

\$Alpha Dragon

How to contribute?

Contribute. Capture Alpha. Get Rewarded.

Last updated: June 11, 2025

email: baddragonmygoodness@gmail.com

Step 1. Submit Your Module Idea

📌 **Where:** [GitHub Discussions → Module Ideas](#)

◆ **Category:** Module Ideas

1. Describe your proposed:

- What signal it captures (e.g., liquidity behavior, Telegram spikes, MEV clustering)
- Why it could be a meaningful alpha indicator
- Source type: on-chain, off-chain, or hybrid

2. Optional: Share links to dashboards or examples that inspired it.

Step 2. Bounty Scoping by Dragon Team


📌 **Where:** GitHub reply + internal tracker

◆ Once a proposal is submitted, the Dragon launch team will:

- Review the idea for feasibility and uniqueness
- Estimate development complexity and potential ecosystem value
- *Respond to your Discussion post* with:
 - Approval status
 - Bounty range (e.g., 500 \$ALPHA depending on implementation depth)
 - Tips or technical notes if needed

Once the bounty is posted, you're free to begin building.

Step 3. Build the Module

 **Where:** A dedicated GitHub repository will be created for your module under the Alpha Dragon GitHub organization.

◆ **Tech Stack:** Python, Rust, or JavaScript

◆ **Standards:** Follow Dragon's module format ([dragon repos](#))


1. Your module should:

- Accept a *token address* as input
- Output a clean, structured JSON object (e.g., {"token": "XYZ", "signal_score": 6.7})
- Be lightweight enough for browser or backend RPC querying

2. Your repo must include:

- A *README.md* explaining the module's logic, data sources, and usage
- Example output under /examples
- Any environment setup or config notes in a requirements.txt or equivalent

Step 4. Testing & Integration

 **Where:** Dragon test extension or local dev environment

◆ **The team will test:**

- Output correctness
- Performance (e.g., latency, API load)
- Alpha signal viability (match to historical events)

◆ Once approved, the module is:

- Added to the extension module list
- Indexed by the orchestration layer (AlphaSwarm)
- Available to users through Dragon UI

Step 5. Bounty Payout

📌 **Where:** On-chain wallet transfer (initially)

◆ Paid in \$ALPHA from the dev bounty pool

After successful integration:

- You'll receive your agreed-upon bounty
- High-performing modules may get ongoing revenue share from protocol dev pool (smart contract to be developed)
- You'll be added to the Dragon Contributor Registry

Coming Soon:

- **Module Forge:** drag-and-drop *WASM/Python module builder*
- **Live Leaderboard:** rank modules by *impact + adoption*
- **Reputation system:** for dev visibility and *DAO governance*