📦 Amazon Multi-Product Price Tracker

Complete Setup Guide & Email Configuration Tutorial

Table of Contents

System Requirements

Minimum Requirements:

- **Operating System**: Windows 10/11, macOS 10.14+, or Linux (Ubuntu 18.04+)
- **Python**: Version 3.6 or higher
- **Internet Connection**: Stable broadband connection
- **Storage**: 50MB free space
- **Email**: Gmail account (for optional price alerts)

Recommended:

- Python 3.8 or higher for best performance
- 1GB RAM available
- SSD storage for faster CSV operations

Python Installation

Windows Users:

1. **Download Python**
   - Go to python.org/downloads
   - Click "Download Python 3.11.x" (latest version)

- Choose the Windows installer (64-bit recommended)

2. **Install Python**
   - Run the downloaded installer
   - ⚠️ **IMPORTANT**: Check "Add Python to PATH" during installation
   - Choose "Install Now"
   - Wait for installation to complete
3. **Verify Installation**
   - Open Command Prompt (Win + R, type `cmd`, press Enter)
   - Type: `python --version`
   - You should see: `Python 3.11.x`

macOS Users:

1. **Using Homebrew (Recommended)**

```bash
# Install Homebrew if not installed
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.

# Install Python
brew install python
```

2. **Using Official Installer**
   - Download from python.org/downloads
   - Run the .pkg installer
   - Follow installation prompts
3. **Verify Installation**
   - Open Terminal
   - Type: `python3 --version`

Linux Users:

```bash
# Ubuntu/Debian
sudo apt update
sudo apt install python3 python3-pip

# CentOS/RHEL
sudo yum install python3 python3-pip

# Verify
python3 --version
```

Script Setup

Step 1: Download and Extract

1. **Create Project Folder**
   - Create a new folder: `Amazon_Price_Tracker`
   - Place the script file inside this folder
2. **Install Required Packages**

```bash
# Windows
pip install requests beautifulsoup4 schedule

# macOS/Linux
pip3 install requests beautifulsoup4 schedule
```

Step 2: Verify Dependencies

Create a test file `test_imports.py` with:

```python
try:
    import requests
    import bs4
    import schedule
    print("✅ All dependencies installed successfully!")
except ImportError as e:
    print(f"❌ Missing dependency: {e}")
```

Run it:

```bash
python test_imports.py  # Windows
python3 test_imports.py  # macOS/Linux
```

Step 3: File Structure

Your folder should contain:

```
Amazon_Price_Tracker/
├── amazon_price_tracker_multiproduct.py
├── test_imports.py (optional)
└── (other files will be created automatically)
```

Gmail App Password Configuration

Why App Passwords?

Google requires App Passwords for third-party applications to access Gmail securely. **Never use your main Gmail password** in scripts - it's insecure and won't work with 2FA enabled.

Step-by-Step App Password Setup:

Step 1: Enable 2-Factor Authentication

1. **Go to Google Account Settings**
   - Visit myaccount.google.com
   - Sign in to your Gmail account
2. **Navigate to Security**
   - Click "Security" in the left sidebar
   - Scroll to "Signing in to Google"
3. **Enable 2-Step Verification**
   - Click "2-Step Verification"
   - Follow the setup wizard
   - Choose your preferred method (SMS, Authenticator app, etc.)
   - Complete the verification process

Step 2: Generate App Password

1. **Access App Passwords**
   - Still in Security settings
   - Look for "2-Step Verification" section
   - Click "App passwords" (appears only after 2FA is enabled)
2. **Create New App Password**
   - Click "Select app" dropdown
   - Choose "Other (Custom name)"
   - Enter: "Amazon Price Tracker"
   - Click "Generate"
3. **Save the Password**
   - Google will display a 16-character password like: `abcd efgh ijkl mnop`
   - **Copy this password immediately** - you won't see it again
   - Store it securely (password manager recommended)

Step 3: Test Email Configuration

Create a test file `test_email.py`:

python

```python
import smtplib
from email.message import EmailMessage

# Replace with your details
gmail_username = "your_email@gmail.com"
app_password = "your_16_character_app_password"  # No spaces
recipient = "recipient@email.com"

try:
    msg = EmailMessage()
    msg["Subject"] = "Test Email from Price Tracker"
    msg["From"] = gmail_username
    msg["To"] = recipient
    msg.set_content("If you receive this, email configuration is working!")

    with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
        smtp.starttls()
        smtp.login(gmail_username, app_password)
        smtp.send_message(msg)

    print("✅ Email test successful!")
except Exception as e:
    print(f"❌ Email test failed: {e}")
```

Common App Password Issues:

Issue	Solution
"App passwords" option missing	Enable 2-Step Verification first
Login fails with app password	Remove spaces from password, ensure 2FA is active
"Less secure app" error	Use App Password, not main password
Connection timeout	Check firewall/antivirus settings

First Run & Usage

Step 1: Launch the Script

```bash
# Windows
python amazon_price_tracker_multiproduct.py

# macOS/Linux
python3 amazon_price_tracker_multiproduct.py
```

Step 2: Configure Email (First Time)

When prompted:

1. Choose "yes" to enable email alerts
2. Enter your Gmail address: `your_email@gmail.com`
3. Enter your **App Password** (16 characters, no spaces)
4. Enter recipient email (can be same as sender)

Step 3: Add Your First Product

1. **Get Amazon URL**
   - Go to any Amazon product page
   - Copy the full URL from address bar
   - Example: `https://www.amazon.com/dp/B08N5WRWNW`
2. **Add Product**
   - Choose option "1" from menu
   - Paste the Amazon URL
   - Wait for product details to load
   - Confirm the product information

Step 4: Test Price Check

1. Choose option "4" (Check all prices now)
2. Wait for the script to fetch current prices
3. Check that data is saved to CSV file
4. Verify email alert was sent (if enabled)

Troubleshooting

Common Issues & Solutions:

Python Issues:

```
Error: 'python' is not recognized
Solution: Reinstall Python with "Add to PATH" checked
Alternative: Use 'py' instead of 'python' on Windows
```

Import Errors:

```
Error: ModuleNotFoundError: No module named 'requests'
Solution: pip install requests beautifulsoup4 schedule
```

Amazon Access Issues:

```
Error: 403 Forbidden or blocked requests
Solution:
- Wait 10-15 minutes between requests
- Don't run script too frequently
- Check your internet connection
```

Email Configuration Issues:

Error Message	Solution
Authentication failed	Verify App Password is correct (no spaces)
Connection refused	Check internet connection, try different SMTP port
Username/password not accepted	Ensure 2FA is enabled and using App Password
Recipient invalid	Verify recipient email address format

File Permission Issues:

```
Error: Permission denied writing to CSV
Solution:
- Run as administrator (Windows)
- Check folder permissions
- Close Excel if CSV file is open
```

Debug Mode:

Add this to the top of the script for detailed logging:

```python
import logging
logging.basicConfig(level=logging.DEBUG)
```

Advanced Usage Tips

1. Optimal Usage Patterns

Best Practices:

- Check prices once per day maximum
- Use scheduled mode for automatic monitoring
- Track 5-20 products for best performance
- Run during off-peak hours (early morning)

Avoid:

- Checking same product multiple times per hour
- Running multiple instances simultaneously
- Tracking hundreds of products at once

2. File Management

Generated Files:

- `amazon_price_history.csv` - Price data (Excel compatible)
- `tracked_products.json` - Product configuration
- `email_config.json` - Email settings (encrypted)

Backup Strategy:

```bash
# Create backup folder
mkdir backups

# Copy important files
cp amazon_price_history.csv backups/
cp tracked_products.json backups/
```

3. Customization Options

Change Check Time: Edit line in script:

```python
CHECK_TIME = "09:00"  # Change to your preferred time
```

Modify Email Template: Find the `send_email_alert` function and customize the message format.

Add Price Thresholds: You can modify the script to only alert when price drops below a specific amount.

4. Running as a Service

Windows (Task Scheduler):

1. Open Task Scheduler
2. Create Basic Task
3. Set trigger: Daily at your preferred time
4. Action: Start program `python`, Arguments: `full_path_to_script.py`

macOS/Linux (Cron):

```bash
# Edit crontab
crontab -e

# Add line for daily 9 AM execution
0 9 * * * /usr/bin/python3 /path/to/amazon_price_tracker_multiproduct.py
```

5. Data Analysis

Excel Analysis:

1. Open `amazon_price_history.csv` in Excel
2. Create pivot tables for price trends
3. Generate charts for visual analysis

Python Analysis:

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load price data
df = pd.read_csv('amazon_price_history.csv')
df['Timestamp'] = pd.to_datetime(df['Timestamp'])

# Plot price trends
for product in df['Product Name'].unique():
    product_data = df[df['Product Name'] == product]
    plt.plot(product_data['Timestamp'], product_data['Price (USD)'], label=product)

plt.legend()
plt.title('Price Trends Over Time')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Security & Privacy Notes

Data Security:

- All data stored locally on your computer
- No information sent to third parties
- Email credentials encrypted in config file
- CSV files can be password protected if needed

Amazon Terms Compliance:

- Script uses respectful request timing
- Proper user agent headers included
- No aggressive scraping or automation
- Personal use only (not for commercial scraping)

Privacy Best Practices:

- Use dedicated email for notifications
- Regularly rotate App Passwords
- Keep script files in secure location

- Don't share email config files

Support & Updates

Getting Help:

1. Check this guide first
2. Review error messages carefully
3. Test individual components (Python, email, internet)
4. Search for specific error messages online

Script Updates:

- Keep Python updated to latest version
- Update dependencies: `pip install --upgrade requests beautifulsoup4 schedule`
- Backup your data before updating script versions

Version Information:

- Script Version: 2.0
- Compatible Python: 3.6+
- Last Updated: 2025
- Supported OS: Windows, macOS, Linux

Conclusion

You now have everything needed to run the Amazon Price Tracker successfully! The script will help you:

✅ **Save Money** - Never miss a price drop again
✅ **Save Time** - Automated monitoring instead of manual checking
✅ **Stay Informed** - Instant email alerts for price changes
✅ **Track Trends** - Historical data for better buying decisions

Remember: Use the script responsibly, respect Amazon's terms of service, and keep your App Password secure.

Happy deal hunting! 🎯💰

This guide covers installation and configuration for Amazon Multi-Product Price Tracker. Keep this document for future reference and troubleshooting.