

# Documentation: Aggregating and Processing Blockchain Token Data

---

## Step 1: Access Trust Wallet's Blockchain Repository

1. Visit the [Trust Wallet Assets Repository](#).
  2. Navigate to the **blockchains** folder.
  3. Select the desired blockchain folder (e.g., ethereum, optimism, polygon, etc.).
- 

## Step 2: Aggregating info.json File

Run the following PowerShell script to collect all info.json files across the chosen blockchain's directory structure and combine their contents into a single JSON file:

```
Get-ChildItem -Recurse -Filter "info.json" |  
ForEach-Object { Get-Content $_.FullName -Raw | ConvertFrom-Json } |  
ConvertTo-Json -Depth 10 |  
Set-Content combined.json
```

- This script recursively scans for info.json files and consolidates their content into combined.json.
- 

## Step 3: Filtering Out Abandoned Coins

To exclude tokens marked as **abandoned**, run the following script:

```
(Get-Content combined.json -Raw | ConvertFrom-Json) |  
Where-Object { $_.status -ne "abandoned" } |  
ConvertTo-Json -Depth 10 |  
Set-Content output.json
```

- This script reads combined.json, filters out entries with status set to "abandoned", and saves the cleaned data into output.json.
- 

## Step 4: Preparing Data for the Swap API

1. **Copy output.json:**
  - Place the cleaned output.json file into the **swap-api** folder of your project.
2. **Modify index.js:**
  - Map the token data into the desired format by adding the following code snippet:
  - `const processData = jsonArray.map(obj => ({`
  - `chainId: 10, // Update this to the appropriate chain ID`

- name: obj.name,
- symbol: obj.symbol,
- decimals: obj.decimals,
- address: obj.id,
- logoURI:  
`https://raw.githubusercontent.com/trustwallet/assets/c37119334a24f9933f373c6cc028a5bdbad2ecb4/blockchains/optimism/assets/\${obj.id}/logo.png`
- `}); // Update "optimism" to the relevant chain name`
- Replace **10** with the correct chain ID for the chosen blockchain.
- Update the chain name in the logoURI path.

### 3. Run index.js:

- Execute the script to generate the final processed token list.
- The data will come into the finalOutput.json file.

---

## Step 5: Integrating with the Frontend

1. Add the generated file from the **swap-api** folder to your frontend project.
2. Update frontend configurations to use the new token list.
3. Ensure the frontend's API integrations align with the processed data format.