Request Structure

What I suggest

- Requests on my projects come in key-value pairs that are easily understandable and clearly defined:
  - *user-input: “This is what the user said!”*
  - *convo-summary: “This is a breakdown of what was said previously”*
- You can list a number of these key-value pairs. Split up the input as much as you need to, as long as it remains logical and the AI will understand what you’re getting at after defining your input elements

Response Structure

What I suggest

- Returning predefined functions, instead of unstructured output helps with parsing, logical coherence, AI understanding, and preventing anomalous responses or AI hallucinations
- Be sure to define your arguments ahead of time and stress to the AI that the arguments must remain in the correct order, with the correct argument type (string, num, likelihood/1000, etc.)
- Examples:
  - *try("Outcome 1", "Outcome 2", 100/1000)* — 10% chance outcome 1 is successful, 90% chance outcome 2 happens
  - *attack(damage, 20)* — the attack does 20 points of damage. In this circumstance, damage is a keyword, not a string
  - *speak("this is some example dialogue")* — Example dialogue coming from the AI