LimeChain

Telerik Academy

Decentralised Application Development - Part 2

Telerik Academy

Creating instances of wallet using ethers.js

Telerik Academy

Signing transactions with private key

Ethers.js Wallet with private key

Telerik Academy

```
// create wallet from private key
const wallet = new ethers.Wallet(this.privateKey, this.infuraProvider);
const connectedContract = this.billboardContract.connect(wallet);
const sentTransatcion = await connectedContract.buy(this.newSlogan, { value: 100, gasPrice: 20000000000, gasLimit: 4700000 });
const transactionResult = await wallet.provider.waitForTransaction(sentTransatcion.hash);
const transactionReceipt = await wallet.provider.getTransactionReceipt(sentTransatcion.hash);
alert('we are done');
```

Telerik Academy

Signing transactions with mnemonic

Ethers.js Wallet with mnemonic

Telerik Academy

```
// create wallet from mnemonic
const initialWallet = ethers.Wallet.fromMnemonic(this.mnemonic);
const wallet = initialWallet.connect(this.infuraProvider);
const connectedContract = this.billboardContrat.connect(wallet);
const sentTransatcion = await connectedContract.buy(this.newSlogan, { value: 100, gasPrice: 20000000000, gasLimit: 4700000 });
const transactionResult = await wallet.provider.waitForTransaction(sentTransatcion.hash);
const transactionReceipt = await wallet.provider.getTransactionReceipt(sentTransatcion.hash);
alert('we are done');
```

Telerik Academy

Signing transactions with JSON + pass

Ethers.js Wallet with JSON + pass

```
// decrypt created wallet from encrypted json and password
const initialWallet = await ethers.Wallet.fromEncryptedJson(this.jsonContent, this.password, callback);

function callback(progress) {
  console.log('Decrypting: ' + progress * 100 + '% complete');
}
const wallet = initialWallet.connect(this.infuraProvider);
const connectedContract = this.billboardContract.connect(wallet);
const sentTransatcion = await connectedContract.buy(this.newSlogan, { value: 100, gasPrice: 20000000000, gasLimit: 4700000 });
const transactionResult = await wallet.provider.waitForTransaction(sentTransatcion.hash);
const transactionReceipt = await wallet.provider.getTransactionReceipt(sentTransatcion.hash);
alert('we are done');
```

Telerik Academy

UX Pro – Creating user wallet and caching the JSON

Creating wallet. Converting it to JSON file. Cache the JSON in the browser.

Telerik Academy

```
// created wallet from encrypted json with password, that are saved in local storage
const json = window.localStorage.getItem('wallet');
const initialWallet = await ethers.Wallet.fromEncryptedJson(json, this.password, callback);

function callback(progress) {
  console.log('Decrypting: ' + progress * 100 + '% complete');
}
const wallet = initialWallet.connect(this.infuraProvider);
const connectedContract = this.billboardContract.connect(wallet);
const sentTransatcion = await connectedContract.buy(this.newSlogan, { value: 51, gasPrice: 20000000000, gasLimit: 4700000 });
const transactionResult = await wallet.provider.waitForTransaction(sentTransatcion.hash);
const transactionReceipt = await wallet.provider.getTransactionReceipt(sentTransatcion.hash);
alert('we are done');
```

Telerik Academy

Download Wallet

Download wallet from local storage

Telerik Academy

```
public downloadOldJSONFile() {
  const json = window.localStorage.getItem('wallet')
  const downloader = document.createElement('a');
  document.body.appendChild(downloader); // Needed for ff;

  const data = JSON.stringify(json);
  const blob = new Blob([data], { type: 'text/json' });
  const url = window.URL;
  const fileUrl = url.createObjectURL(blob);

  downloader.setAttribute('href', fileUrl);
  downloader.setAttribute('download', 'pro-wallet-backup.json');
  downloader.click();
}
```

Telerik Academy

Further reading

- Ethers.js Documentation - https://docs.ethers.io/ethers.js/html/index.html

Telerik Academy

Homework

1. *Extend your crypto cars contract to list all cars bought. Add a list of all bought cars with owner.*
2. *Add a way for the user to create wallet (json + pass) and backup their json file*
3. *Add a way for the user to buy a new car*
4. *Add a way for the user to buy an existing car*