

Blokzincir Geliştirici Sertifika Programı | Teknik Ödev 1 | Mustafa Mert Kısa

- Cüzdanın test ağına bağlanması ve uygulama aracılığıyla yeni adres oluşturulması.

```
JS main.js > ...
1  const { exec } = require('child_process');
2
3  //cüzdan oluşturuyoruz
4  const createWallet = () => new Promise((resolve, reject) => {
5    exec(`electrum --testnet create`, (err, data) => {
6      if (err) {
7        reject(err);
8      }
9
10     resolve(data);
11   });
12 });
13
14 //cüzdanımız mevcut ise yüklüyoruz
15 const loadWallet = () => new Promise((resolve, reject) => {
16   exec(`electrum --testnet load_wallet`, (err, data) => {
17     if (err) {
18       reject('Cüzdan yüklenirken bir hata oluştu!');
19     }
20
21     resolve(data);
22   });
23 });
24
25 //cüzdan işlemlerimiz tamamlandığında adres oluşturuyoruz
26 const createNewAddress = () => new Promise((resolve, reject) => {
27   exec(`electrum --testnet createnewaddress`, (err, data) => {
28     if (err) {
29       reject("Yeni adres oluşturulamadı!");
30     }
31
32     resolve(data);
33   });
34 });
35
36 const main = async () => {
37   const infoWallet = await loadWallet();
38
39   if (infoWallet) {
40     const newAddress = await createNewAddress();
41     console.log(`Yeni adresiniz kullanıma hazır: ${newAddress}`);
42   } else {
43     console.log('Lütfen cüzdan oluşturun!');
44     const newWallet = await createWallet();
45     console.log(`Yeni cüzdanınız kullanıma hazır: ${newWallet}`);
46   }
47
48   main();


```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
mstfmrt@mstfmrt:~/Desktop/Teknik0dev1$ node main.js
Password:
Yeni adresiniz kullanıma hazır: tblqdwmcgh5vcajjv24g99954jl6tx979h7c93vy5c2qptg7tsw54clsssgq30

mstfmrt@mstfmrt:~/Desktop/Teknik0dev1$
```

- Oluşturulan adrese test BTC'si gönderilmesi



Yet Another Bitcoin Testnet Faucet

The **Bitcoin Testnet Faucet** is quick!

The **Bitcoin Testnet Faucet** is bech32 ready, for modern clients!

No ads! No BS! No tracking stuff! Best dev experience EVER!

2 years up, more than 23,000 tBTC over >2,500,000 txs shipped, and counting...

Donate BTC: 37NFX8KWAQbaodUG6pE1hNUH1dXgkpbzY

Transaction sent

TxID:
27689a0669b540427f2ace56be721110ee339da4c87c7f0c34fd4c7473574d80
Address: tb1qzt3j0rfst2s30phc72s3hww9d2ma7cy228f8gm
Amount: 0.002

- Hesap bakiyesinin uygulama aracılığıyla sorgulanması

```
36 //hesabımızın bakiyesini sorguluyoruz
37 const getBalance = () => new Promise((resolve, reject) => {
38   exec(`electrum --testnet getbalance`, (err, data) => {
39     if (err) {
40       reject("Bakiye sorgulanırken bir hata oluştu!");
41     }
42
43     resolve(data);
44   });
45 });
46
47 const main = async () => {
48   const infoWallet = await loadWallet();
49
50   if (infoWallet) {
51     const newAddress = await createNewAddress();
52     console.log(`Yeni adresiniz kullanıma hazır: ${newAddress}`);
53   } else {
54     console.log('Lütfen cüzdan oluşturun!');
55     const newWallet = await createWallet();
56     console.log(`Yeni cüzdanınız kullanıma hazır: ${newWallet}`);
57   }
58   const infoBalance = await getBalance();
59   console.log(`Hesabınızın bakiyesi: ${infoBalance}`);
60 }
61
62 main();
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
mstfmr@stfmr:~/Desktop/Teknik0dev1$ node main.js
Password:
Yeni adresiniz kullanıma hazır: tblqzs4lkrrefnd3k25pal2np9z99qp83gc7azj70lnd8wa8pdz9585qthk2aw

Hesabınızın bakiyesi: {
  "confirmed": "0.002"
}

mstfmr@stfmr:~/Desktop/Teknik0dev1$
```

- Uygulama aracılığı ile BTC gönderimi yapılması

```
47 //BTC gönderimi yapıyoruz, hash kodunu alıyoruz ve işlemi ağ ile paylaşıyoruz
48 const payto = (address, amount) => new Promise((resolve, reject) => {
49   exec(`electrum --testnet payto ${address} ${amount}`, (err, data) => {
50     if (err) {
51       reject(err);
52     }
53
54     resolve(data);
55   });
56 });
57
58 const broadcast = (bytecode) => new Promise((resolve, reject) => {
59   exec(`electrum --testnet broadcast ${bytecode}`, (err, data) => {
60     if (err) {
61       reject(err);
62     }
63
64     resolve(data);
65   });
66 });
67
68 //Gönderim adresi ve miktarı
69 const destinationAddress = 'mkHS9ne12qx9pS9VojpwU5xtRd4T7X7Zut';
70 const amount = 0.00000001;
71
72 const byteCode = await payto(destinationAddress, amount);
73 console.log(`Bytecode: ${byteCode}`);
74
75 if (byteCode) {
76   const transactionId = await broadcast(byteCode);
77   console.log(`İşlem numarası: ${transactionId}`);
78 }
79
80 }
81
82 main();
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
13: bash
Yeni adresiniz kullanıma hazır: tblqfj2k5shqw7f5su79x4llv78srntszp7k96pnjt

Hesabınızın bakiyesi: {
  "confirmed": "0.002"
}

Password:
Bytecode: 02000000000101804d5773744cfd340c7f7cc8a49d33ee101172be56ce2a7f4240b569069a68270100000000fdffffff0201000000000000001976a914344a0f48ca150ec2b903817660b9
b68b13a6702688acaf0c0300000000001600144d0a48dad0837518474ff5ba9133bf36c922aed802473044022002718e7178f8051a3c925d8d6d05781086a11ba01db2a49cf204d43686e24c0102206f
eb1185e44f5bd598ad0f256f1703494736c755c3f33167d036ffe18d97eeb2012103c46a95d23994ebfcc89b4237bdf70fda9451c7dca5e00fc618d6771da69b68541071e00

İşlem numarası: 5e0460d06f870a6d337ce081f9a60f540401f9f8d234ad694caa88237f5ab460
```

İşlem Numarası: [5e0460d06f870a6d337ce081f9a60f540401f9f8d234ad694caa88237f5ab460](#)







Özet ⓘ

USD BTC

Transaction 5e0460d06f870a6d337ce081f9a60f540401f9f8d234ad694caa88237f5ab460 was first broadcast to the Bitcoin network on Mart 28, 2021 at 12:45 ÖÖ GMT +03:00. The transaction currently has 4 confirmations on the network.

At the time of this transaction, 0.00199856 BTC was sent with a value of \$111.98. The current value of this transaction is now \$111,45.

Learn more about [how transactions work](#).

Hash	5e0460d06f870a6d337ce081f9a60f540401f9f8d234ad694caa... 	2021-03-28 00:45
	tb1qzt3j0rfst2s30phc72s3hvv9d2ma7cy228f... 0.00200000 BTC 	
	mkHS9ne12qx9pS9VojpwU5xtRd4T7X7ZUt 0.00000001 BTC 	
	tb1qf59y3kkssd63s3607kafzvalxmyj9tkc6drkft 0.00199855 BTC 	
Ücret	0.00000144 BTC (0.640 sat/B - 0.251 sat/WU - 225 bytes)	0.00199856 BTC 
		4 Onaylar

UYGULAMANIN KODLARI

```
const { exec } = require('child_process');

//cüzdan oluşturuyoruz
const createWallet = () => new Promise((resolve, reject)
=> {
  exec(`electrum --testnet create`, (err, data) => {
    if (err) {
      reject(err);
    }

    resolve(data);
  });
});

//cüzdanımız mevcut ise yüklüyoruz
const loadWallet = () => new Promise((resolve, reject) =
> {
  exec(`electrum --
testnet load_wallet`, (err, data) => {
    if (err) {
      reject('Cüzdan yüklenirken bir hata oluştu!');
    }

    resolve(data);
  });
});
```

```

    });
  });

  //cüzdan işlemlerimiz tamamlandığında adres oluşturuyoruz
  const createNewAddress = () => new Promise((resolve, reject) => {
    exec(`electrum --testnet createnewaddress`, (err, data) => {
      if (err) {
        reject("Yeni adres oluşturulamadı!");
      }

      resolve(data);
    });
  });

  //hesabımızın bakiyesini sorguluyoruz
  const getBalance = () => new Promise((resolve, reject) => {
    exec(`electrum --testnet getbalance`, (err, data) => {
      if (err) {
        reject("Bakiye sorgulanırken bir hata oluştu!");
      }

      resolve(data);
    });
  });

  //BTC gönderimi yapıyoruz, hash kodunu alıyoruz ve işlemi ağ ile paylaşıyoruz
  const payto = (address, amount) => new Promise((resolve, reject) => {
    exec(`electrum --testnet payto ${address} ${amount}`, (err, data) => {
      if (err) {
        reject(err);
      }
    });
  });

```

```

        resolve(data);
    });
});

const broadcast = (bytecode) => new Promise((resolve, reject) => {
    exec(`electrum --
testnet broadcast ${bytecode}`, (err, data) => {
        if (err) {
            reject(err);
        }

        resolve(data);
    });
});

const main = async () => {
    const infoWallet = await loadWallet();

    if (infoWallet) {
        const newAddress = await createNewAddress();
        console.log(`Yeni adresiniz kullanıma hazır: ${newAddress}`);
    } else {
        console.log('Lütfen cüzdan oluşturun!');
        const newWallet = await createWallet();
        console.log(`Yeni cüzdanınız kullanıma hazır: ${newWallet}`);
    }

    const infoBalance = await getBalance();
    console.log(`Hesabınızın bakiyesi: ${infoBalance}`);

    //Gönderim adresi ve miktarı
    const destinationAddress = 'mkHS9ne12qx9pS9VojpwU5xtRd4T7X7ZUt';
    const amount = 0.00000001;

```

```
    const byteCode = await payto(destinationAddress, amount);
    console.log(`Bytecode: ${byteCode}`);

    if (byteCode) {
        const transactionId = await broadcast(byteCode);
        console.log(`İşlem numarası: ${transactionId}`);
    }
}

main();
```