Address Scrape Tool

Table of Contents

[Introduction 2](#_Toc133353500)

[About the Tool 2](#_Toc133353501)

[Installation and Prerequisites 2](#_Toc133353502)

[Step 1: Install Node.js 2](#_Toc133353503)

[Step 2: Clone the Repository 3](#_Toc133353504)

[Step 3: Install Dependencies 3](#_Toc133353505)

[Running the tool 3](#_Toc133353506)

[Step 1: Configure the Tool 4](#_Toc133353507)

[Step 2: Place the Input File in the `files` Folder 4](#_Toc133353508)

[Step 3: Run the Tool 6](#_Toc133353509)

[1: Open the Terminal 6](#_Toc133353510)

[2: Navigate to the Repository Directory 7](#_Toc133353511)

[3: Choose a County to Scrape 7](#_Toc133353512)

[5: Wait for the Script to Finish 8](#_Toc133353513)

# Introduction

This is a software documentation for a web scraping tool that is designed to extract and format addresses for mass mailings. The tool is highly efficient, capable of processing large volumes of data accurately and quickly. This documentation provides comprehensive information on the purpose, features, dependencies, configuration, and usage of the tool.

# About the Tool

This web scraping tool is designed to automate the process of extracting and formatting addresses for mass mailings, saving time and increasing efficiency. It uses web scraping techniques to recognize and extract the necessary data fields from a public website's search engine, formats them according to pre-defined rules, and compiles them into a document suitable for mass mailings.

The tool is highly versatile and can handle large volumes of data, making it ideal for businesses and organizations that rely on mass mailings for marketing and outreach purposes. By automating the process of extracting and formatting addresses, this tool eliminates the need for manual data entry and formatting, saving time and reducing errors.

This documentation provides a detailed overview of the tool's features and functionality, including how to configure and run the tool, as well as troubleshooting tips for optimal performance. By providing clear and concise documentation, users will be able to use the tool effectively and efficiently, resulting in more streamlined business processes and improved outcomes.

# Installation and Prerequisites

To install the web scraping tool and its prerequisites, please follow the steps outlined below:

## Step 1: Install Node.js

Before installing the web scraping tool, you need to have Node.js installed on your computer. To install Node.js, follow these steps:

- Go to the Node.js website at <https://nodejs.org/>

- Download the version of Node.js that matches your operating system.

- Follow the installation wizard to install Node.js on your computer.

## Step 2: Clone the Repository

The web scraping tool is available on a public repository, and you can clone it to your computer by following these steps:

- Open a terminal window on your computer.

- Navigate to the directory where you want to clone the repository.

- Run the following command to clone the repository:

```

git clone <https://github.com/valeriezenn27/address-scrape-tool.git>

```

Text

Description automatically generated with medium confidence

## Step 3: Install Dependencies

The web scraping tool requires several dependencies to function correctly. To install these dependencies, follow these steps:

- Navigate to the directory where you cloned the repository.

- Run the following command to install the dependencies:

```

npm install

```

Graphical user interface, text

Description automatically generated

# Running the tool

## Step 1: Configure the Tool

The web scraping tool requires configuration before you can use it. Open the `config\{county}.json` file in a text editor and update the values for the `url`, `filePath`, and `outputPath` fields to match your use case.

* `url`: The URL of the website that the web scraping tool will search for addresses. This field should be updated to match the URL of the public website that contains the addresses you want to extract.
* `filePath`: The file path of the input file that contains the data for the county you want to scrape addresses for. This field should be updated to match the file path of the `.xlsx` input file for the county you want to scrape addresses for. For example, if you want to scrape addresses for Tarrant County, set the value of this field to `files/TARRANT COUNTY.xlsx`.
* `outputPath`: The file path of the output file that will be generated by the web scraping tool. This field should be updated to match the file path and name of the output file that you want to generate. By default, the output file will be saved to the `output` folder with a name that includes the current date and time stamp. The `{0}` and `{1}` placeholders in the file name will be replaced with the county name and current date/time, respectively.

Graphical user interface, application

Description automatically generated

\*Note: This is already configured so no need to do anything.

## Step 2: Place the Input File in the `files` Folder

Place the input file for the county you want to scrape data for in the `files` folder. The file should be named according to the county you want to scrape data for:

- `TRAVIS COUNTY.xlsx` for Travis County

- `DALLAS COUNTY.xlsx` for Dallas County

- `TARRANT COUNTY.xlsx` for Tarrant County

- `HARRIS COUNTY.xlsx` for Harris County

Please ensure that the file name is exactly as listed above and that it is in the `.xlsx` format. This input file contains the data that the web scraping tool will use to search for addresses on the public website.

Diagram

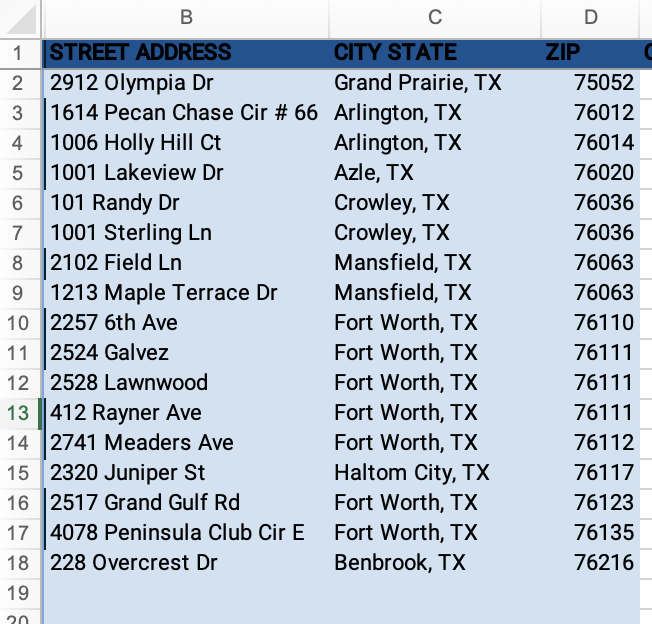
Description automatically generated with medium confidence

The input files for the web scraping tool should have the following columns:

- `STREET ADDRESS`: The street address of the location.

- `CITY STATE`: The city and state of the location.

- `ZIP`: The ZIP code of the location.



Please ensure that the input data is accurate and correctly formatted and spelled. This information will be used by the web scraping tool to search for addresses on the public website.

By providing accurate and correctly formatted input data, you can improve the accuracy and reliability of the extracted data, resulting in more streamlined business processes and improved outcomes.

## Step 3: Run the Tool

There are four separate scripts for scraping addresses for different counties. To run the script for a specific county, follow these steps:

### 1: Open the Terminal

### 2: Navigate to the Repository Directory

Navigate to the directory where you cloned the repository using the `cd` command. For example, if you cloned the repository to your `Documents` folder, you can navigate to it using the following command:

```

cd Documents/address-scrape-tool

```

### 3: Choose a County to Scrape

Choose the county you want to scrape addresses for. For example, if you want to scrape addresses for Tarrant County, run the following command:

```

npm start Tarrant

```

Graphical user interface, text, application

Description automatically generated

Similarly, replace `Tarrant` with the name of the county you want to scrape addresses for. The available options are `Dallas`, `Harris`, and `Travis`.

### 4: View the Logging Information

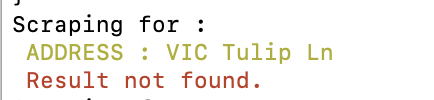
As the web scraping tool runs, it will display logging information on the console application. This logging information includes the scraped data and any errors that occur during the scraping process.

Text

Description automatically generated

The logging information is helpful for tracking the progress of the scraping process and identifying any issues that may arise. By monitoring the logging information, you can quickly identify and resolve any errors, ensuring that the scraping process runs smoothly and efficiently.

If you encounter any errors during the scraping process, the logging information can provide valuable insight into the cause of the error, allowing you to troubleshoot and fix the issue more quickly.



### 5: Wait for the Script to Finish

The script will automatically start scraping data from the specified county's website. Wait for the script to finish running, and it will generate a formatted CSV file with the extracted data.

The output file generated by the web scraping tool will be saved to the `output` folder using the `outputPath` specified in the `config.json` file.

Text

Description automatically generated

By default, the output file will be named with the current date and time stamp, along with the county name, and will be saved in CSV format. For example, if you scraped addresses for Tarrant County on April 25th, 2023, the output file would be named `tarrant\_2023-4-24\_13-22-32.csv` and would be saved in the `output` folder.

Please ensure that you check the `output` folder after running the web scraping tool to retrieve the output file containing the extracted addresses.

Congratulations! You have successfully run the web scraping script for a specific county. Repeat the process for other counties if needed.