djoseph-png / **py-matecoin-trades**

Code Pull requests Actions Projects Wiki Security Insights Settings

0 stars 1.4k forks 0 watching Branches Activity
Tags

Public repository · Forked from mate-academy/py-matecoin-trades

1 Branch 0 Tags Go to file t Go to file Add file Code

This branch is up to date with `mate-academy/py-matecoin-trades:master` . Contribute Sync fork

IvanRamyk Fix typo in readme bccf8e7 · 2 years ago

.github/workflows	add auto-approve and tests	3 years ago
app	Matecoin trades	3 years ago
tests	fix tests	3 years ago
.flake8	Update .flake8	3 years ago
.gitignore	Initial commit	3 years ago
README.md	Fix typo in readme	2 years ago
requirements.txt	Update requirements.txt	3 years ago

README

Matecoin trades

- Read the guideline before start

You have made some trades with the new cryptocurrency "Matecoin". In order to calculate profit you loaded json-file `trades.json` with trades description. Content of that file is `list of JavaScript Objects` , `js objects` very similar to python `dict` , where keys are `"bought"` - volume of coins, if you bought, `"sold"` - volume of coins, if you sold and `matecoin_price` - price for 1 Matecoin in dollars that day.

Example:

```
// trades.json
[
  {
    "bought": "0.00111",
    "sold": null,
    "matecoin_price": "48911.23"
```

```
  },
  {
    "bought": null,
    "sold": "0.00058",
    "matecoin_price": "77830.83"
  }
]
```

Write `calculate_profit` function, which takes the name of the file with trades information, converts inside data into python native data structure, counts money profit in dollars and current coin account, dumps that information in json and saves in file `profit.json`

```
// profit.json
{
  "earned_money": "-9.1495839",
  "matecoin_account": "0.00053"
}
```

Use `Decimal` instead of `float` here, because `float` do not provide sufficient accuracy for monetary transactions. All numbers in json should be presented in string format.

IMPORTANT: use `load` and `dump` methods for write/read.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Suggested workflows

Based on your tech stack

Python Package using Anaconda

Configure

Create and test a Python package on multiple Python versions using Anaconda for package management.

Django

Configure

Build and Test a Django Project

Pylint

Configure

Lint a Python application with pylint.

More workflows

Dismiss suggestions