

Bitcoinlib

Библиотека для работы с кошельками. Операции, связанные с кошельком:

- + Создание нового кошелька
- Экспорт данных о созданном кошельке
- Импорт существующего кошелька
- + Информация о кошельке
- Транзакция

Создание кошелька. Создается хранилище данных в ~/.bitcoinlib Затем можно проводить операции.

```
def create_wallet():
    # Создаем новый testnet кошелек
    wallet = Wallet.create(
        name='my_testnet_wallet',
        network='testnet'
    )

    print(f"Адрес: {wallet.get_key().address}")
    print(f"Приватный ключ (WIF): {wallet.get_key().wif}")
    print(f"Баланс: {wallet.balance()} satoshi")

    # Получить информацию об адресе
    print(f"Это testnet адрес? {wallet.get_key().address.startswith(('m', 'n', '2', 'tb1'))}")
```

Для получения стартовых btc в сети testnet использовал <https://coinfaucet.eu/en/btc-testnet/>

Информация о кошельке. Кошелек с данным названием уже установлен в системе.

```
def wallet_info():
    """Полная информация о кошельке (исправленная)"""

    wallet = Wallet('my_testnet_wallet')
    wallet.scan() # Важно: синхронизируем с сетью
```

```

print("=" * 60)
print(f"КОШЕЛЁК: {wallet.name}")
print(f"СЕТЬ: {wallet.network.name}")
print("=" * 60)

# Баланс
balance = wallet.balance()
print(f"\n    БАЛАНС: {balance:,} satoshi")
print(f"        ≈ {balance / 100000000:.8f} BTC")

# UTXOs
utxos = wallet.utxos()
print(f"\n UTXOs: {len(utxos)}")

if utxos:
    utxo_total = 0
    for i, utxo in enumerate(utxos, 1):
        print(f"\n  UTXO #{i}:")
        print(f"    Транзакция: {utxo['txid'][:20]}...:{utxo['output_n']}")
        print(f"    Адрес: {utxo['address']}")
        print(f"    Сумма: {utxo['value']:,} sat")

        if 'confirmations' in utxo:
            confs = utxo['confirmations']
            status = "✓ Подтверждено" if confs > 0 else "Ожидание"
            print(f"    Статус: {status} ({confs} подтверждений)")

        utxo_total += utxo['value']

    print(f"\n  Сумма всех UTXOs: {utxo_total:,} sat")

# Транзакции
transactions = wallet.transactions()
print(f"\n ТРАНЗАКЦИИ: {len(transactions)}")

if transactions:
    for tx in transactions:
        print(f"\n  Транзакция: {tx.txid[:20]}...")
        print(f"    Дата: {tx.date}")

```

```

if tx.confirmations:
    print(f"    Подтверждений: {tx.confirmations}")
else:
    print(f"    Статус: Неподтверждена")

print(f"    Комиссия: {tx.fee} sat")

# Анализируем сумму
our_addresses = wallet.addresslist()
received = 0
sent = 0

# Выходы (получение)
for output in tx.outputs:
    if output.address in our_addresses:
        received += output.value

# Входы (отправка)
for input_tx in tx.inputs:
    if input_tx.address in our_addresses:
        sent += input_tx.value

if received > 0 and sent > 0:
    print(f"    Тип: Перевод")
    print(f"    Изменение баланса: {received - sent:.,} sat")
elif received > 0:
    print(f"    Тип: Получение")
    print(f"    Сумма: +{received:.,} sat")
elif sent > 0:
    net_sent = sent - tx.fee
    print(f"    Тип: Отправка")
    print(f"    Сумма: -{net_sent:.,} sat (включая комиссию)")

# Ключи и адреса
print(f"\n КЛЮЧИ И АДРЕСА:")
keys = wallet.keys()
print(f"    Всего ключей: {len(keys)}")

used_addresses = [key.address for key in keys if key.used]
print(f"    Использованных адресов: {len(used_addresses)}")

```

```
# Показываем первые 5 адресов
for i, key in enumerate(keys[:5], 1):
    status = " Использован" if key.used else " Не использован"
    print(f" {i}. {key.address} - {status} ({key.balance} sat)")

if len(keys) > 5:
    print(f" ... и ещё {len(keys) - 5} адресов")

# Сетевая информация
print(f"\n СЕТЕВАЯ ИНФОРМАЦИЯ:")
#print(f" Последний блок: {wallet.last_block}")
print(f" ID кошелька: {wallet.wallet_id}")

return wallet
```

Revision #2

Created 2 February 2026 06:31:09 by Admin

Updated 2 February 2026 09:41:16 by Admin