

Creating a Monero wallet with the specified features involves several steps. Please note that interacting with cryptocurrency involves security risks, and it's crucial to follow best practices. Additionally, always be careful with private keys and sensitive information.

Below is a basic example using the Monero Python library (`monero-python`). Make sure to install the library first:

```
bash
```

```
pip install monero
```

Now, let's create a basic Monero wallet with the specified features:

```
python
```

```

from monero import Monero

class MoneroWallet:
    def __init__(self, config_file):
        # Load configuration from file
        self.config = self.load_config(config_file)

        # Initialize Monero instance
        self.monero = Monero()

    def load_config(self, config_file):
        # Implement a function to load configuration from a file
        # Example: read a JSON file with public and private keys
        # Return a dictionary with the configuration
        pass

    def get_balance(self):
        # Get the current wallet balance
        return self.monero.get_balance(address=self.config['public_address'])

    def send_transaction(self, recipient_address, amount):
        # Get the before transaction balance
        before_balance = self.get_balance()

        # Send the transaction
        tx_hash = self.monero.transfer(
            address=recipient_address,
            amount=amount,
            mixin=4, # mixin size for privacy
            unlock_time=0 # unlock time for the transaction
        )

        # Get the after transaction balance
        after_balance = self.get_balance()

        # Display transaction details
        print(f"Before transaction balance: {before_balance}")
        print(f"Transaction hash: {tx_hash}")
        print(f"After transaction balance: {after_balance}")

if __name__ == "__main__":
    # Example usage
    wallet = MoneroWallet("config.json") # Replace with your actual config file
    print(f"Current balance: {wallet.get_balance()} XMR")

    # Example transaction
    recipient_address = "RECIPIENT_ADDRESS" # Replace with the recipient's address
    amount_to_send = 1.0 # Replace with the amount to send
    wallet.send_transaction(recipient_address, amount_to_send)

```

In this example:

- The `MoneroWallet` class is initialized with a configuration file.
- The `load_config` method should be implemented to load the configuration from a file (e.g., JSON).
- The `get_balance` method retrieves the current wallet balance.
- The `send_transaction` method sends Monero to a recipient and displays the before and after transaction balances.

Ensure you handle errors, exceptions, and security considerations properly in a production environment. Always test thoroughly before using real funds.