SDU Collection Automation

Contents

Disclaimer

Star Atlas developer, ATMTA, does not officially endorse this tool or its use. They maintain that while they support innovation and development in the game ecosystem, the use of this tool is being permitted on a trial basis to test the early access feature and the impact on its economy. If you are using this feature, please tune into official Star Atlas communication channels in case the advisory for this tool changes.

If the DAC team feels that this tool is causing damage to the SAGE Labs economy, it will be shut down.

Introduction

Introduction

Welcome to the step-by-step guide on setting up LABS-SDU-AUTOMATION, a tool designed to automate SDU collection using a single fleet. By following these detailed

instructions, you can save time and effort while maximizing your SDU gains. Please ensure you read carefully to avoid any issues during the setup process.

Basic Information

LABS-SDU-AUTOMATION is a collection automation tool specifically designed for SDUs in a fleet. SDUs are crucial in the game and this automation tool aims to simplify the process, allowing you to focus on more important aspects of your main account. The tool runs every 8 hours (10 Airbikes) and can produce 400-600 SDUs, enabling you to generate up to 1800 SDUs passively per day when reset and restocked.

Important Display Notes

Before you begin, ensure your monitor resolution is set to 1920 X 1080 and the taskbar at the bottom of your screen is not hidden. Failure to do so might cause the automation to click incorrectly. This tool is created for 1080p only currently. Users with a higher resolution will have to downsize their resolution in order to use the tool.

IMPORTANT: Make sure that the taskbar at the bottom of your screen is not hidden as it will not click into the right spots. Zoom in browser must be set at 100%

Step 1: Installing required software

Note: All software is free, only download from the links provided.

NPM & NodeJS

Download and install NodeJS (which includes NPM) from https://nodejs.org/en/download.

Visual Studio Code

Download and install VSCode from https://code.visualstudio.com/.

Solflare Extension for Chrome

Go to your extension settings on chrome web browser. Search for Solflare. Download and install Solflare but do set up a wallet at this time.

IMPORTANT: After you have downloaded and installed all the above, close VSCode. NPM requires a VSCode restart to install completely.

Step 2: Downloading files and opening folder in VSCode

Download the GitHub folder

Download the folder from GitHub where you found this readme. The folder is named “LABS-SDU-AUTOMATION”. Create a folder wherever you like on your drive and extract these files to this new folder.

Opening the folder in VSCode

In VSCode you will open your newly created folder by clicking on “File” -> “Open Folder” and then selecting the folder you have extracted to.

Install Libraries

Now that you have your folder open in VSCode, you are ready to properly finish installing NPM.

To do so you will:

Open a new terminal using the toolbar on the top of the screen. Terminal -> New Terminal

In the terminal Type “npm i” to install all NPM libraries.

Set Solflare extension path within the sa.ts file

1. Open the sa.ts file in VSCode from the folder
2. On line 8 in sa.ts update the Solflare extension path to include your windows profile name.

It should look something like this:

"C:\\Users\\YourWindowsProfleName\\AppData\\Local\\Google\\Chrome\\UserData\\Default\ \Extensions\\bhhhlbepdkbapadjdnnojkbgioiodbic\\1.52.0_0".

(ALL EXTENSION PATHS FOR SOLFLARE WILL HAVE "bhhhlbepdkbapadjdnnojkbgioiodbic".)

Make sure between each folder in the extension path there are two backslashes and should look like "\\" if this is incorrect it will not run.

IMPORTANT: Solflare extension will auto update. When this occurs, you will get an error when trying to start the automation tool. To correct this error just update the path set for your extension with the new version number of Solflare extension.

You can find this info by going into your extensions in chrome, clicking details next to Solflare. The version # shown on this page is the version number you need in your extension path on line 17.

Step 3: Setup your Solflare extension within the tool

Users with a current Solflare wallet will need to create a new wallet as well as a new burner wallet specific to this project

Launch the automation browser

In the VSCode terminal type “npm start run” (Main automation start command). This will start the chromium browser used by the tool.

IMPORTANT: Do not interact with any pop-ups during the initial setup

Setup Solflare burner wallet

Click the Solflare tab at the top of the browser. Go through the settings of setting up your new Solflare wallet

IMPORTANT: Save your phrase before moving on

After you have set up your new Solflare wallet you will need to create a burner. To do so, follow these steps:

1. Make sure you have the Solflare extension pinned within the tool browser
2. Open the Solflare extension
3. Click the circle in the top left corner of the extension
4. Click the “+” in the top right corner
5. Click “Create Burner Wallet”
6. Click continue 3 times

Backup your new burner wallet

Now that you have created your burner wallet within the tool browser, you will want to backup your new burner wallet within your normally used browser. To do so, follow these steps:

1. Swap the active wallet to the newly created burner wallet within the Solflare extension while still in the tool's browser
    a. To do so click the circle in the top left
    b. Locate the burner wallet and click on it
2. Click the 3 vertical dots that are next to the "copy" when you scroll over the burner wallet.
3. Click "Export Private Key"
4. Copy this key
5. Open up the browser with your normal Solana wallet.
    a. For Solflare
        i. Login
        ii. Click the circle in the top left corner iii. Click the "+"
        iv. Click "Import Private Key"
        v. Paste the key you copied in the earlier steps and give the wallet a name
        vi. Click "Import"
    b. For Phantom –
        i. Login
        ii. Click the 3 horizontal bars in the top left corner
        iii. Click the "+"
        iv. Click "Import Private Key"
        v. Paste the private key you copied in the earlier steps and give the wallet a name
        vi. Make sure the coin is set to Solana vii. Click "Import"

Setup Solflare burner wallet

Now you have your wallet backed up and have the burner wallet selected as the active wallet within the tool's browser.

Click the gear icon in the right corner of the extension while still in the tool's browser.

Click "Security & Privacy"

Click "Auto-Lock" and set this option to "Never"

IMPORTANT: After completing all the steps above for the burner wallet you will need to close both the tool's browser and VSCode. Once both have been closed, reopen VSCode to continue with the next steps.

Configure LABS settings within the extension

The last step for setting up the burner wallet to work properly within the tool's browser is to allow auto approval. To do so, follow these steps:

1. In VSCode terminal, type "npm start run"
2. Click the Solflare tab at the top of the browser and confirm that it shows your new burner wallet. Then click back onto the labs tab.
3. First, make sure that you try logging into LABS first where a popup will show up asking you to connect to the browser. Click connect.
4. Click back onto the Solflare tab and click the gear icon near the top right corner
5. Click "Security & Privacy" -> "Manage Apps" -> "SAGE Labs" -> Enable autoconnect & auto-approve.

IMPORTANT: Again, close the tool's browser and VSCode. Then reopen VSCode.

Step 4: Prepare your fleet

Add Ships, Resources, & Coin

Now you will stock your newly created burner wallet using your normal browser and normal wallet extension.

You will need to send the following items from your main wallet to the newly created burner wallet. You can do this easily as you have already imported your burner wallet into your main wallet extension within your normal browser.

Your burner account will need the following items:

- Toolkits
- Fuel
- Solana
- Atlas

- 10 Airbikes

Setting up your fleet on the burner account

Now that you have sent the required resources and the ships needed to the burner account you will need to set them up in a fleet and get them stocked for deployment.

1. Open up your normally used browser and login to your burner wallet in the extension.
2. Load SAGE Labs while logged into the burner and go through the account creation process.
3. Once completed you will need to go into the CSS and create the Fleet. Load all 10 of the airbikes into the fleet and stock the fleet 100% with fuel and load 2480 toolkits.
4. Undock your fleet
5. Select sub warp then select the tile you want your fleet to scan at. Your fleet will not move, they will sit in this tile scanning until they are out of toolkits
6. Once your fleet has arrived at its destination you will have to “Exit Subwarp” as the tool will not do this for you.

Now your fleet should be sitting “Undocked/Idle” at your chosen tile. Once you reach this step you can close this browser.

Step 5: Activate the automation

Activate the tool

You must reopen VSCode and then open the sa.ts file within the folder you opened previously.

On line 7 of this file, set the variable paused equal to false “let paused = false”

You have now completed all of the required steps and you’re ready to start the automation tool.

To do so: type “npm start run”

Monitor and replenish resources

While the scanning process of SDU is automated using this tool, the resource replenishment is not. You will have to monitor your fleet. When your fleet runs out of resources you will have to replenish them. To do so follow Step 4: Prepare your fleet > Setting up your fleet on the burner account. You will also want to deposit your collected SDUs to your star base/wallet at this time.

Once your fleet is replenished and back in an undocked/idle state in your chosen scanning zone you will type “npm start run” in the VSCode terminal.

IMPORTANT: Every time you need to replenish your fleet, you will need to close the tool’s browser and VSCode. Once you have stocked your fleet using your normal browser you will then reopen VSCode and run the command shown above.

Conclusion

Congratulations! You have successfully set up LABS-SDU-AUTOMATION for your SDU collection process. Enjoy your increased efficiency and SDU gains in the game. Remember to monitor the process periodically and replenish resources as needed for uninterrupted automation.

IMPORTANT: As mentioned many times. This process is only meant to automate the collection process of SDUs. PLEASE, do not abuse this system. Going against the Star Atlas terms of service could get you banned.

If the ATMTA team feels that this tool is causing damage to the economy, it will be shut down.

Thank you for installing our automation tool! If this helped you out in any way, Donations would be greatly appreciated! Do so in any one of the three ways listed below:

- Use our Starpath link (Get 10% off select items in the Star Atlas marketplace):
  - https://play.staratlas.com/?r=nhtk3t9cljg7hzkd
- Purchase our photo contest winner NFT “PCC #1”:
  - https://play.staratlas.com/market/FfRV7CcedJ284kkaQf4CY9XaaZHdkHaBcFU77qTAKT4n/las
- Directly send donations to the following Solana address:
  - DKKuwZ8J7u9FLbgfSzhNFcUKKYDgX2eLkvhvobubs78L

While the Star Atlas team does not officially endorse this tool, its use is being permitted on a trial basis to test the game systems and the impact on the in-game economy. As the tool has not been audited or reviewed for stated functionality, you are using it at your own risk. Star Atlas supports innovation and development in the gaming ecosystem, and welcomes products that are easily accessible to the whole community, open for use and free of charge.