

How to Create Stock Locators By Using API

1. Create Table

Create a new table by using following script

```
CREATE TABLE DEV_DATA
(
  SR_NO          NUMBER,
  INV_ORG_ID     NUMBER,
  INVENTORY_ORGANIZATION NUMBER,
  SUB_INVENTORY CHAR(3 BYTE),
  WAREHOUSE      CHAR(3 BYTE),
  FLOR           CHAR(3 BYTE),
  ROWN           CHAR(3 BYTE),
  RACK           CHAR(3 BYTE),
  BIN            CHAR(3 BYTE),
  BOX            CHAR(3 BYTE),
  SUPPLIER       CHAR(4 BYTE),
  SPARE1         CHAR(3 BYTE),
  SPARE2         CHAR(3 BYTE),
  CODE           CHAR(45 BYTE),
  CREATION_DATE  DATE,
  STATUS        CHAR(1 BYTE)
)
```

2. Upload Data in Table by Using SQLLDR

Control File Syntax

```
LOAD DATA
  INFILE 'Data.csv'
  APPEND INTO TABLE Dev_Data
  REPLACE FIELDS TERMINATED BY ','
  TRAILING NULLCOLS
(
  SR_NO "trim(:SR_NO)",
  INV_ORG_ID "trim(:INV_ORG_ID)",
  INVENTORY_ORGANIZATION "trim(:INVENTORY_ORGANIZATION)",
  SUB_INVENTORY "trim(:SUB_INVENTORY)",
  WAREHOUSE "trim(:WAREHOUSE)",
  FLOR "trim(:FLOR)",
  ROWN "trim(:ROWN)",
  RACK "trim(:RACK)",
  BIN "trim(:BIN)",
  BOX "trim(:BOX)",
  SUPPLIER "trim(:SUPPLIER)",
  SPARE1 "trim(:SPARE1)",
  SPARE2 "trim(:SPARE2)",
  CODE "trim(:CODE)",
  CREATION_DATE "trim(:CREATION_DATE)"
)
```

Verify Data is uploaded successfully by using following two queries

```
Select count(*) from dev_data  
Select * from dev_data
```

3. Uploading Script (API)

Connect by Using APPS Credentials to Database

Run the following scripts this will create locator

```
DECLARE  
l_msg_data VARCHAR2(100);  
l_msg_count NUMBER;  
l_return_status VARCHAR2(1);  
l_locator_id NUMBER;  
l_locator_exists VARCHAR2(1);  
l_org_id NUMBER := 110; /*Organization_id */  
l_organization_code VARCHAR2(10) := 'V01'; /*Organization_Code */  
l_sub_code VARCHAR2(10) ; /*Variable for Subinventory*/  
l_concatenated_segments VARCHAR2(100); /*Variable for Locator Segment*/  
l_user_id NUMBER := 1262; /* User ID From FND_users Table */  
l_resp_id NUMBER := 20634; /*Responsibility Id*/  
l_resp_appl_id NUMBER := 401; /* Responsibility Application id */  
  
/*Fetch data into a cursor for creation of Locator*/  
cursor c_loc is select sr_no,sub_inventory,code  
from dev_data;  
  
v_loc c_loc%rowtype;  
  
BEGIN  
/*  
* APPS_INITIALIZE Required because indirectly use profile options  
*/  
FND_GLOBAL.APPS_INITIALIZE(l_user_id, l_resp_id,l_resp_appl_id);  
/*  
Open Cursor  
*/  
OPEN c_loc;  
LOOP  
FETCH c_loc INTO v_loc;  
EXIT WHEN c_loc%NOTFOUND;  
  
l_concatenated_segments := v_loc.code;  
l_sub_code := v_loc.sub_inventory;  
  
FND_MSG_PUB.INITIALIZE;  
  
/*  
* First create physical locators first and use the returned value of x_inventory_location_  
* in the next call  
*  
*/  
DBMS_OUTPUT.PUT_LINE('Trying to create ' || l_concatenated_segments);
```

```

INV_LOC_WMS_PUB.CREATE_LOCATOR(
x_return_status => l_return_status,
x_msg_count => l_msg_count,
x_msg_data => l_msg_data,
x_inventory_location_id => l_locator_id,
x_locator_exists => l_locator_exists,
p_organization_id => l_org_id,
p_organization_code => l_organization_code,
p_concatenated_segments => l_concatenated_segments,
p_description => 'Locator Created By API', /*You can also use here description of Your Locator
Combination*/
p_inventory_location_type => 3, -- Storage locator
p_picking_order => NULL,
p_location_maximum_units => NULL,
p_subinventory_code => l_sub_code, /*Subinventory Code */
p_location_weight_uom_code => NULL,
p_max_weight => NULL,
p_volume_uom_code => NULL,
p_max_cubic_area => NULL,
p_x_coordinate => NULL,
p_y_coordinate => NULL,
p_z_coordinate => NULL,
p_physical_location_id => NULL,
p_pick_uom_code => NULL,
p_dimension_uom_code => NULL,
p_length => NULL,
p_width => NULL,
p_height => NULL,
p_status_id => 1, -- Default status 'Active'
p_dropping_order => NULL
);
DBMS_OUTPUT.PUT_LINE('Return Status ' || l_return_status);
IF l_return_status IN ('E', 'U') THEN
DBMS_OUTPUT.PUT_LINE('# of Errors ' || l_msg_count);
update dev_data set status = l_return_status
where sr_no = v_loc.sr_no ;
commit;
IF l_msg_count = 1 THEN
DBMS_OUTPUT.PUT_LINE('Error ' || l_msg_data);
ELSE
FOR i IN 1..l_msg_count LOOP
DBMS_OUTPUT.PUT_LINE('Error ' || FND_MSG_PUB.GET(i, 'F'));
END LOOP;
END IF;
ELSE
DBMS_OUTPUT.PUT_LINE('Locator Id is ' || l_locator_id);
END IF;
/*
* Now try to create the Project/Task Locator
*
* p_concatenated_segments should always be the value that you see in the desktop forms
* not internal ids
* If the desktop form shows 'Stor.1.1.Pacific Pumps.' -- the same should be passed to
* the API
*

```

*/

```
DBMS_OUTPUT.PUT_LINE('Trying to create ' || I_concatenated_segments);
INV_LOC_WMS_PUB.CREATE_LOCATOR(
x_return_status => I_return_status,
x_msg_count => I_msg_count,
x_msg_data => I_msg_data,
x_inventory_location_id => I_locator_id,
x_locator_exists => I_locator_exists,
p_organization_id => I_org_id,
p_organization_code => I_organization_code,
p_concatenated_segments => I_concatenated_segments,
p_description => 'Locator Created By API',
p_inventory_location_type => 3, -- Storage locator
p_picking_order => NULL,
p_location_maximum_units => NULL,
p_subinventory_code => I_sub_code,
p_location_weight_uom_code => NULL,
p_max_weight => NULL,
p_volume_uom_code => NULL,
p_max_cubic_area => NULL,
p_x_coordinate => NULL,
p_y_coordinate => NULL,
p_z_coordinate => NULL,
p_physical_location_id => I_locator_id, -- Value got from call above
p_pick_uom_code => NULL,
p_dimension_uom_code => NULL,
p_length => NULL,
p_width => NULL,
p_height => NULL,
p_status_id => 1, -- Default status 'Active'
p_dropping_order => NULL
);
```

```
DBMS_OUTPUT.PUT_LINE('Return Status ' || I_return_status);
IF I_return_status IN ('E', 'U') THEN
DBMS_OUTPUT.PUT_LINE('# of Errors ' || I_msg_count);
update dev_data set status = I_return_status
where sr_no = v_loc.sr_no ;
commit;
IF I_msg_count = 1 THEN
DBMS_OUTPUT.PUT_LINE('Error ' || I_msg_data);
ELSE
FOR i IN 1..I_msg_count LOOP
DBMS_OUTPUT.PUT_LINE('Error ' || FND_MSG_PUB.GET(i, 'F'));
END LOOP;
END IF;
ELSE
DBMS_OUTPUT.PUT_LINE('Locator Id is ' || I_locator_id);
END IF;
END LOOP;
END LOOP;
CLOSE c_loc;
END;
/
```

4. Check Created values

Select * from mtl_item_locations where description = 'Locator Created by API' order by creation_date desc

Reference: - [Metalink.oracle.com](http://metalink.oracle.com)