`@elizaos/plugin-bnb`

This plugin enables interaction with the BNB Chain ecosystem, providing support for BNB Smart Chain, opBNB, and BNB Greenfield networks.

Configuration

Default Setup

By default, **plugin-bnb** is not enabled. To use it, simply add your private key and/or public key to the `.env` file. If private key is not provided, some actions will be disabled.

Security Note: Your private key grants full access to your associated funds. Store it securely and never share it with anyone. Do not commit or upload your `.env` file to version control systems like Git.

```env
BNB_PRIVATE_KEY=your-private-key-here
BNB_PUBLIC_KEY=your-public-key-here
```

Custom RPC URLs

By default, the RPC URL is inferred from the `viem/chains` config. To use custom RPC URLs, add the following to your `.env` file:

```env
BSC_PROVIDER_URL=https://your-custom-bsc-rpc-url
BSC_TESTNET_PROVIDER_URL=https://your-custom-bsc-rpc-url
OPBNB_PROVIDER_URL=https://your-custom-opbnb-rpc-url
```

Provider

The **Wallet Provider** initializes with BSC as the default. It:

- Provides the **context** of the currently connected address and its balance.
- Creates **Public** and **Wallet clients** to interact with the supported chains.

Actions

Get Balance

Get the balance of an address on BSC. Just specify the:

- **Chain**
- **Address**
- **Token**

Example usage:

```bash
Get the USDC balance of 0x1234567890 on BSC.
Get the BNB balance of 0x1234567890 on BSC Testnet.
```

Transfer

Transfer tokens from one address to another on BSC/opBNB. Just specify the:

- **Chain**
- **Token**
- **Amount**
- **Recipient Address**

- **Data**(Optional)

Example usage:

```bash
Transfer 1 BNB to 0xRecipient on BSC.
Transfer 1 USDC to 0xRecipient on BSC.
```

Swap

Swap tokens from one address to another on BSC. Just specify the:

- **Chain**(Only BSC is supported for now)
- **Input Token**
- **Output Token**
- **Amount**
- **Slippage**(Optional)

Example usage:

```bash
Swap 0.001 BNB to USDC on BSC.
```

Bridge

Bridge tokens from one chain to another on BSC/opBNB. Just specify the:

- **From Chain**
- **To Chain**
- **From Token**
- **To Token**
- **Amount**
- **Recipient Address**(Optional)

Example usage:

```bash
Bridge 0.001 BNB from BSC to opBNB.
Withdraw 0.001 BNB from opBNB to BSC
```

Stake

Perform staking operations on BSC through [Lista Dao](https://lista.org/liquid-staking/BNB). User will receive sliBNB(0xB0b84D294e0C75A6abe60171b70edEb2EFd14A1B) as staking credit. Just specify the:

- **Chain**(Only BSC is supported for now)
- **Action**
- **Amount**

Example usage:

```bash
Deposit 0.0001 BNB to Lista Dao.
```

Deploy
Deploy ERC1155/ERC721/ERC20 smart contract
Example usage:

```bash
Deploy an ERC20 token with name 'ClydePepper', symbol ClydePepper, decimals 18, total supply 10000 on BSC
```

~~~~markdown
### Faucet

Request testnet tokens from the faucet. You could request any of the supported
tokens(BNB, BTC, BUSD, DAI, ETH, USDC). Just specify the:

- **Token**(Optional)
- **Recipient Address**

The faucet is rate-limited. One claim is allowed per IP address within a 24-hour period.
And the recipient address must maintain a minimum balance of 0.002 BNB on BSC Mainnet to
qualify.

**Example usage:**

```bash
Get some testnet USDC from the faucet.
```


### Greenfield

Support greenfield chain actions

- **createBucket**
- **uploadObject**
- **deleteObject**
- **crossChainTransfer**

**Example usage:**

- `create a bucket on greenfield TESTNET, and the name is test-eliza-2`

- `upload the attachment as an object to the greenfield, the bucket is "test-eliza-1" and
object name is "test.png"`

- `delete object test.png from bucket "test-eliza-1" on Greenfield Testnet`

- `crossChainTransfer 0.01 bnb from BSC testnet to Greenfield Testnet`
```




---

## Contribution

The plugin contains tests. Whether you're using **TDD** or not, please make sure to run
the tests before submitting a PR.

### Running Tests

Navigate to the `plugin-bnb` directory and run:

```bash
pnpm test
```
~~~~