

## GMX Python SDK Grant Bi-Weekly Update

Date: 27-12-23

### I. Executive Summary:

- Refactoring of all scripts underway, incl design and creation of “Orders” Class
- Implemented threading to reduce latency on multi contract reads
- Additional tasks were added and completed, including querying latest borrow rate, funding rate, gm token price, and utilisation

#### a) Roadmap Progress:

A rough and expanding guideline of required scripts can be found here, with status of each denoted:

<https://www.notion.so/gmx-python-sdk/Required-Scripts-Overview-04da3b61d0e147039a8323316b0effec>

This is not an exhaustive list and is being continually updated as new requirements are made.

#### b) Milestone/KPI Achievements:

Currently Milestones 2 & 3 are being addressed. The implementation of the encompassing “Orders” Class will take the isolated 3 scripts for increase/decrease/swap orders and combine into one easy building block for users to use at their whim. This is expected to be completed come the next bi-weekly review.

#### c) Dune Dashboards - n/a

### II. Current Progress:

The last 1-2 weeks saw the implementation of:

- First formatting and linting based code review completed
- get GM token prices
- get borrow APRs
- get funding APRs
- get Utilisation
- Implemented threading - As multithreaded python packages are not compatible with the latest version of web3py threading has been used when multiple onchain calls are required. This will require a higher rate limit on the users RPC
- allow output of [read] data to be saved to csv

### III. Community Engagement:

n/a

### IV. Partnerships and Collaborations:

n/a