SOLD
-AGAINST_SALE
-INTER_BRANCH_SOLD
-INTER_BRANCH_SALE
-INTER_BRANCH_REFUNDED
-INTER_BRANCH_REFUND

=>Input Grouping:-
----------------
->strGrouping -> Mandatory

- Branch -> BRANCH
- Service Grouping Type -> SYS_SERVICE_GROUPING_TYPE
- Service Group -> --
- Service Sales Category -> SYS_--
- Service Master Category
- Service Master Group
- Service Master Type
- Service Master Family
- Service
- Service Provider
- Supplier
- Customer
- Commission Agent
- Booking Counter Staff
- Issuing Counter Staff
- Airline
-> Hotel Chain
- Hotel
- GDS
- User
- Currency
- Month
- Year

=>Input Sub Grouping:-
----------------
->strSubGrouping -> Mandatory

- Branch -> BRANCH
- Service Grouping Type -> SYS_SERVICE_GROUPING_TYPE
- Service Group -> --
- Service Sales Category -> SYS_--
- Service Master Category
- Service Master Group
- Service Master Type
- Service Master Family
- Service
- Service Provider
- Supplier
- Customer
- Commission Agent
- Booking Counter Staff
```

```
- Issuing Counter Staff
- Airline
-> Hotel Chain
- Hotel
- GDS
- User
- Currency
- Month
- Year

=>Input Record Limit:-
----------------
-> intRecordLimit = 0;
* Record Limit = 0 -> 0/Empty That means all records

=> Target Input:-
=================
-> strTargetBase -> Parent - COMPANY, BRANCH, ....

=>Output:-
----------------
-> ERP Code -> Based On Grouping
-> Code -> Based On Grouping
-> Name  -> Based On Grouping
-> Sub ERP Code -> Based On Sub Grouping
-> Sub Code -> Based On Sub Grouping
-> Sub Name  -> Based On Sub Grouping
-> Booking/Issue/Sale date --> CS logic - Not OUTPUT -> LOGIC-2
-> Cancel/VOID/Refund date --> CS logic - Not OUTPUT -> LOGIC-2

-> Currency --Base1(Agency) Currency
-> Booked Count
-> Booked Cost
-> Booked Price
-> Booked Profit
-> Issued Count
-> Issued Cost
-> Issued Price
-> Issued Profit
-> Sold Count
-> Sold Cost
-> Sold Price
-> Sold Profit
-> Canceled Count
-> Canceled Cost-> Received from Supplier
-> Canceled Price -> Paid to Customer
-> Canceled Profit
-> Direct Void Count
-> Direct Void Cost -> 0.00
-> Direct Void Price -> 0.00
-> Direct Void Profit -> 0.00
-> Refunded Void Count
-> Refunded Void Cost -> Received from Supplier
-> Refunded Void Price -> Paid to Customer
-> Refunded Void Profit
-> Refunded Count
-> Refunded Cost -> Received from Supplier
-> Refunded Price  -> Paid to Customer
-> Refunded Profit
---------------------
-> Target Amount
-> Target Profit
```

#3 - 02/04/2025 03:45 PM - Anil KV

- Due date set to 02/08/2025

- Assignee changed from travvise Admin to Amal Ck

#4 - 02/05/2025 05:44 PM - Anil KV

```
LOGIC-2
==========
intRootTargetType = 1
intSubTargetType = 1
```

```
// @@@ WERE
strSqlWhere = ""
strSqlBookingIssueSaleDate = ""
strSqlVoidCancelRefundDate = ""
// Period
if Period Type == BOOKING/ISSUE/CANCEL_DATE
    strSqlWhere = (supp.dat_issue BETWEEN datFromDate AND datToDate OR supp.dat_cancel BETWEEN datFromDate AND
 datToDate)
    strSqlBookingIssueSaleDate = "supp.dat_issue"
    strSqlVoidCancelRefundDate = "supp.dat_cancel"
else if Period Type = DOCUMENT_DATE
    strSqlWhere = (supp.dat_sale BETWEEN datFromDate AND datToDate OR supp.dat_refund BETWEEN datFromDate AND
datToDate)
    strSqlBookingIssueSaleDate = "supp.dat_sale"
    strSqlVoidCancelRefundDate = "supp.dat_refund"
else
    strSqlWhere = ((supp.dat_booking BETWEEN datFromDate AND datToDate AND supp.sin_sys_sale_side_status == 0)

            OR (supp.dat_sale BETWEEN datFromDate AND datToDate AND supp.sin_sys_sale_side_status == 1)
            OR (supp.dat_sale BETWEEN datFromDate AND datToDate AND supp.sin_sys_sale_side_status == 2)
            OR (supp.dat_cancel BETWEEN datFromDate AND datToDate AND supp.sin_sys_refund_side_status IN (1,2,
3,6))
                        OR (supp.dat_refund BETWEEN datFromDate AND datToDate AND supp.sin_sys_refund_side_sta
tus IN (4,5))
    strSqlBookingIssueSaleDate = CASE
                WHEN supp.sin_sys_sale_side_status == 0 THEN supp.dat_booking
                WHEN supp.sin_sys_sale_side_status == 1 THEN supp.dat_sale
                WHEN supp.sin_sys_sale_side_status == 2 THEN supp.dat_sale
                 END
    strSqlVoidCancelRefundDate = CASE
                WHEN supp.sin_sys_refund_side_status IN (1,2,3,6) THEN supp.dat_cancel
                WHEN supp.sin_sys_refund_side_status IN (4,5) THEN supp.dat_refund
                 END

// Branch
if lstBranchCode.len > 0
    strSqlWhere += AND br.vhr_branch_code IN lstBranchCode
// Service Grouping Type(Sys) -> 1=SERVICE; 2=FIXED_PACKAGE; 3=INVENTORY
if lstSysServiceGroupingType.len > 0
    lstSysServiceGroupingTypeNew = []
    for
        if == SERVICE
            lstSysServiceGroupingTypeNew.add(1)
        ...
    strSqlWhere += AND srvg.sin_grouping_sys_type IN lstSysServiceGroupingTypeNew

-----
if lstServiceCode.len > 0
    strSqlWhere = AND supp.jsn_related_data->strServiceCode IN lstServiceCode
-----

// @@@ GROUP
strSqlGrouping = ""
strSqlErpCode = ""
strSqlCode = ""
strSqlName  = ""
if strGrouping == Branch:
    strSqlGrouping = "supp.fk_branch_id"
    strSqlErpCode = ""
    strSqlCode = "br.vhr_branch_code"
    strSqlName  = "br.vhr_branch_name"
    intRootTargetType = 5

------
if Month(Month means - Month + Year) or Year
    strSqlGrouping += strSqlBookingIssueSaleDate, strSqlVoidCancelRefundDate
------

// @@@ SUB GROUP
strSqlSubGrouping = ""
strSqlSubErpCode = ""
strSqlSubCode = ""
strSqlSubName  = ""
if strSubGrouping == Branch:
    strSqlSubGrouping = "supp.fk_branch_id"
```

```
    strSqlSubErpCode = ""
    strSqlSubCode = "br.vhr_branch_code"
    strSqlSubName  = "br.vhr_branch_name"
    intRootTargetType = 5

strFullSqlGrouping = strSqlGrouping
if strSqlSubGrouping
    strFullSqlGrouping = strFullSqlGrouping + ',' + strSqlSubGrouping

------
if Month(Month means - Month + Year) or Year
    strSqlGrouping += strSqlBookingIssueSaleDate, strSqlVoidCancelRefundDate
------

// @@@ LIMIT
strSqlLimit = ''
if intRecordLimit > 0
    strSqlLimit = "LIMIT =" intRecordLimit

// @@@ SQL
SELECT strSqlErpCode AS vhr_erp_code
    strSqlCode AS
    strSqlName AS
    strSqlSubErpCode AS
    strSqlSubCode AS
    strSqlSubName AS
        MIN(strSqlBookingIssueSaleDate) AS dat_issue_or_sold
    MIN(strSqlVoidCancelRefundDate) AS dat_cancel_refund
    'QAR->BASE1' AS vhr_currency
    COUNT(*) FILTER (WHERE supp.sin_sys_sale_side_status = 0) AS bin_booked_ount
    SUM(supp.jsn_related_data->dblBaseCurSupplierAmount::DOUBLE) FILTER (WHERE supp.sin_sys_sale_side_status =
 0) AS dbl_booked_cost
    ----
        ----
FROM service.tbl_supp_doc_no AS supp
    LEFT JOIN organization.tbl_branch br
        ON supp.fk_branch_id = br.pk_branch_id
        LEFT JOIN service.tbl_service_grouping srvg
        ON supp.fk_service_id = srvg.pk_service_grouping_id
        ---
        ---

WHERE strSqlWhere
GROUP BY strFullSqlGrouping
ORDER BY
strSqlLimit;

// @@@ TARGET
**Get FY Id based on datToDate
intfinancialYearId =
sin_month + / + FY.year AS
***Main sql logic based on intRootTargetType and intSubTargetType
strSqlErpCode =
    strSqlCode=
    strSqlName=
    strSqlSubErpCode=
    strSqlSubCode=
    strSqlSubName =

strTargetWhere = fk_financial_year_id = intfinancialYearId AND sin_root_splitup_type = intRootTargetType AND i
ntSubTargetType = intSubTargetType AND sin_target_type != 8
strTargetJoinOn = "supp.fk_branch_id = tar.pk_branch_id"
...
if sin_root_splitup_type == 5
    strTargetJoinOn =
...

SELECT strSqlErpCode AS vhr_erp_code
    strSqlCode AS
    strSqlName AS
    strSqlSubErpCode AS
    strSqlSubCode AS
    strSqlSubName AS

            dbl_revenue_target AS dbl_profit_target,
```

```
            dbl_yield_target AS dbl_sale_target
        FROM target.tbl_set_target tarm
            INNER JOIN target.tbl_set_target_splitup tard
                ON tarm.pk_set_target_id = tard.fk_set_target_id
        WHERE strTargetWhere

// @@@ In css
lstMdlSalesSummaryOutput = []
for
    model = new model

    ...

    if group or su group -> Month or Year
        strIssueMonthYear = ""
        strCancelMonthYear = ""
        if Month
            strIssueMonthYear = dat_issue_or_sold.month / dat_issue_or_sold.year
            strCancelMonthYear = dat_cancel_refund.month / dat_cancel_refund.year

        if Year
            strIssueMonthYear = dat_issue_or_sold.year
            strCancelMonthYear = dat_cancel_refund.year

        // Issue/Sold
        if group
                     mdl = lstMdlSalesSummaryOutput.find(mdl.group = strIssueMonthYear)
        if sub group
             mdl = lstMdlSalesSummaryOutput.find(mdl.subgroup = strIssueMonthYear)

        if mdl:
            mdl.booking count += model.booking count
            --
            --All Booking and Issue and Sold
                        // --
            clear the model Booking and Issue and Sold
        else
            modelNew = model Booking and Issue and Sold set to new model
            lstMdlSalesSummaryOutput.add(modelNew)

        // Void/Cancel/Refunded
        if group
                     mdl = lstMdlSalesSummaryOutput.find(mdl.group = strCancelMonthYear)
        if sub group
             mdl = lstMdlSalesSummaryOutput.find(mdl.subgroup = strCancelMonthYear)

        if mdl:
            mdl.Canceled count += model.Canceled count
            --
            --All Void and Cancel and Refunded
                        // --
            clear the model Booking and Issue and Sold
        else
            modelNew = model Void and Canencl and refunded set to new model
            lstMdlSalesSummaryOutput.add(modelNew)

                // Set Target
        group name base
```

#5 - 02/05/2025 05:50 PM - Anil KV

```
LOGIC-1
==========

// @@@ WERE
strSqlIssueSoldWhere = ""
strSqlCancelRefundedWhere = ""
// Period
if Period Type == BOOKING/ISSUE/CANCEL_DATE
    strSqlIssueSoldWhere = supp.dat_issue BETWEEN datFromDate AND datToDate
    strSqlCancelRefundedWhere = supp.dat_cancel BETWEEN datFromDate AND datToDate
else if Period Type = DOCUMENT_DATE
    strSqlIssueSoldWhere = supp.dat_sale BETWEEN datFromDate AND datToDate
    strSqlCancelRefundedWhere = supp.dat_refund BETWEEN datFromDate AND datToDate
else
```

```
    strSqlIssueSoldWhere = ((supp.dat_booking BETWEEN datFromDate AND datToDate AND supp.sin_sys_sale_side_sta
tus == 0)
            OR (supp.dat_sale BETWEEN datFromDate AND datToDate AND supp.sin_sys_sale_side_status == 1)
            OR (supp.dat_sale BETWEEN datFromDate AND datToDate AND supp.sin_sys_sale_side_status == 2))
    strSqlCancelRefundedWhere = ((supp.dat_cancel BETWEEN datFromDate AND datToDate AND supp.sin_sys_refund_si
de_status IN (1,2,3,6))
                        OR (supp.dat_refund BETWEEN datFromDate AND datToDate AND supp.sin_sys_refund_side_sta
tus IN (4,5))

OTHER LOGIC Check LOGIC-2
-------------------------

SQL->

SELECT

FROM
(BOOKING/ISSUE/SOLD
    strSqlBookingIssueSaleDate AS dat_date
    CASE
                    WHEN supp.sin_sys_sale_side_status == 0 THEN Booking
                    WHEN supp.sin_sys_sale_side_status == 1 THEN Issue
                    WHEN supp.sin_sys_sale_side_status == 2 THEN Sold
                     END str type

UNION ALL

VOID/CANCEL/REFUNED VOID/REFUNED
strSqlVoidCancelRefundDate AS dat_date)
CASE

SQL GEOSS TAB -> Check Ageing
```

#6 - 02/08/2025 07:10 PM - Amal Ck

- Status changed from New to Coding Started

- % Done changed from 0 to 30

#7 - 02/14/2025 09:53 AM - Amal Ck

- Status changed from Coding Started to Coding Done

- % Done changed from 30 to 60

Files

Top Performers.pdf	1.41 MB	02/03/2025	Arathy PS
Flights Target Acheivement.pdf	1.78 MB	02/03/2025	Arathy PS
Profitability Analysis _ Inventory (1).pdf	1.41 MB	02/03/2025	Arathy PS
Profitability Analysis _ Inventory.pdf	1.28 MB	02/03/2025	Arathy PS
Supplier Sales.pdf	1.4 MB	02/03/2025	Arathy PS