

Deployment Guide - Klaviyo Flow Email HTML Extractor

This guide provides step-by-step instructions for deploying your Klaviyo Flow Email HTML Extractor application through GitHub and Streamlit.

Prerequisites

Before you begin, ensure you have:

1. A GitHub account
2. A Streamlit account (free tier is sufficient)
3. Git installed on your local machine

Step 1: Prepare Your Repository Structure

Ensure your project has the following structure:

```
klaviyo-flow-email-extractor/  
|  
├─ app.py                # Main Streamlit application  
├─ requirements.txt      # Python dependencies  
├─ README.md            # Project documentation  
|  
├─ .streamlit/          # Streamlit configuration  
|   └─ config.toml      # Streamlit theme and settings  
|  
├─ utils/               # Utility functions  
|   └─ __init__.py      # Make utils a proper package  
|       └─ klaviyo_api.py # Klaviyo API interaction functions  
|           └─ html_utils.py # HTML processing utilities
```

Make sure to create an empty `__init__.py` file in the utils directory to make it a proper Python package.

Step 2: Set Up a GitHub Repository

1. Create a new repository on GitHub:
 - Go to <https://github.com/new>
 - Name it `klaviyo-flow-email-extractor` (or your preferred name)
 - Choose whether to make it public or private

- Click "Create repository"

2. Initialize your local repository and push to GitHub:

```
bash
```

```
# Navigate to your project directory
```

```
cd path/to/klaviyo-flow-email-extractor
```

```
# Initialize Git repository
```

```
git init
```

```
# Add all files to the staging area
```

```
git add .
```

```
# Commit the files
```

```
git commit -m "Initial commit"
```

```
# Link to your remote repository (replace YOUR_USERNAME with your GitHub username)
```

```
git remote add origin https://github.com/YOUR_USERNAME/klaviyo-flow-email-extractor.git
```

```
# Push to GitHub
```

```
git branch -M main
```

```
git push -u origin main
```

Step 3: Deploy to Streamlit Cloud

1. Sign in to Streamlit Cloud:

- Go to <https://streamlit.io/cloud>
- Sign in with your Streamlit account (or create one)

2. Deploy your app:

- Click on "New app" button
- Connect your GitHub account if you haven't already
- Select your repository ((klaviyo-flow-email-extractor))
- Select the branch (usually (main))
- Set the main file path to (app.py)
- Click "Deploy"

3. Wait for the deployment to complete. Streamlit will install dependencies and start your app.

Step 4: Configure Advanced Settings (Optional)

After your app is deployed, you can configure advanced settings:

1. Resource limits:
 - In your app settings, you can adjust the memory and CPU limits
 - For this app, the default limits should be sufficient
2. Secrets management:
 - Go to "Advanced settings" > "Secrets"
 - Here you can add default values for sensitive information like API keys
 - For example:

```
[klaviyo]  
default_api_key = "your-default-key"
```

Step 5: Updating Your App

When you make changes to your code:

1. Commit and push changes to GitHub:

```
bash  
  
git add .  
git commit -m "Your update message"  
git push
```

2. Streamlit will automatically detect changes and rebuild your app.

Troubleshooting Common Issues

Dependencies Not Installing

If dependencies fail to install, check your `requirements.txt` file for version conflicts. You may need to specify compatible versions.

Import Errors

If you see import errors related to the `utils` module, ensure:

1. The `__init__.py` file exists in the utils directory
2. Your imports use the correct path
3. The directory structure matches what's expected

API Rate Limiting

Klaviyo has rate limits that may affect your app. Consider implementing:

1. Caching responses
2. Adding delay between requests
3. Batch processing for bulk operations

Monitoring and Maintenance

- Regularly check your app logs in Streamlit Cloud to identify issues
- Update dependencies periodically to ensure security patches are applied
- Monitor your app's performance and adjust resource limits if necessary

Best Practices for Production Use

1. Use a more robust secret management solution for production
2. Implement proper error handling and logging
3. Add monitoring for application health
4. Consider adding user authentication for sensitive operations
5. Regularly backup any stored data

By following this guide, your Klaviyo Flow Email HTML Extractor should be successfully deployed and accessible via Streamlit Cloud!