# New Billing Project

## Number Selection Requirement API
## For Billing Team

**Revision History**

| Revision | Date | Description | Requester |
|----------|------|-------------|-----------|
| 1.0 | 26-Mar-2025 | List of API to get thenumber list from RMM tables for new billing system that are called by ICP API | Claudia Fung |

**CONTENTS**

# Objectives

This document is to request billing team to develop some APIs functions for extraction numbers from RMM tables directly based on different number type.  Those APIs are called only by ICP API to provide the final numbering list to the north bound caller (e.g. OM Change Form, FES etc)

**List of Requirements Raised by OM team**

- get 1638 IDD number
    - Refer to the below API function: get1638Number
- get FibreBB number (for SA/OM standalone but not online)
    - Refer to the below API function: getDummyNumber
- get CU China number (return 1 number)
    - Refer to the below API function: getChinaNumber
- get the number list of China number (for OM Change Form)
    - Refer to the below API function: ChinaNumberList
- get dummy number – CPS
    - Refer to the below API function: getDummyNumber
- get dummy number – PIE
    - Refer to the below API function: getDummyNumber
- get HomeTel dummy SIM
    - Refer to the below API function: getDummySim

Note:

- The existing program details can be referenced to the "Get Number List.doc" and "Get_Next_Resources.fnc" provided by OM team
- Tables mentioned to the requirement specifications are read from OM database

## 1. Get Available 1638 Number

**Purpose**

To get available number from RMM for 1638

**Production Endpoints:** TBC

| System | Endpoint (UAT) |
|--------|----------------|
| UAT | TBC |
| TYK | TBC |

*API:* /get1638Number

*Type:* REST

*Method:* POST

**Add Key to HTTP Request Header for TYK API:**

| Name | Auth_ID |
|------|---------|
| Value | TBC |

**Request Parameters**

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| typeId | String | Type Id for 1638 numbering | 3 | Mandatory |
| typeId2 | String | Another type ID for 1638 numbering provided by caller | | Optional |
| phlr | String | pHLR | 0 | Mandatory |
| vasType | String | Vas type for 1638 numbering | ISR | Mandatory |
| numStart | String | Number range for search | 1440000000 | Mandatory |
| numEnd | String | Number range for search | 1449999999 | Mandatory |

**Sample Request**

```
{
        "typeId": "3",
        "typeId2": "37",
        "phlr": "0",
        "vasType": "ISR",
        "numStart": "1442000000",
        "numEnd": "1449999999"
}

{
        "typeId": "3",
        "typeId2": "",
        "phlr": "0",
        "vasType": "ISR",
        "numStart": "42000000",
        "numEnd": "'49999999'"
}
```

## Response Parameters

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| resultCode | Number | Return code | 0 -Success<br>-1 - Failure | |
| resultMsg | String | Error message | | |
| msisdn | String | Return 1638 number | | |

Success response

```
{
        "resultCode": 0,
        "resultMsg": "",
        "msisdn": "42000010"
}
```

Fail response

```
{
        "resultCode": -1,
        "resultMsg": "Available number cannot be found"
}
```

## Programming Logic

sTypeId = input typeId
sTypeId2 = input typeId2
sPhlr = input phlr
sVasType = input vasType
sNumStart = input numStart
sNumEnd = input numEnd

Open cursor
       SELECT /*+ index(a L_RMM_MSISDN_AK2) */ msisdn
       FROM l_rmm_msisdn a
       WHERE sequence_no = (SELECT /*+ index(b PKL_RMM_MSISDN) */ MAX (sequence_no) FROM l_rmm_msisdn c
              WHERE a.msisdn = c.msisdn)
              AND msisdn BETWEEN **sNumStart** AND **sNumEnd**
              AND status = 'R' AND last_switch_on_date IS NULL
              AND (type_id = **sTypeId** OR (type_id = **sTypeId2** AND **sTypeId2** is not null))
              AND phlr = **sPhlr**
              AND vas_type = **sVasType** AND live_indicator = 'N'
              AND ROWNUM = 1
Loop
       Fetch record into sMsisdn

       // mark to hold for the available 1638 number by the caller
       Update l_rmm_msisdn a
              Set live_indicator = **'P',** last_updated_date_time = SYSDATE
       WHERE msisdn = **sMsisdn**
              AND sequence_no = ( SELECT MAX(Sequence_no) FROM l_rmm_msisdn
                     WHERE msisdn = a.msisdn)
              AND live_indicator = **'N'**
       COMMIT

       set response parameter {
              "resultCode": 0,
              "msisdn": sMsisdn
       }
       Return
End Loop

set response parameter {
       "resultCode": -1,
       "resultMsg": 'Available number cannot be found'
}
Return

## 2. Get Available Dummy Number

### Purpose

To get available dummy number from RMM for different number type including CPSM, RIN2, PIE, FBBR

*Production Endpoints:* TBC

| System | Endpoint (UAT) |
|--------|----------------|
| UAT | TBC |
| TYK | TBC |

*API:* /getDummyNumber

*Type:* REST

*Method:* POST

### Add Key to HTTP Request Header for TYK API:

| Name | Auth_ID |
|------|---------|
| Value | TBC |

### Request Parameters

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| typeId | String | Type Id for dummy numbering | 17 | Mandatory |
| phlr | String | pHLR | 0 | Mandatory |
| vasType | String | Vas type for dummy numbering | CPSM | Mandatory |

### Sample Request

```
// Get number for PIE
{
        "typeId": "19",
        "phlr": "0",
        "vasType": "PIE"
}
// Get number for CPS
{
        "typeId": "17",
        "phlr": "0",
        "vasType": "CPSM"
}
// Get number for FibreBB No for Sales Activation
{
        "typeId": "28",
        "phlr": "0",
        "vasType": "FBBR"
}
```

**Response Parameters**

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| resultCode | Number | Return code | 0 -Success<br>-1 - Failure | |
| resultMsg | String | Error message | | |
| msisdn | String | Return dummy number | | |

Success response

```
{
        "resultCode": 0,
        "resultMsg": "",
        "msisdn": "42000010"
}
```

Fail response

```
{
        "resultCode": -1,
        "resultMsg": "Available number cannot be found"
}
```

## Programming Logic

```
sTypeId = input typeId
sPhlr = input phlr
sVasType = input vasType

Open cursor
        SELECT /*+ index(a L_RMM_MSISDN_AK2) */ msisdn
        FROM l_rmm_msisdn a, l_rmm_msisdn_type b
        WHERE sequence_no = (SELECT /*+ index(b PKL_RMM_MSISDN) */ MAX (sequence_no) FROM l_rmm_msisdn c
                                WHERE a.msisdn = c.msisdn)
                                AND a.type_id = b.type_id
                                AND b.type_code = sVasType
                                AND msisdn like b.prefix || '%'
                                AND length(msisdn) = type_length
                                AND status = 'R' AND last_switch_on_date IS NULL
                                AND a.type_id = sTypeId AND phlr = sPhlr
                                AND vas_type = sVasType AND live_indicator = 'N'
                                AND ROWNUM = 1
Loop
        Fetch record into sMsisdn

        // mark to hold for the available dummy number by the caller
        Update l_rmm_msisdn a
                Set live_indicator = 'P', last_updated_date_time = SYSDATE
        WHERE msisdn = sMsisdn
                AND sequence_no = ( SELECT MAX(Sequence_no) FROM l_rmm_msisdn
                                WHERE msisdn = a.msisdn)
                AND live_indicator = 'N'
        COMMIT

        set response parameter {
                "resultCode": 0,
                "msisdn": sMsisdn
        }
        Return
End Loop

set response parameter {
        "resultCode": -1,
        "resultMsg": 'Available number cannot be found'
}
Return
```

## 3. Get Available China Number

**Purpose**

To get available number from RMM for China Numbe r(U1C2N)

***Production Endpoints:*** TBC

| System | Endpoint (UAT) |
|--------|----------------|
| UAT | TBC |
| TYK | TBC |

***API:*** /getChinaNumber

***Type:*** REST

***Method:*** POST

**Add Key to HTTP Request Header for TYK API:**

| | |
|------|--------|
| Name | Auth_ID |
| Value | TBC |

**Request Parameters**

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| typeId | String | Type Id for China numbering | 4 | Mandatory |
| phlr | String | pHLR | 0 | Mandatory |
| vasType | String | Vas type for China numbering | GD_CU<br>SH_CU<br>BJ_CU | Mandatory |
| searchMth | String | Search Method | 1 or 2 | Mandatory |

## Sample Request

```
// Get number for China Number - Guangdong
{
        "typeId": "4",
        "phlr": "0",
        "vasType": "GD_CU",
        "searchMth": 1
}

// Get number for China Number - Shanghai
{
        "typeId": "4",
        "phlr": "0",
        "vasType": "SH_CU",
        "searchMth": 1
}

// Get number for China Number - Beijing
{
        "typeId": "4",
        "phlr": "0",
        "vasType": "BJ_CU",
        "searchMth": 2
}
```

## Response Parameters

| Parameter | Data Type | Description | Possible Value | Remarks |
|---|---|---|---|---|
| resultCode | Number | Return code | 0 -Success<br>-1 - Failure | |
| resultMsg | String | Error message | | |
| msisdn | String | Return china number | | |

## Success response

```
{
        "resultCode": 0,
        "resultMsg": "",
        "msisdn": "42000010"
}
```

## Fail response

```
{
        "resultCode": -1,
        "resultMsg": "Available number cannot be found"
}
```

## Programming Logic

```
sTypeId = input typeId
sPhlr = input phlr
sVasType = input vasType

Open cursor

        If searchMth = 1
                SELECT /*+ index(a L_RMM_MSISDN_AK2) */ msisdn, NULL
                FROM l_rmm_msisdn
                WHERE sequence_no =
                        (SELECT /*+ index(b PKL_RMM_MSISDN) */ MAX (sequence_no) FROM l_rmm_msisdn c
                                WHERE a.msisdn = c.msisdn)
                                AND msisdn like 'U%'
                                AND status = 'R'
                                AND a.type_id = sTypeId AND phlr = sPhlr
                                AND vas_type = sVasType AND live_indicator = 'N'
                                AND ROWNUM = 1
        Else
                // PS: L_RMM_MSISDN_VW3 is synonym of OM database.
                SELECT /*+ INDEX(v l_rmm_msisdn_ak2) */ msisdn, last_switch_on_date
                FROM l_rmm_msisdn_vw3 v
                WHERE type_id = sTypeId AND status = 'R'
                        AND (vas_type = sVasType OR sVasType IS NULL) AND phlr = sPhlr;

        End If

Loop
        Fetch record into sMsisdn, sLastSwitchOnDate

        // mark to hold for the available dummy number by the caller
        Update l_rmm_msisdn a
                Set live_indicator = 'P', last_updated_date_time = SYSDATE
        WHERE msisdn = sMsisdn
                AND sequence_no = ( SELECT MAX(Sequence_no) FROM l_rmm_msisdn
                                WHERE msisdn = a.msisdn)
                AND live_indicator = 'N'
        COMMIT

        set response parameter {
                "resultCode": 0,
                "msisdn": sMsisdn
        }
        Return
End Loop

set response parameter {
        "resultCode": -1,
        "resultMsg": 'Available number cannot be found'
}
Return
```

## 4. Get HomeTel Dummy SIM

### Purpose

To get available dummy Sim from RMM for HomeTel

***Production Endpoints:*** TBC

| System | Endpoint (UAT) |
|--------|----------------|
| UAT | TBC |
| TYK | TBC |

***API:***          /getDummySim

***Type:***         REST

***Method:***      POST

### Add Key to HTTP Request Header for TYK API:

| Name | Auth_ID |
|------|---------|
| Value | TBC |

### Request Parameters

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| typeId | String | Type Id for HomeTel | 4 | Mandatory |
| phlr | String | pHLR | 4 | Mandatory |
| vasType | String | Vas type for HomeTel | FBTEL | Mandatory |

### Sample Request

```
// Get dummy sim for HomeTel
{
        "typeId": "4",
        "phlr": "4",
        "vasType": "FBTEL"
}
```

## Response Parameters

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| resultCode | Number | Return code | 0 -Success<br>-1 - Failure | |
| resultMsg | String | Error message | | |
| imsi | String | Return IMSI number | | |
| sim | String | Return SIM number | | |

## Success response

```
{
        "resultCode": 0,
        "resultMsg": "",
        "imsi": "454065210008819",
        "sim": "89852061209230538192"


}
```

## Fail response

```
{
        "resultCode": -1,
        "resultMsg": "Available sim cannot be found"
}
```

## Programming Logic

sTypeId = input typeId
sPhlr = input phlr
sVasType = input vasType

Open cursor

```
    SELECT imsi_num FROM (
        SELECT /*+ INDEX(i L_RMM_IMSI_AK2) */ imsi as imsi_num,
                (select count(*) from l_rmm_imsi rr where rr.sequence_no =
                        (SELECT /*+ INDEX(b PKL_RMM_IMSI) */ MAX(b.sequence_no) FROM l_rmm_imsi b
                            WHERE b.imsi = i.imsi AND cancel_flag = 'N') and rr.imsi = i.imsi
                                    and rr.last_switch_on_date is null and rr.status = 'R'
                                    and rr.sim_type_id = sTypeId
                                    and (rr.vas_type = sVasType OR sVasType IS NULL)
                                    and rr.phlr = sPhlr
                    ) as sequence_num
        FROM l_rmm_imsi i
        WHERE sim_type_id = sTypeId
                AND (vas_type = sVasType OR sVasType IS NULL)
                AND status = 'R' AND phlr = sPhlr
    ) where sequence_num > 0;
```

Loop

```
        Fetch record into sImsi

        Open cursor
                Select sim from l_rmm_imsi
                Where imsi = sImsi
                Order by sequence_no desc
        Loop
                Fetch record into sSim
                Exit Loop
        End Loop

        set response parameter {
                "resultCode": 0,
                "imsi": sImsi,
                "sim": sSim
        }
        Return
End Loop

set response parameter {
        "resultCode": -1,
        "resultMsg": 'Available imsi and sim cannot be found'
}
Return
```

## *5.* *Get China Number List*

**Purpose**

To get the available China numbering list from RMM

***Production Endpoints:*** TBC

| System | Endpoint (UAT) |
|--------|----------------|
| UAT | TBC |
| TYK | TBC |

***API:***             /chinaNumberList

***Type:***            REST

***Method:***       POST

**Add Key to HTTP Request Header for TYK API:**

| Name | Auth_ID |
|------|---------|
| Value | TBC |

**Request Parameters**

| Parameter | Data Type | Description | Possible Value | Remarks |
|-----------|-----------|-------------|----------------|---------|
| numRec | String | Number of records to be returned | 4 | Mandatory |

**Sample Request**

```
// Request the available China number records to be returned
{
        "numRec": 4
}
```

## Response Parameters

| Parameter | Data Type | Description | Possible Value | Remarks |
|---|---|---|---|---|
| resultCode | Number | Return code | 0 -Success<br>-1 - Failure | |
| resultMsg | String | Error message | | |
| numberList | Json Array | | | |
| numberList -> msisdn | String | Available China number | | |

Success response

```
{
        "resultCode": 0,
        "resultMsg": "",
        "numberList": {
                "msisdn": "91234567"
        }
}
```

Fail response

```
{
        "resultCode": -1,
        "resultMsg": "Available sim cannot be found"
}
```

## Programming Logic

nRec = input numRec

cnt = 0;
Open cursor
    select replace(a.msisdn,'U','') as msisdn from l_RMM_MSISDN a,
    (select msisdn, max(sequence_no) sequence_no from l_RMM_MSISDN group by msisdn) b
    where a.msisdn like 'U%'
       and a.sequence_no = b.sequence_no and a.msisdn = b.msisdn
       and (status = 'R' AND (live_indicator = 'N') and vas_type <> 'CHINA')
    order by dbms_random.value


Loop
       Fetch record into sMsisdn

       If cnt < nRec
           set response parameter
           {
               "numberList": {
                   "msisdn": sMsisdn
               }
           }
           Cnt++
       Else
           Exit loop
       end if
End Loop

If cnt = 0
       set response parameter {
           "resultCode": -1,
           "resultMsg": 'Available china number cannot be found'
       }
End if
Return