

Here's a conversational script that simulates the interaction between a **User** and an **Agent**. The Agent guides the User through the MOR20 launch process based on the MOR20 Launch Testnet Guide:

Script: MOR20 Token Launch

Step 1: Start Interaction

Agent:

Hello! I'm here to assist you with launching your MOR20 token on the Ethereum and Arbitrum Sepolia testnets. Shall we get started?

User:

Yes, let's begin!

Step 2: Project Creation

Agent:

Great! First, name your project. What would you like to call it?

User:

Let's call it "MyTokenProject."

Agent:

Got it. "MyTokenProject" is set as your project name.

Now, what will the name of your token be?

User:

I'll call the token "MyToken."

Agent:

Understood. Your token will be named "MyToken." What about the token symbol? Traditionally, this is 3–4 characters (e.g., MOR).

User:

Let's use "MTK."

Agent:

Perfect. "MTK" will be your token symbol.

Would you like the contract to be upgradeable? Note, this choice is irreversible once deployed.

User:

Yes, make it upgradeable.

****Agent:****

Understood. The contract will be upgradeable.

Next, provide the Arbitrum wallet address or safe address that will administer this token and its associated contracts.

****User:****

Use this address: `0xArbitrumAdminAddress`.

****Agent:****

Address `0xArbitrumAdminAddress` has been set as the administrator.

Finally, enter the address of the yield token on Arbitrum Sepolia that will generate rewards (e.g., wstETH for testing).

****User:****

Use the test token address: `0xYieldTokenAddress`.

****Agent:****

Yield token address `0xYieldTokenAddress` has been recorded.

**Step 3: Trading Pair Configuration**

****Agent:****

Now, enter the address of the token that will serve as the trading pair (e.g., wETH for testing).

****User:****

Use the address `0xTradingPairTokenAddress`.

****Agent:****

Trading pair token address `0xTradingPairTokenAddress` has been set.

Next, set the swap fee percentage for the token pair. What fee would you like to configure?

****User:****

Let's use 0.3%.

****Agent:****

The swap fee is configured to 0.3%.

What fee would you like to set for the MOR20 pool itself?

****User:****

Let's use 0.2%.

****Agent:****

The pool fee is set to 0.2%. Moving on!

**Step 4: Pool Configuration**

****Agent:****

The MOR20 ecosystem relies on pools to manage token emissions and staking. Let's configure the first pool.

What is the name of the pool?

****User:****

Name it "Capital Pool."

****Agent:****

"Capital Pool" has been created. What is the starting emission amount for this pool?

****User:****

Set it to 3456 tokens.

****Agent:****

Starting emission amount is 3456 tokens. What is the daily emission decrease rate?

****User:****

Use 0.5925.

****Agent:****

The emission decrease rate is set to 0.5925.

What is the minimum stake amount for this pool?

****User:****

1 token.

****Agent:****

Minimum stake amount is 1 token. What is the lock period (in days) for staking?

****User:****

Set it to 90 days.

****Agent:****

The lock period is set to 90 days. Would you like to configure another pool?

****User:****

Yes, name it "Community Pool."

****Agent:****

“Community Pool” created! Repeat the same process for additional configurations.

**Step 5: Deployment**

****Agent:****

All pools are configured. Now, it's time to deploy your contracts.

Let's deploy to the Ethereum Sepolia testnet. Confirm your wallet is connected and switched to Ethereum Sepolia.

****User:****

It's connected. Let's deploy!

****Agent:****

Submitting the transaction... Please confirm the gas fees in your wallet.

****User:****

Confirmed.

****Agent:****

Deployment successful! Your MOR20 token contracts are live on the Ethereum Sepolia testnet. Next, let's deploy to the Arbitrum Sepolia testnet. Switch your wallet to Arbitrum Sepolia and confirm.

****User:****

Switched. Deploy now.

****Agent:****

Submitting the Arbitrum deployment... Please confirm the transaction in your wallet.

****User:****

Confirmed.

****Agent:****

Deployment successful! Your MOR20 contracts are live on both Ethereum and Arbitrum Sepolia testnets.

**Step 6: Testing and Finalization**

****Agent:****

Would you like to test staking and token emissions using the mock yield token (stETHmock)?

****User:****

Yes, let's test it.

****Agent:****

Navigate to the distribution contract, connect your wallet, and mint test tokens.

1. Approve the distribution contract with the amount to stake.
2. Stake your tokens into the pool.
3. Monitor your emissions using the ``getCurrentUserReward`` function.

****User:****

Done. It's working perfectly!

****Agent:****

Congratulations! Your MOR20 token is now live and fully functional. Let me know if there's anything else I can assist with!

This conversational flow simulates the entire MOR20 deployment process, incorporating all critical configurations and actions described in the guide. Let me know if you'd like this adjusted or integrated into a specific development framework!