

**ESOL TECHNOLOGY SOLUTION JOINT STOCK COMPANY**

**USER GUIDELINE**

***BRKT - PHASE 2***

| Code: | BRKT |
| --- | --- |
| Version: | 1.0.0 |
| Date of version: | August 23, 2024 |
| Created by: | Quality Assurance - ESOLLABS |
| Confidentiality level: | Confidential |

**Change history**

| **Date** | **Version** | **Created by** | **Description of change** |
| --- | --- | --- | --- |
| 23/08/2024 | 1.0.0 | Quality Assurance - ESOLLABS | Create new |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**I. OVERVIEW 3**](#_2et92p0)

[1. INTRODUCTION 3](#_tyjcwt)

[2. PRE-REQUISITES 3](#_ipcx72tmcfl7)

[3. INTENDED AUDIENCE 3](#_xcsvz1qnx7iq)

[**II. SETUP AND INSTALLATION 4**](#_3cnvp57kauqk)

[1. INTRODUCTION 4](#_s9kvn4jmg9c0)

[2. INSTALL APTOS CLI 4](#_rgtpl6u77inv)

[2.1 WINDOW VERSION 4](#)

[2.2 LINUX VERSION 7](#_sj96qxv1k5y9)

[**III. PROJECT STRUCTURE 7**](#_gcsm1oynb04w)

[1. STRUCTURE 7](#_yhmvapyyhwib)

[2. DETAILS EXPLANATION 8](#_amyzbnz00thw)

[1. INITIALIZE 10](#_h2qtzn405hwl)

[2. COMPILE 11](#_zdqi0lyld7zp)

[3. PUBLISH 11](#_qe2k7k1epkm)

[3.1 CONFIRM ACCOUNT 11](#_gbtt7cipauao)

[3.3 PUBLISH MODULES 12](#_ofz1n58af0k2)

[4. VERIFY DEPLOYMENT 13](#_hrn80javt2f9)

[**V. INTERACT WITH THE SMART CONTRACT 17**](#_ra43mrq29th7)

[1. CODE SECTION 19](#_2ta1y2a02vh3)

[2. RUN SECTION 20](#_142ra3l18qv1)

[3. VIEW SECTION 25](#_ikl0bvxrxg8v)

# OVERVIEW

## INTRODUCTION

Welcome to the future of competitive gaming on the blockchain! This document serves as a guide for users to understand, deploy, and interact with the smart contracts.

## PRE-REQUISITES

* Aptos CLI version: v3.4.0+
* Python 3.6+

## INTENDED AUDIENCE

* Developers
* QA/QC
* Product Owner
* Project Manager

# SETUP AND INSTALLATION

## INTRODUCTION

The easiest way to install the Aptos CLI tool is via Python script. If that does not work, you can also install manually via pre-compiled binaries. The pre-compiled binaries approach is not generally recommended as updating is very manual.

## INSTALL APTOS CLI

### **WINDOW VERSION**

1. Go to the [Aptos CLI release page](https://github.com/aptos-labs/aptos-core/releases?q=cli&expanded=true).
2. Expand “Assets” to see the pre-compiled binaries.
3. Download the zip file for Windows.

* It will have a name like: aptos-cli-<version>-Windows-x86\_64.zip .

1. Unzip the downloaded file.

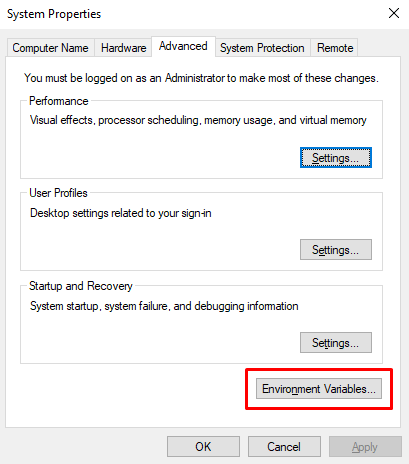
* Move the file to whichever folder you would like to call **aptos** from in the future.

1. Set up the system environment variable

* Right click, then copy the path to the executable.

Ex: C:\Users\<username>\Downloads\aptos-cli-3.5.0-Windows-x86\_64\aptos.exe .

* Go to “Edit the system environment variables” in your computer and select “Environment Variables”.



* Add this path to your PATH environment variable to simplify calling the Aptos CLI going forward.

1. Verify the installation by running the help command

* Open PowerShell via the Start Menu
* Type “aptos help” to call the Aptos CLI.

| PS C:\Users\bkrt> aptos help Command Line Interface (CLI) for developing and interacting with the Aptos blockchain  Usage: aptos <COMMAND>  Commands:  account Tool for interacting with accounts  config Tool for interacting with configuration of the Aptos CLI tool  genesis Tool for setting up an Aptos chain Genesis transaction  governance Tool for on-chain governance  info Show build information about the CLI  init Tool to initialize current directory for the aptos tool  key Tool for generating, inspecting, and interacting with keys  move Tool for Move smart contract related operations  multisig Tool for interacting with multisig accounts  node Tool for operations related to nodes  stake Tool for manipulating stake and stake pools  update Update the CLI or other tools it depends on  help Print this message or the help of the given subcommand(s)  Options:  -h, --help Print help  -V, --version Print version |
| --- |

### **LINUX VERSION**

1. In the terminal, use one of the following commands:

| $ curl -fsSL "https://aptos.dev/scripts/install\_cli.py" | python3 |
| --- |

Or use the equivalent wget command :

| $ wget -qO- "https://aptos.dev/scripts/install\_cli.py" | python3 |
| --- |

1. Verify the script is installed by opening a new terminal and running “aptos help”

# PROJECT STRUCTURE

## STRUCTURE

| smart-contract/  ├── .aptos/  │ └── config.yaml  ├── sources/  │ ├── base/  │ │ ├── competition\_state.move  │ │ ├── competition.move  │ │ └── match\_outcome.move  │ ├── predictable/  │ │ ├── paid\_predictable\_competition\_state.move  │ │ ├── paid\_predictable\_competition.move  │ │ ├── predictable\_competition\_state.move  │ │ ├── predictable\_competition.move  │ │ ├── registration\_fee\_info.move  │ ├── competition\_factory.move  │ ├── competition\_route.move  ├── .gitignore  ├── Move.toml  └── README.md |
| --- |

## DETAILS EXPLANATION

* “.aptos/”: A hidden directory containing configuration files and necessary information to connect and interact with the Aptos network.
* “Config.yaml”: The main configuration file specifying environment parameters, variables, and settings for the project.
* “sources/”:
  + “base/”:
    - “competition\_state.move”: Defines the state of the competitions.
    - “competition.move”: Smart contract related to basic competitions.
    - “Match\_outcome.move”: Defines the outcomes of matches within the competition.
  + “predictable/”:
    - “Paid\_predictable\_competition\_state.move”: Manages the state of paid predictable competitions, including tracking entry fees, participant status, and the overall progress of the competition.
    - “Paid\_predictable\_competition.move”: Handles the logic for paid predictable competitions, managing predictions, calculating outcomes, and distributing rewards based on prediction accuracy.
    - “Predictable\_competition.move”: Smart contract for predictable competitions.
    - “Predictable\_competition\_state.move”: Manages the state, tracking participants, predictions, and the overall competition status.
    - “Registration\_fee\_info.move”: Handles the management of registration fees and storing.
  + “Competition\_factory.move”: Factory contract to create competitions.
  + “Competition\_route.move”: Defines the routing and flow control for different stages of the competition, ensuring smooth transitions between phases such as registration, prediction, and prize distribution.
  + “.gitignore”: A file that specifies which files and directories should not be tracked by Git, such as personal configuration files or build directories.
  + “Move.toml”: The project configuration file within the Move ecosystem, containing information about the package and necessary parameters.
  + “README.md”: Documentation file that provides an overview of the project, including installation instructions, usage, and other important details.

# PUBLISH SMART CONTRACT

## INITIALIZE

The CLI reads the account profile, applies its configurations, and create a new profile account with the correct settings. This helps streamline the setup process and ensures consistency across different environments or setups.e.g.

| aptos init --profile default |
| --- |

* **First Enter**: The CLI will prompt you for the necessary details to create the profile. The first press of enter is to confirm the default network is devnet. You can change according to your wishes.

| A:\BRKT\src\smart-contract> aptos init --profile default Configuring for profile default Choose network from [devnet, testnet, mainnet, local, custom | defaults to devnet] |
| --- |

* **Second Enter**: After providing the details, you may need to press Enter again to confirm and complete the profile creation process by the new account address with information such as public key and private key.

| Enter your private key as a hex literal (0x...) [Current: None | No input: Generate new key (or keep one if present)]  No key given, generating key... Account 0x0000000000000000000000000000000000000000000000000000000000000000 doesn't exist, creating it and funding it with 100000000 Octas Account 0x0000000000000000000000000000000000000000000000000000000000000000 funded successfully  --- Aptos CLI is now set up for account 0x0000000000000000000000000000000000000000000000000000000000000000 as profile new!  See the account here: https://explorer.aptoslabs.com/account/0x0000000000000000000000000000000000000000000000000000000000000000?network=devnet  Run `aptos --help` for more information about commands {  "Result": "Success" } |
| --- |

## COMPILE

Next, use the CLI to compile the modules. In window version use the following command:

| aptos move compile |
| --- |

And in Linux version, you need to use the following command to compile modules:

| $ sudo aptos move compile --package-dir sources |
| --- |

## PUBLISH

### **CONFIRM ACCOUNT**

Ensure that you have a valid account in the .aptos/config.yaml file, e.g.

| --- profiles:  default:  private\_key: "0x0000000000000000000000000000000000000000000000000000000000000000"  public\_key: "0x0000000000000000000000000000000000000000000000000000000000000000"  account: 0000000000000000000000000000000000000000000000000000000000000000  rest\_url: "https://fullnode.devnet.aptoslabs.com"  faucet\_url: "https://faucet.devnet.aptoslabs.com" |
| --- |

### **SETUP MOVE FILE**

Change the value of brkt\_addr variable to fit your account, e.g.

| [package] name = "brkt" version = "1.0.0" authors = []  [addresses] brkt\_addr='0000000000000000000000000000000000000000000000000000000000000000'  [dev-addresses]  [dependencies.AptosFramework] git = "https://github.com/aptos-labs/aptos-core.git" rev = "mainnet" subdir = "aptos-move/framework/aptos-framework"  [dev-dependencies] |
| --- |

### **PUBLISH MODULES**

Use the CLI to publish the modules to the aptos explorer. The following command in window version:

| aptos move publish --profile default |
| --- |

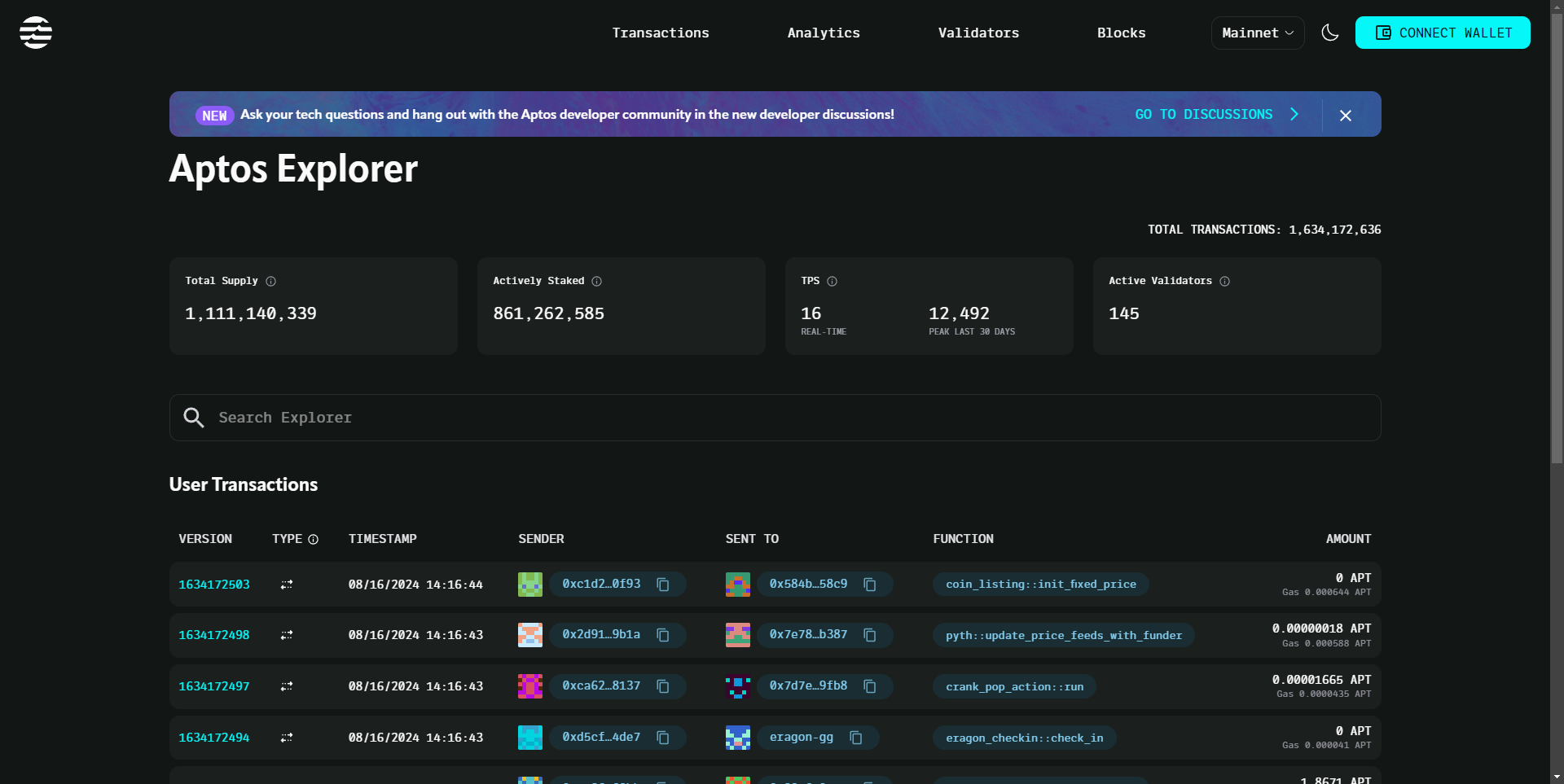
Using the following command to publish modules in Linux version:

| $ sudo aptos move publish --profile <the-account-name-in-the-config.yaml-file> |
| --- |

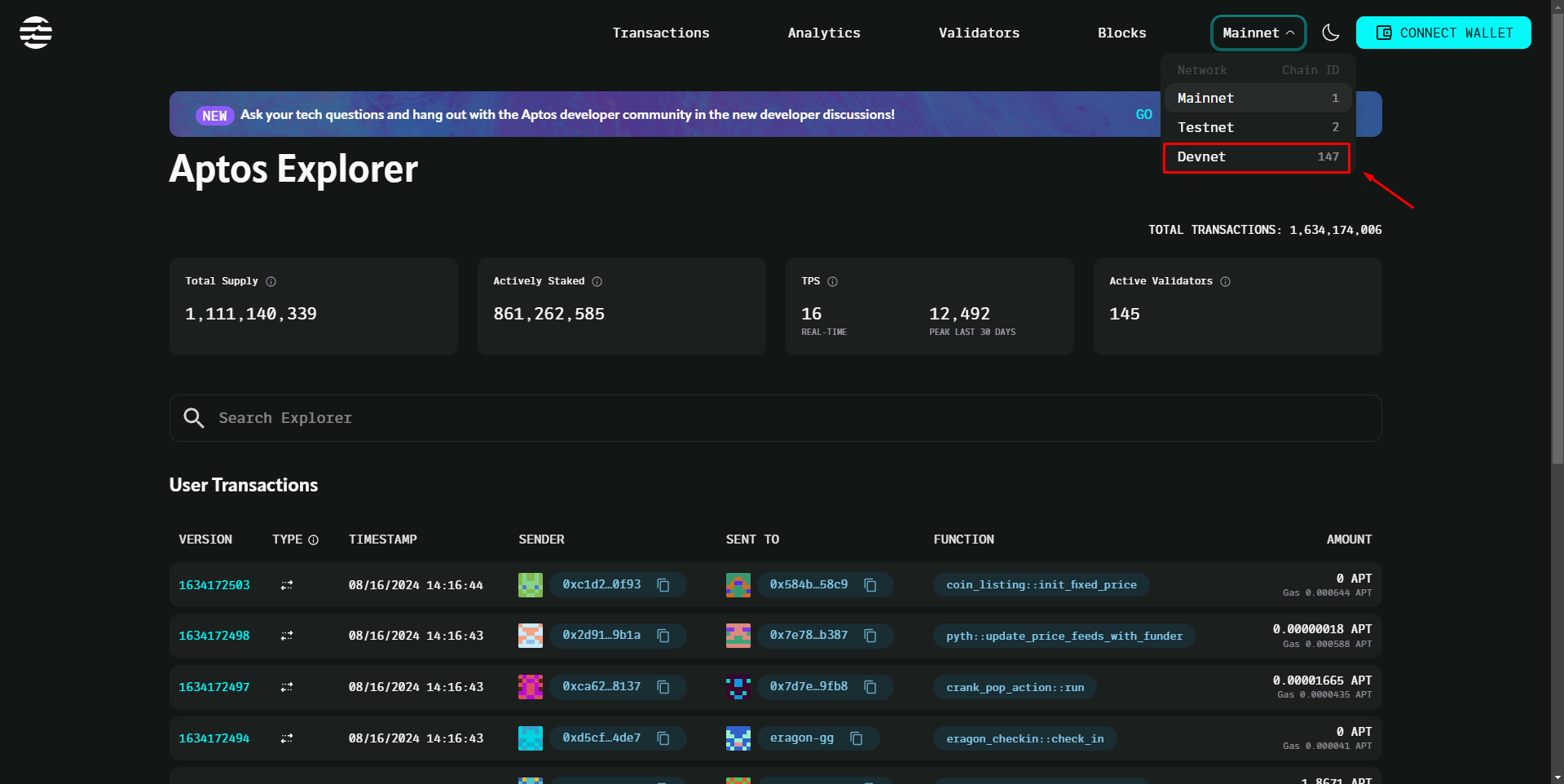
The contract is now live on the blockchain and can be interacted with according to the logic defined in the Move code.

## VERIFY DEPLOYMENT

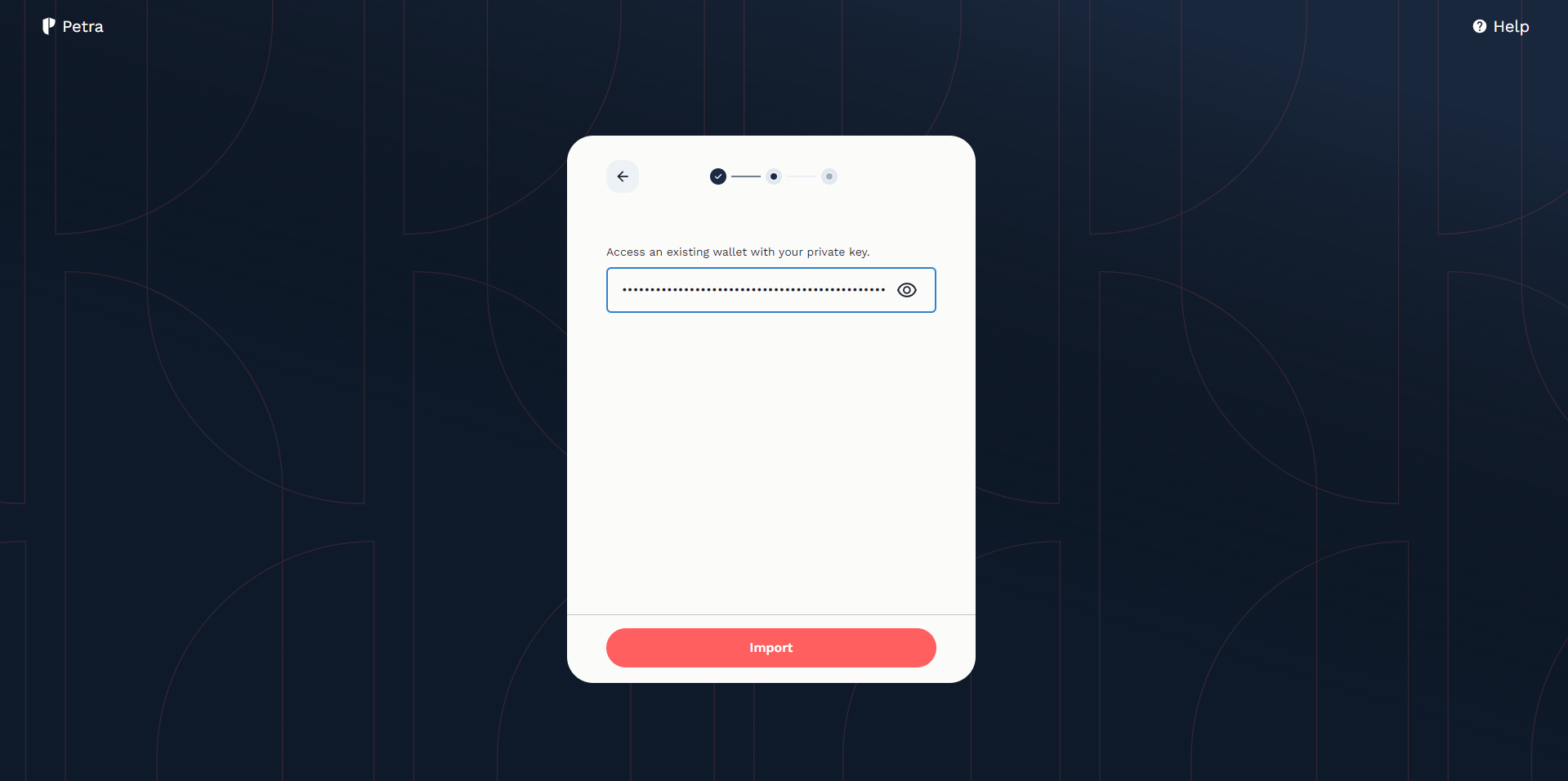
1. Open your web browser and navigate to the [Aptos Explorer](https://explorer.aptoslabs.com/?network=devnet).



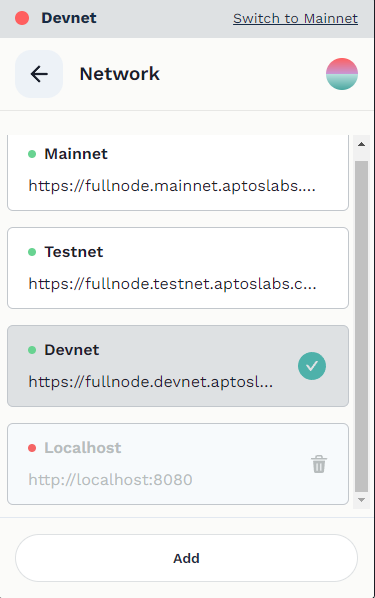
1. Switch network to Devnet (depends on the network you choose when creating your profile).

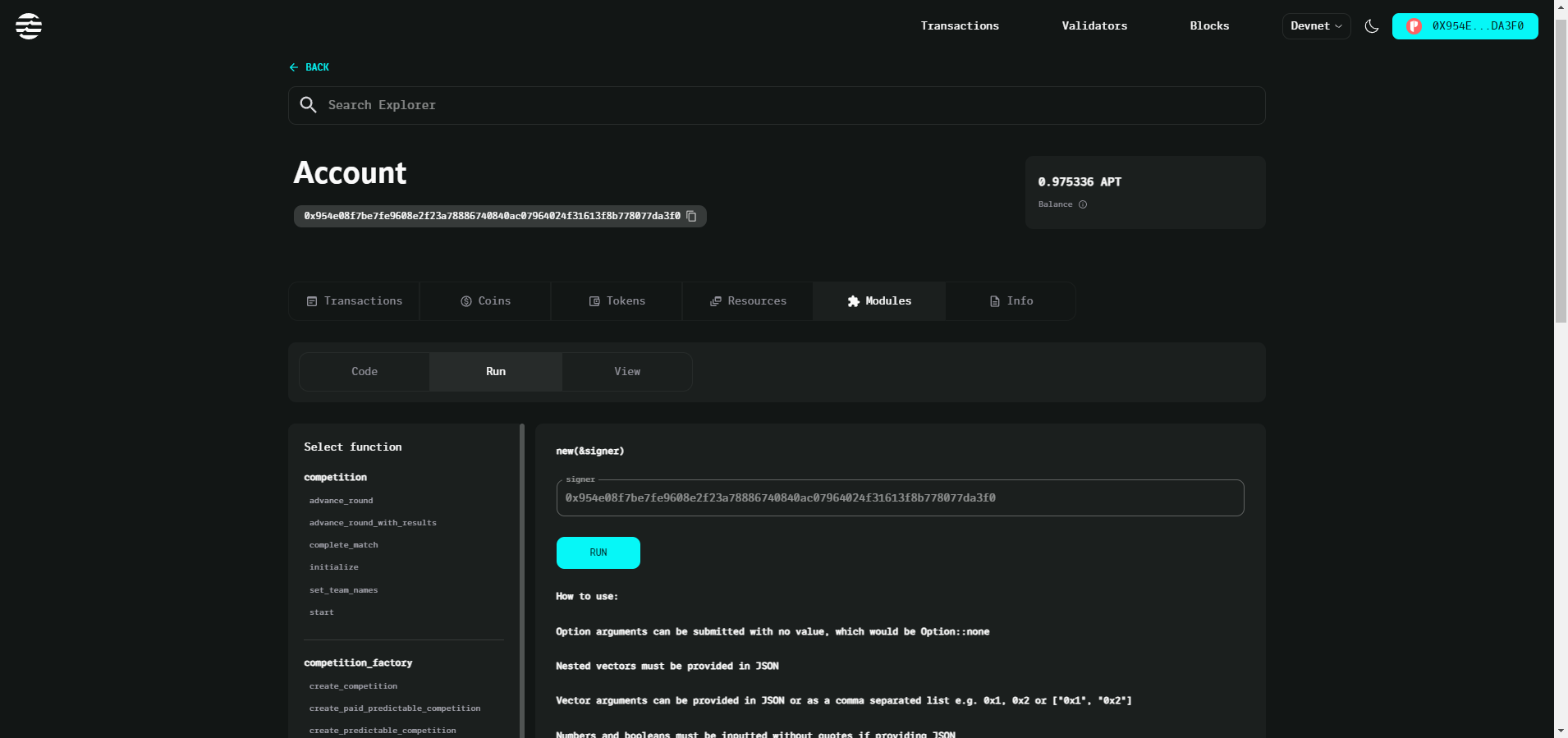


1. Import Private Key of your new account into Petra Wallet.



1. Connect to Petra wallet with your new account, switch network inside wallet to Devnet (depends on the network you choose when creating your profile).



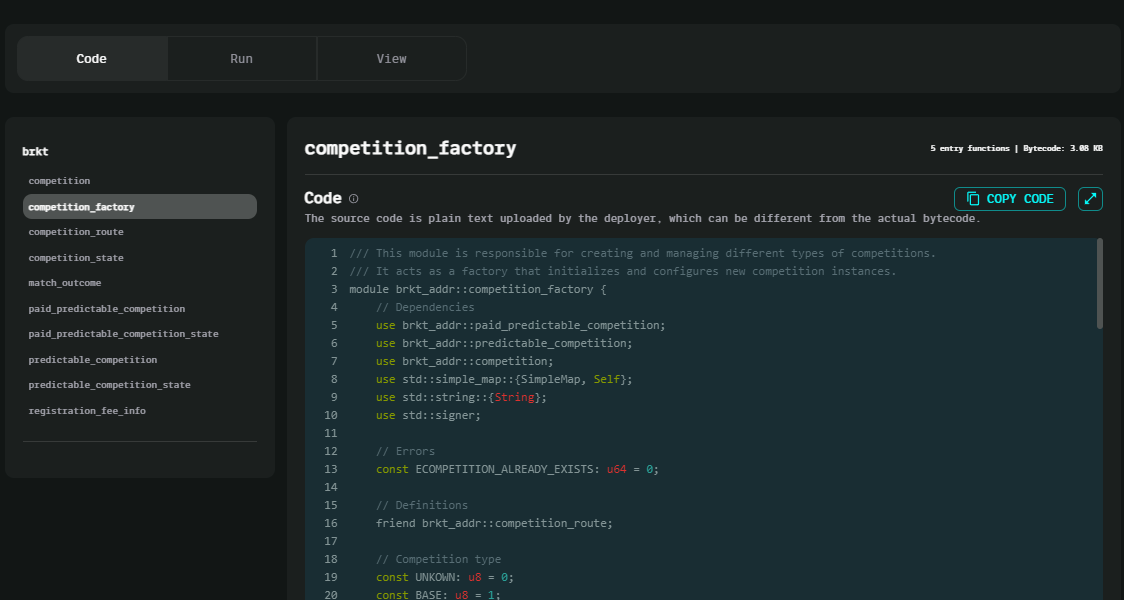
1. Search for Your Module to find all the functions in your module.

# INTERACT WITH THE SMART CONTRACT

## CODE SECTION

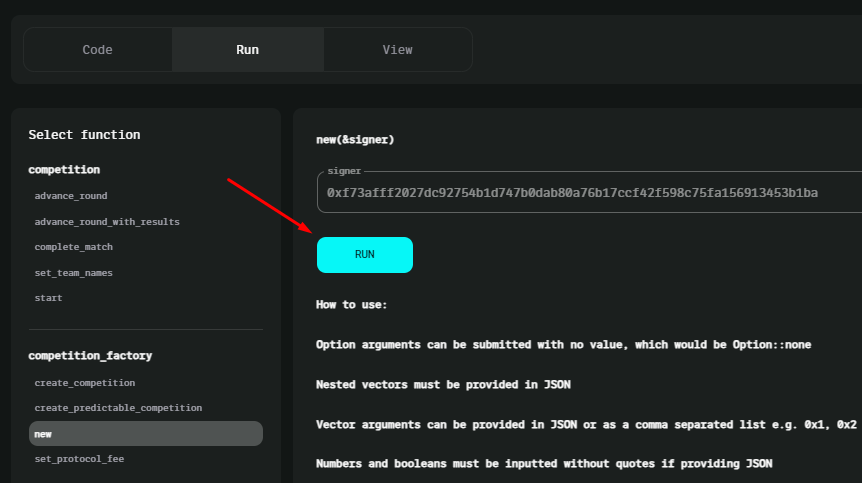
This section allows you to view and manage the source code of the smart contract. You can inspect the contract's code and verify its components.

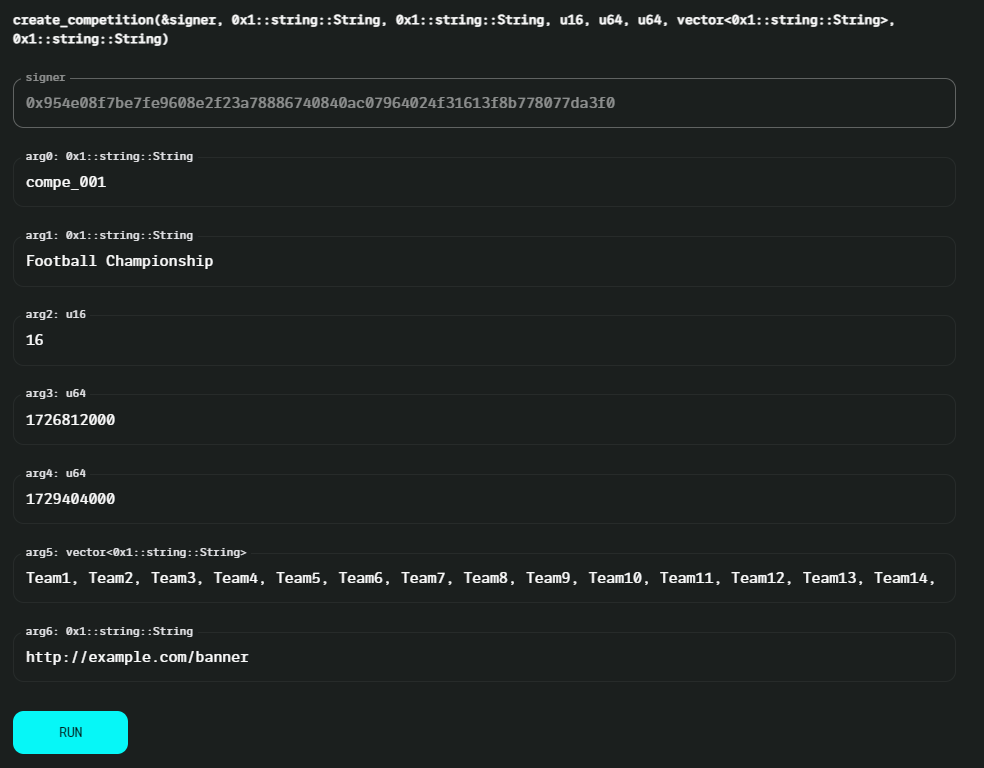
* **View Source Code:** Displays the source code of the deployed smart contract.



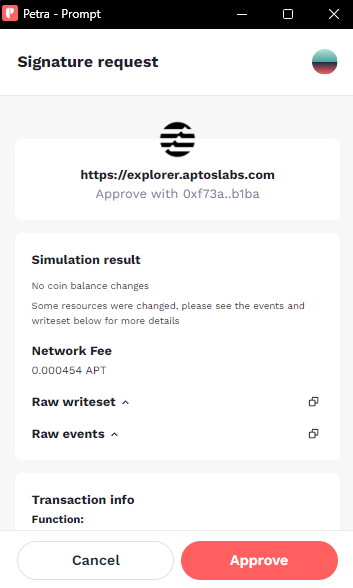
## RUN SECTION

This section allows you to execute commands or call functions from the smart contract. It provides a way to interact directly with the contract to test its functionalities.

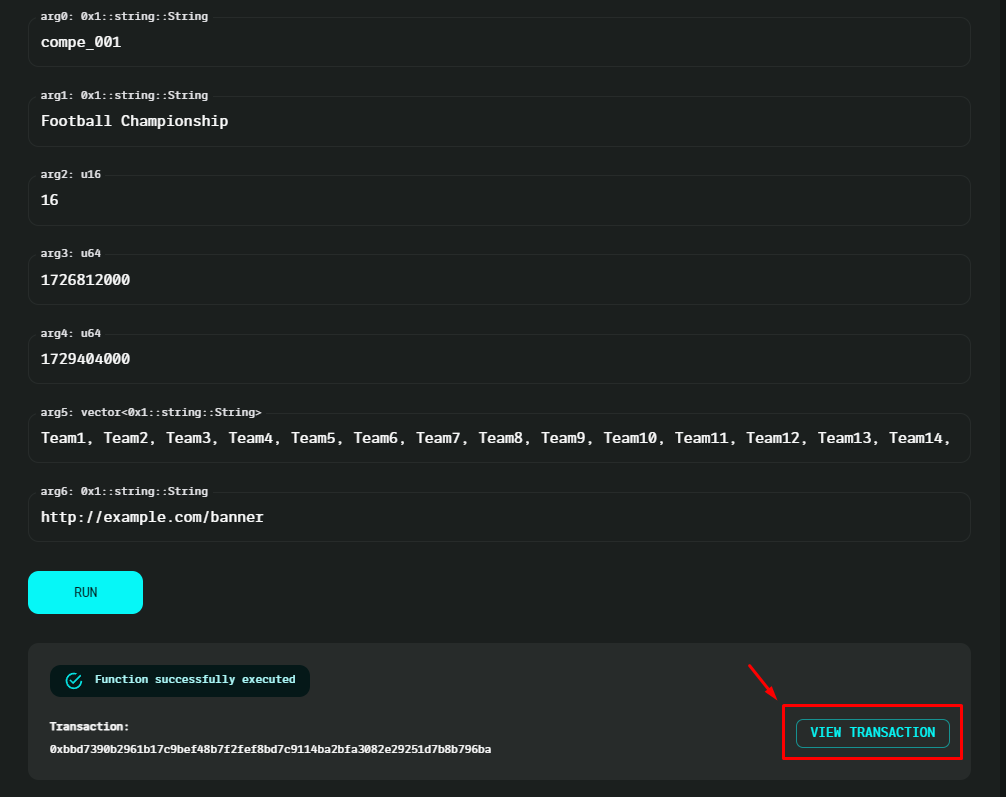
* Select the Module: Click on the module name to view its details.
* Navigate to the Functions Section: Within the module details, locate and click the “Run” tab or section to interact with functions in the smart contract.
* From the list of available functions, select the function you want to call.
* Input the required parameters for the function. This may include fields such as addresses, competition IDs, names, and other specific data.

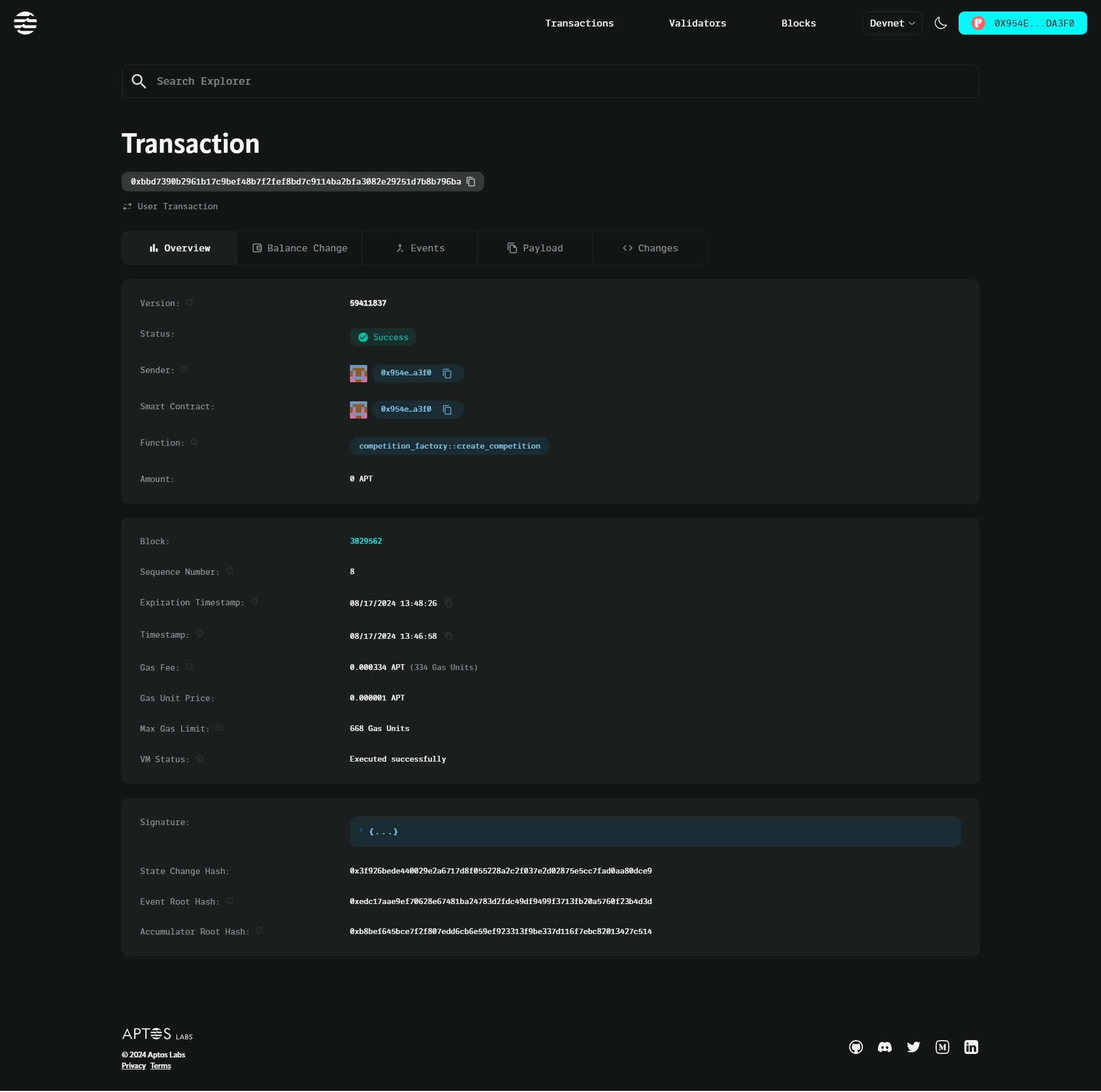


* After entering the parameters, click on the “Run” to call the function.
* Before executing the smart contract, an approval is required in Petra Wallet to authorize the transaction and confirm it on the Aptos blockchain. This ensures user consent.



* Review Transaction Status: Monitor the transaction status on Aptos Explorer to see if it has been processed successfully. Check for any errors or confirmations.

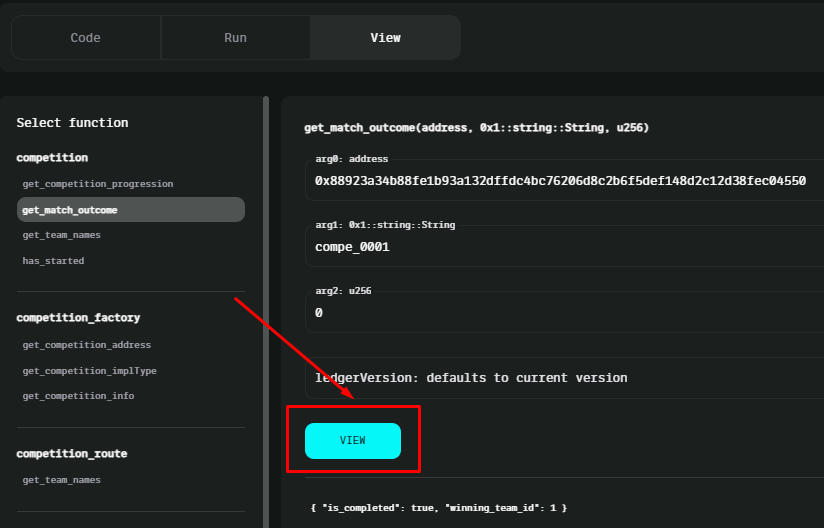




## VIEW SECTION

This section lets you view the current states and data of the smart contract. It provides insights into the contract's values and status.

* **View Contract State:** Displays the current state information of the contract, such as variable values by inputting data of the competition.



* **Track Events:** Provides an interface to monitor events emitted by the contract.





