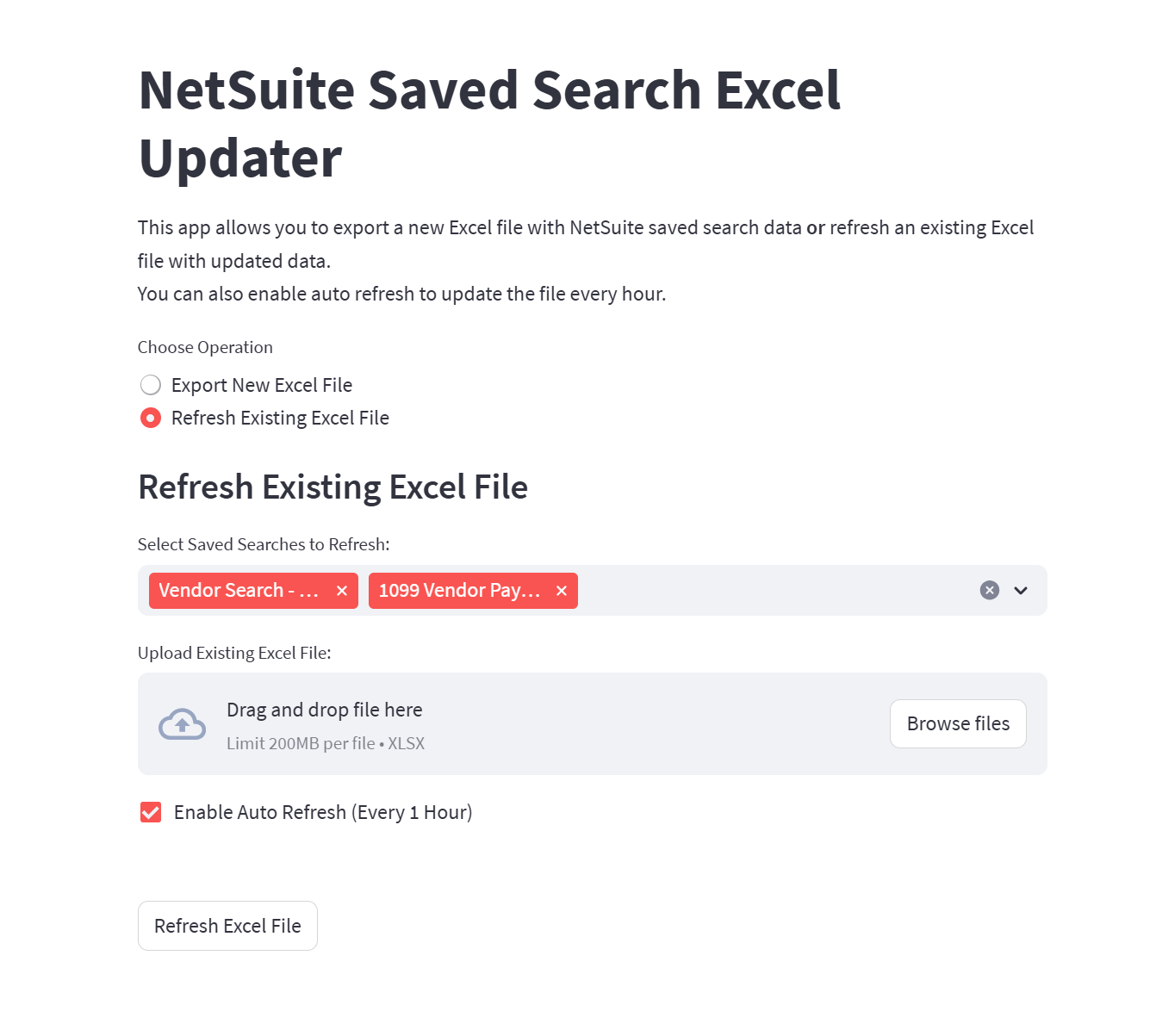
# **NetSuite Saved Searches - Excel Integration**

This repository contains scripts for integrating NetSuite ERP saved searches with a Streamlit web application using a NetSuite RESTlet.

## **Overview**

This project provides scripts for integrating NetSuite ERP saved searches with a Streamlit web application using a NetSuite RESTlet.



## **Project Structure**

This repository includes the following scripts:

* restlet\_savedsearch\_final.js: A NetSuite RESTlet to fetch saved search data.
* ns\_ss\_refresh.py: A Streamlit application script to interact with the RESTlet and manage Excel exports.

## **Requirements**

* NetSuite Account with RESTlet script deployment
* Streamlit installed locally or hosted deployment

Python dependencies:  
streamlit

pandas

openpyxl

requests

requests\_oauthlib

## **Installation**

### **Setup Streamlit App**

1. Clone this repository:

git clone <repository-url>

cd <repository-folder>

1. Install Python dependencies:

pip install streamlit pandas openpyxl requests requests\_oauthlib

## **Configuration**

Create a secrets.toml file in the .streamlit directory with the following variables:

NS\_ACCOUNT\_ID = "your\_account\_id"

NS\_CONSUMER\_KEY = "your\_consumer\_key"

NS\_CONSUMER\_SECRET = "your\_consumer\_secret"

NS\_TOKEN\_ID = "your\_token\_id"

NS\_TOKEN\_SECRET = "your\_token\_secret"

NS\_RESTLET\_URL = "your\_restlet\_url"

NS\_DEPLOY\_ID = "your\_deploy\_id"

NS\_SCRIPT\_ID = "your\_script\_id"

## **Deployment in NetSuite**

* Navigate to Customization → Scripting → Scripts → New
* Upload restlet\_savedsearch\_final.js
* Create a RESTlet script deployment:
  + Set the deployment to 'Released'.
  + Record the RESTlet URL, script ID, and deployment ID for use in Streamlit configuration.

## **How It Works**

### **Mode Selection**

The app allows users to choose between two modes:

* **Export New Excel File**
* **Refresh Existing Excel File**

### **Export New Excel File**

1. Retrieves saved searches from NetSuite.
2. The user selects which saved searches to include.
3. The app fetches data via the RESTlet.
4. Data from each search is written into individual sheets.
5. Provides a download button for the user to retrieve the generated Excel file.

### **Refresh Existing Excel File**

1. The user uploads an existing Excel file.
2. Selects saved searches to refresh.
3. The app retrieves the updated data from NetSuite.
4. Updates the respective sheets in the Excel file.
5. Provides a download button to obtain the refreshed Excel file.

## **Auto-Refresh**

Supports periodic auto-refresh to maintain updated data without manual intervention.

For further assistance or reporting issues, please open an issue on GitHub.

***Reach out to support@dataants.org if you need assistance deploying this. We also have a paid desktop app with a one click install option***