

# Dune AI Prompt Engineering

How to craft a successful Dune AI prompt

Welcome to the Dune AI Prompt Engineering Guide! This guide is designed to help users understand how to effectively communicate with Dune AI, focusing on its powerful capabilities in parsing smart contracts and identifying project names within the blockchain analytics domain. Whether you're a developer, researcher, or enthusiast, this guide will provide you with the knowledge to leverage Dune AI's features for your projects.

## Getting Started

Before we dive into the examples, it's important to understand the basics of prompt engineering with Dune AI. Prompt engineering involves crafting questions or commands that guide the AI in performing specific tasks. A well-designed prompt results in more accurate and useful responses.

## Principles of Effective Prompting

- Clarity
  - : Be clear and specific about what you're asking.
- Context
  - : Provide enough context to guide the AI's response.
- Conciseness
  - : Keep prompts concise to avoid overwhelming the AI.

## Capabilities Overview

Dune AI offers a range of capabilities, especially suited for blockchain analytics. Key features include:

- Parsing smart contracts addresses
- and identifying the relevant functions and their tables.
- Identifying and categorizing project names
- and mapping them to their respective contract tables and spells (expertly curated datasets)
- Identifying token symbols
- and identifying their contract addresses and contract types.
- Injecting your wallet address
- (when linked in profile) to provide personalized insights and analytics.

Below are examples of how you can structure your prompts to make the most out of Dune AI's capabilities.

### Smart Contracts: Extracting relevant events via contract addresses

Prompt : "List recent trades on 0xcf205808ed36593aa40a44f10c7f7c2f67d4a4d4 from trader 0x0b660e916537a75f8d7d3ac7dfe6c8678a8ac5fa"

Prompt Engineering Tip : Include a smart contract address. Given the contract address 0xcf205808ed36593aa40a44f10c7f7c2f67d4a4d4 , we can find that it corresponds to the FriendtechSharesV1 contract and the LLM can then pick the appropriate eventTrade to recommend the tablefriendtech\_base.FriendtechSharesV1\_evt\_Trade .

### Identifying project names and finding their respective "spells" or crafted datasets

Prompt : "list of wallets that have traded more than 2500 usd on blur"

Prompt Engineering Tip : Include project names or schemas in your prompt. Below including blur in the prompt will help Dune AI identify all spells under that schema. Blur has many spells but the LLM selects the most relevant one (blur.trades ) based on the context of the prompt. If you want to validate the spelling of the project name, all schemas are listed in the [data explorer](#)

Identifying project names and finding relevant events

Prompt : “top transfer receiver on lido on base”

Prompt Engineering Tip : Include project names or schemas in your prompt. Similar to finding relevant spells, including project names in your prompt will help Dune AI identify the relevant functions and tables. In this example forlido , the LLM selects the most relevant eventTransfer and thelido\_base.wsteth\_evt\_Transfer based on the context of the prompt.

Identifying token symbols and finding their contract addresses and contract types

Prompt : “DAI token swaps today”

Prompt Engineering Tip : Including token symbols (in CAPs) in your prompt will help Dune AI identify the relevant contract addresses and contract types. In this example, the LLM selects the most relevant contract address0x6b175474e89094c44da98b954eedeac495271d0f and contract typeERC20 based on the context of the prompt. Many token addresses are already embedded in the LLM’s knowledge base, but using CAPS will help this happen deterministically. This isn’t limited to EVM and this works also for Solana tokens.

Asking about your own wallet

Prompt : “what chain has my wallet transacted the most value?”

Prompt Engineering Tip : Saying “my wallet” will prompt the LLM to use the wallet address linked to your profile to provide personalized insights and analytics.

Best Practices for Prompt Engineering

- Iterate
- : Don’t hesitate to refine your prompts based on the responses you get.
- Be Specific
- : The more specific your prompt, the more targeted the AI’s response will be.
- Use Examples
- : When possible, provide examples within your prompt to guide the AI’s understanding.

Was this page helpful?

Yes No [Raise issue](#) [Create AI-Assisted Queries](#) [Create New Content](#) [linkedin](#) [github](#) [twitter](#) [discord](#) [telegram](#) [youtube](#)  
[Powered by Mintlify](#)