Contents

[Vaxtor Web Scraper User Guide 1](#_Toc195262318)

[1. Overview 1](#_Toc195262319)

[2. Installation and Setup 1](#_Toc195262320)

[2.1 Prerequisites 1](#_Toc195262321)

[2.2 Software Installation 1](#_Toc195262322)

[2.3 AWS Server Setup 2](#_Toc195262323)

[2.4 Configuration 2](#_Toc195262324)

[3. Running the System 2](#_Toc195262325)

[3.1 Starting the Scraper 2](#_Toc195262326)

[3.2 Monitoring the System 2](#_Toc195262327)

[3.3 Viewing Results 2](#_Toc195262328)

[4. Troubleshooting 2](#_Toc195262329)

[4.1 Common Errors and Solutions 2](#_Toc195262330)

[4.2 System Recovery 3](#_Toc195262331)

[4.3 Camera Setup Issues 3](#_Toc195262332)

# Vaxtor Web Scraper User Guide

## 1. Overview

The Vaxtor Web Scraper is an automated system that:

* Continuously monitors a Vaxtor web interface for license plate data
* Captures and processes license plate images
* Performs plate recognition to identify vehicle make and model
* Sends the processed data to a remote server
* Saves data locally for record-keeping

The system runs on a 5-second interval by default and includes built-in error handling and retry mechanisms.

## 2. Installation and Setup

### 2.1 Prerequisites

* Python 3.x installed on your system
* Latest version of Google Chrome browser
* Git installed on your system
* AWS account with EC2 instance access
* Ethernet connection for camera setup

### 2.2 Software Installation

1. Clone the repository to your local system
2. Install required Python packages through this command: pip install -r requirements.txt

This will install:

* selenium (4.18.1)
* webdriver-manager (4.0.1)
* beautifulsoup4 (4.12.3)
* pandas (2.2.1)
* requests (2.31.0)

### 2.3 AWS Server Setup

1. Log in to AWS and navigate to EC2 Instances
2. Start the Vaxtor\_Server instance
3. Connect to the instance and execute the following commands:

* sudo su
* cd LATEST\_SERVER
* cd Vaxtor-Server
* git pull
* source venv/bin/activate
* python app.py

### 2.4 Configuration

Edit the following in config.py:

* VAXTOR\_URL: Update with your local Vaxtor system URL
* LOCAL\_ENDPOINT\_URL: Set to your AWS instance URL (http://XX.XX.XX.XX:5000)
* API\_TOKEN: Update if the PlateRecognizer token changes

## 3. Running the System

### 3.1 Starting the Scraper

1. Ensure the AWS server is running
2. Open a terminal in the project directory
3. Run the main script: python main.py

### 3.2 Monitoring the System

* The scraper will run continuously, checking for new data every 5 seconds
* Console output will show:
* Scraping attempts and results
* New data detection
* Plate recognition processing
* Data saving status

### 3.3 Viewing Results

Access the following URLs in your browser:

* http://XX.XX.XX.XX:5000/home/platerecognizer
* http://XX.XX.XX.XX:5000/platerecognizer-record-latest

## 4. Troubleshooting

### 4.1 Common Errors and Solutions

**Browser Initialization Issues**

Error: "Failed to start browser"

Solution:

* Ensure Chrome is installed and up to date
* Check if ChromeDriver is compatible with your Chrome version
* Verify no other Chrome instances are running

**Connection Issues**

Error: "Failed to refresh data" or "Connection error"

Solution:

* Verify network connectivity
* Check if the Vaxtor system is accessible
* Ensure AWS server is running and accessible

**Plate Recognition Failures**

Error: "Failed to get plate recognition results"

Solution:

* Verify API token is valid
* Check image quality and format
* Ensure proper lighting conditions

**Data Processing Issues**

Error: "Failed to extract data"

Solution:

* Check if the Vaxtor interface structure has changed
* Verify table elements are present
* Ensure proper permissions for data saving

### 4.2 System Recovery

* The system includes automatic retry mechanisms (3 attempts)
* After 3 consecutive failures, the browser session is recreated
* If issues persist, restart the scraper and AWS server

### 4.3 Camera Setup Issues

* Ensure all lights are green on the Ethernet switch
* Use AXIS IP Utility to verify camera connection
* Check if the Vaxtor URL is accessible in the browser