

Success Factor API

```
USE ROLE ACCOUNTADMIN;
USE DATABASE SANDBOX;
USE SCHEMA SOURCING;

CREATE OR REPLACE NETWORK RULE SUCCESS_FACTOR_API_NETWORK_RULE
  MODE = EGRESS
  TYPE = HOST_PORT
  VALUE_LIST = ('api55.sapsf.eu');

CREATE OR REPLACE EXTERNAL ACCESS INTEGRATION SUCCESS_FACTOR_API_INTEGRATION_EXTERNAL_ACCESS
  ALLOWED_NETWORK_RULES = (SUCCESS_FACTOR_API_NETWORK_RULE)
  ALLOWED_AUTHENTICATION_SECRETS = ALL
  ENABLED = true;

CREATE OR REPLACE SECRET SUCCESS_FACTOR_API_TOKEN_SECRET
  TYPE = GENERIC_STRING
  SECRET_STRING = '<SECRET-KEY>';

CREATE OR REPLACE PROCEDURE PS_GET_FROM_SUCCESS_FACTOR_API (URL STRING, table_name STRING)
  RETURNS STRING
  LANGUAGE PYTHON
  RUNTIME_VERSION = 3.10
  HANDLER = 'get_data'
  EXTERNAL_ACCESS_INTEGRATIONS = (SUCCESS_FACTOR_API_INTEGRATION_EXTERNAL_ACCESS)
  PACKAGES = ('snowflake-snowpark-python','requests')
  SECRETS = ('cred' = SUCCESS_FACTOR_API_TOKEN_SECRET)
  EXECUTE AS CALLER
  AS
  $$
import _snowflake
import requests
import json

def get_token():

    return _snowflake.get_generic_secret_string("cred")

def get_headers():
    headers = {
        'Accept': 'application/json',
        'Authorization': 'Basic {}'.format(get_token())
    }

    return headers

def get_nextPageURL(last_result):
    return last_result.get('d').get('__next', None)

def get_data(session, URL, table_name):

    headers = get_headers()
    current_url = URL
    session.sql(f"CREATE OR REPLACE TABLE {table_name} (page_id NUMBER, json_data VARIANT);").collect()
    last_result = None;
    page = 0
    while current_url:
        try:
            page += 1

            last_result = requests.get(url=current_url, headers=headers).json()

            if not last_result:
                break

            # Use a parameterized query with placeholders
            query = "INSERT INTO {0} (page_id, json_data) SELECT :1, PARSE_JSON(:2)".format(table_name)
```

```

        session.sql(query, (page, json.dumps(last_result))).collect()

        current_url = get_nextPageURL(last_result)

    except Exception as e:
        return f"TABLE NAME:\t{table_name}\nURL:\t{current_url}\nCURRENT PAGE:\t{page}\nERROR:\t{e}\nLAST RESULT:\t{last_result}"

    return f"TABLE NAME:\t{table_name}\nURL:\t{URL}\nTOTAL PAGES:\t{page}\nLATEST RESULT:\t{last_result}"

$$;

GRANT USAGE ON PROCEDURE SANDBOX.SOURCING.PS_GET_FROM_SUCCESS_FACTOR_API(VARCHAR, VARCHAR) TO ROLE
HRDP_NP_DOMAIN_ADMIN;

GRANT USAGE ON SCHEMA SANDBOX.SOURCING TO ROLE HRDP_NP_DOMAIN_ADMIN;
GRANT USAGE ON SCHEMA SANDBOX.TESTING TO ROLE HRDP_NP_DOMAIN_ADMIN;
GRANT USAGE ON DATABASE SANDBOX TO ROLE HRDP_NP_DOMAIN_ADMIN;

use role HRDP_NP_DOMAIN_ADMIN;
---example

CALL SANDBOX.SOURCING.PS_GET_FROM_SUCCESS_FACTOR_API('https://api55.sapsf.eu/odata/v2/Country?
paging=cursor&customPageSize=500&', 'SANDBOX.TESTING.SF_API_COUNTRY');

SELECT JD.VALUE:code::string as Country_Code, JD.VALUE:currency::string AS Country_Currency, JD.VALUE
FROM SANDBOX.TESTING.SF_API_COUNTRY,
LATERAL FLATTEN(JSON_DATA, 'd.results') AS JD ;

```