Relay SDK with Viem

Getting Started with Relay SDK and Viem

This section is designed to assist developers in implementing the Relay SDK with Viem as an alternative to Ethers.

Installation

To start using the Relay SDK with Viem, you need to install the necessary packages:

npm yarn ```

Copy npminstall@gelatonetwork/relay-sdk-viem

Copy yarnadd@gelatonetwork/relay-sdk-viem

```

Configuration

Basic Configuration

Set up your project to use Viem by configuring the Relay SDK as follows:

```

Copy import{ GelatoRelay }from"@gelatonetwork/relay-sdk-viem"; import{ createPublicClient,http }from"viem"; import{ sepolia }from"viem/chains";

constclient=createPublicClient({ chain:sepolia, transport:http(), });

constrelay=newGelatoRelay(client);

```

Example Usage: Sponsored Call with ERC-2771

This example demonstrates how to use thesponsoredCallERC2771 method from the Relay SDK, which allows transactions to be sponsored using EIP-2771. Learn more about it atsponsoredCallERC2771

Prerequisites

Ensure your environment variables are set up by configuring your.env file:

```

Copy PRIVATE_KEY=your_private_key_here ALCHEMY_KEY=your_alchemy_key_here GELATO_RELAY_API_KEY=your_gelato_relay_api_key_here

```

```

Copy import{ CallWithERC2771Request, GelatoRelay, }from"@gelatonetwork/relay-sdk-viem"; import{ createWalletClient,http,encodeFunctionData,Hex }from"viem"; import{ privateKeyToAccount }from"viem/accounts"; import{ sepolia }from"viem/chains"; import*asdotenvfrom"dotenv"; import{ Contract,BytesLike }from"ethers"; dotenv.config({ path:".env"}); import{ counterAbi }from"../utils/counterAbi"; constprivateKey=process.env.PRIVATE_KEY;

consttestSponsoredCallERC2771=async()=>{ constrelay=newGelatoRelay();

constaccount=privateKeyToAccount(privateKeyasHex); constclient=createWalletClient({ account, transport:http(https://eth-sepolia.g.alchemy.com/v2{process.env.ALCHEMY_KEY}), }); console.log(account.address); constGELATO_RELAY_API_KEY=process.env.GELATO_RELAY_API_KEYasstring; constcounterAddress="0x00172f67db60E5fA346e599cdE675f0ca213b47b"; constchainId=awaitclient.getChainId();

//encode function data constdata=encodeFunctionData({ abi:counterAbi, functionName:"increment", });

constrelayRequest={ user:account.address, chainId:BigInt(chainId), target:counterAddress, data:dataasBytesLike, }asCallWithERC2771Request;

constresponse=awaitrelay.sponsoredCallERC2771(relayRequest, clientasany, GELATO_RELAY_API_KEY);
console.log(https://relay.gelato.digital/tasks/status{response.taskId}); };

testSponsoredCallERC2771();

```

This guide outlines the initial setup and demonstrates how to utilize the Relay SDK with Viem for sponsoring transactions. To explore more SDK methods and capabilities, please visit:

?
```