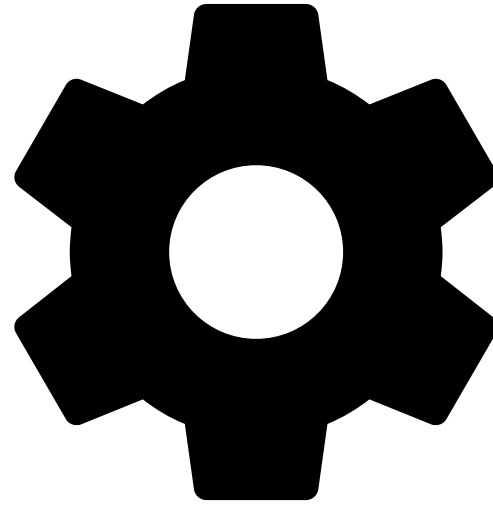
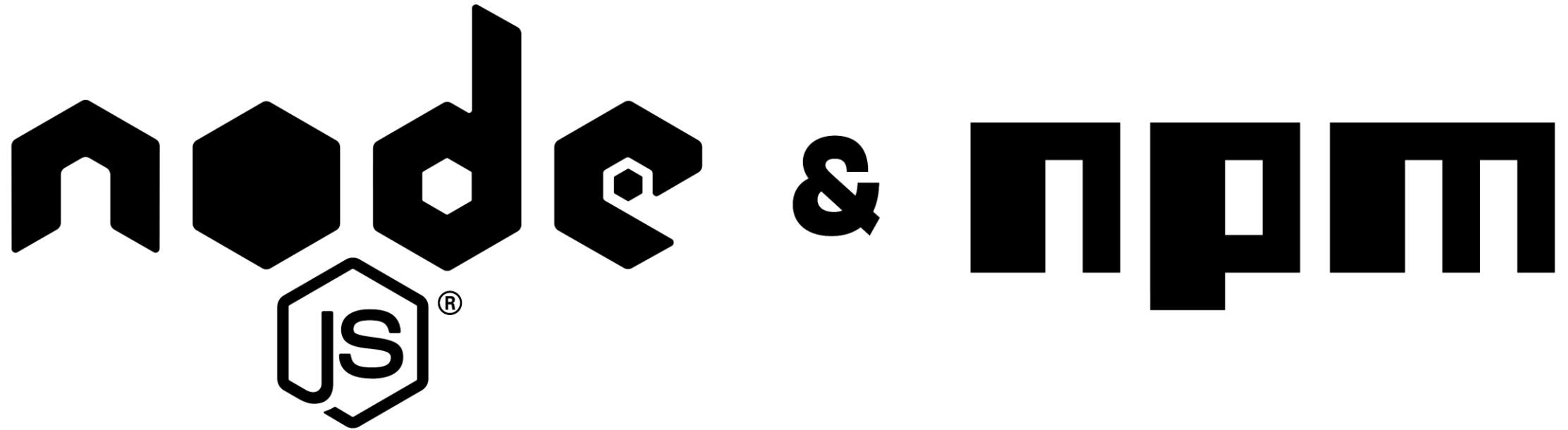


GETTING STARTED



Setup steps for EasyFHE usage

PRE-REQUIREMENTS



<https://nodejs.org/en/download/>

INSTALLING

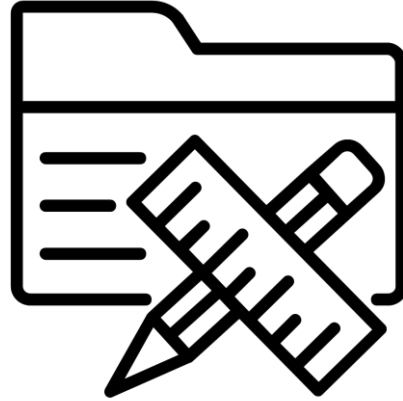


Quasar

Open the terminal and type the following command:

```
npm i -g @quasar/cli
```

CREATING THE QUASAR PROJECT



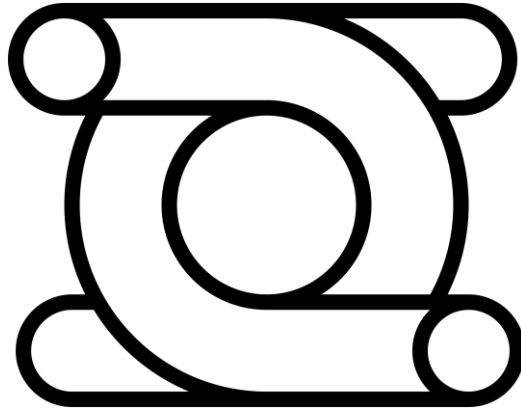
Open the terminal and type the following command:

```
npm init quasar
```

Go in '*quasar.config.js*' and add to the build property the following:

```
extendWebpack(cfg) {  
  cfg.experiments = {  
    asyncWebAssembly: true,  
  };  
},
```

CREATING THE QUASAR PROJECT



Open the terminal and type the following command:

```
git clone https://github.com/mihailpreda/fhe-module-typescript-wrapper.git easyFHE
```

```
cd easyFHE
```

```
npm install
```

Then go to Quasar project's *'package.json'* file and add EasyFHE to dependencies like:

```
easyFHE : file:../easyFHE
```

USAGE EXAMPLE

```
1 <script lang='ts'>
2 import getEasyFHE, {
3   EasyScheme,
4   EasySecurity,
5   EasySpeed,
6   EasyFHE,
7 } from 'easyFHE';
8 (async () => {
9   // Get the WebAssembly module
10  const easyFhe: EasyFHE = await getEasyFHE();
11  // Initialize it
12  await easyFhe.Setup.initialize();
13  // Set your preferred encryption parameters
14  easyFhe.Setup.fastSetup(
15    EasyScheme.BFV,
16    EasySecurity.TC128,
17    EasySpeed.FAST,
18  );
19  // Generate a keypair
20  const [publicKey, secretKey] = easyFhe.generateKeys();
21  // ciphertext + ciphertext
22  //-----
23  const leftValue0 = easyFhe.encrypt(Int32Array.from([3]), publicKey);
24  const rightValue0 = easyFhe.encrypt(Int32Array.from([6]), publicKey);
25  const encResult0 = easyFhe.Cipher.add(leftValue0.save(), rightValue0.save());
26  const result0 = easyFhe.decrypt(encResult0.save(), secretKey);
27  //-----
28  easyFhe.deallocateLibrary();
29  publicKey.delete();
30  secretKey.delete();
31 }())
32 </script>
```

CONTACT



Preda Mihail Irinel

mihaipreda1997@gmail.com