

day47-rn-web3

☰ 태그	
📅 날짜	@2022년 12월 6일

```
PS C:\reactnative\expoweb3> expo init
WARNING: expo-cli has not yet been tested against Node.js v18.12.0.
If you encounter any issues, please report them to https://github.com/expo/expo-cli/issues

expo-cli supports following Node.js versions:
* >=12.13.0 <15.0.0 (Maintenance LTS)
* >=16.0.0 <17.0.0 (Active LTS)

Migrate to using:
> npx create-expo-app --template

? What would you like to name your app? > expoweb3
```

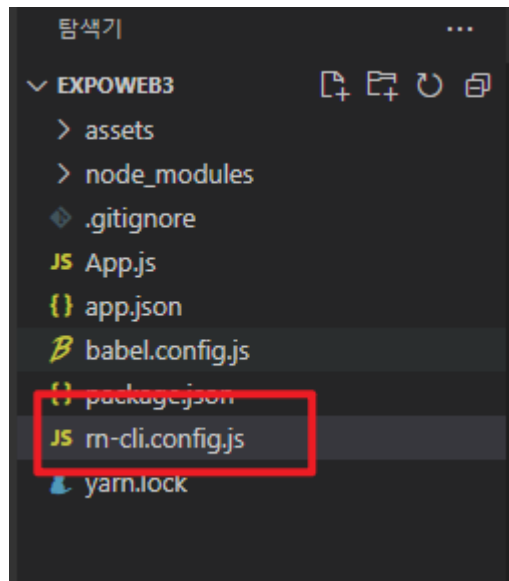
expo init

```
문제   출력   디버그 콘솔   터미널

Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\reactnative\expoweb3\expoweb3>npm install --save node-libs-browser
```

설치



rn-cli.config.js

```
시작하기 x JS rn-cli.config.js
JS rn-cli.config.js > ...
1  const nodeLibs = require('node-libs-browser');
2
3  module.exports = {
4    resolver: {
5      extraNodeModules: nodeLibs
6    }
7  };
```

```
JS m-cli.config.js JS global.js X JS App.js
JS global.js > ...
1 // Inject node globals into React Native global scope.
2 global.Buffer = require('buffer').Buffer;
3 global.process = require('process');
4
5 if (typeof btoa === 'undefined') {
6   global.btoa = function (str) {
7     return new Buffer(str, 'binary').toString('base64');
8   };
9 }
10
11 if (typeof atob === 'undefined') {
12   global.atob = function (b64Encoded) {
13     return new Buffer(b64Encoded, 'base64').toString('binary');
14   };
15 }
```

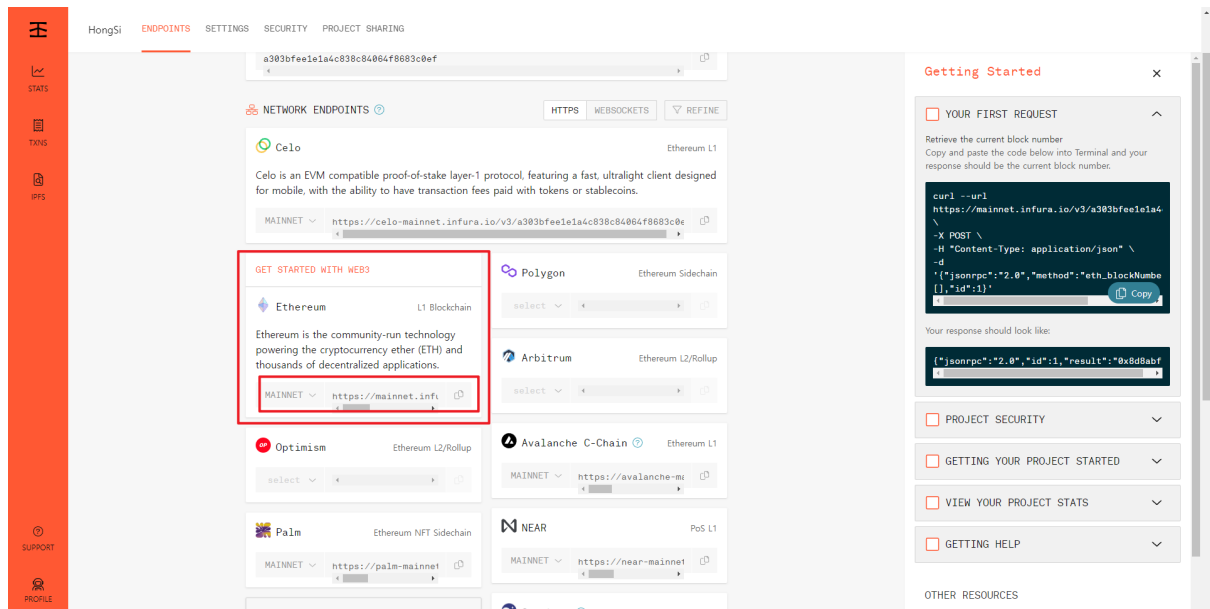
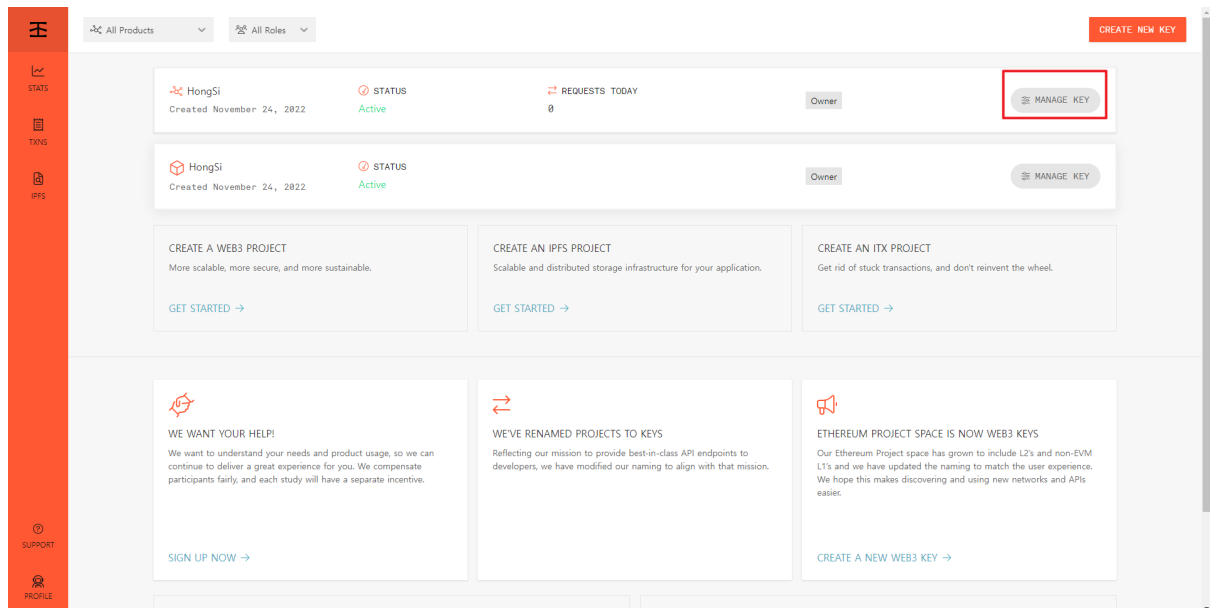
```
문제 출력 디버그 콘솔 터미널
54 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\reactnative\expoweb3\expoweb3>npm install --save web3
```

```
JS rn-cli.config.js JS global.js JS App.js
JS App.js > [⌘] styles > container
1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3  import './global';
4  import Web3 from 'web3';
5
6  export default function App() {
7    return (
8      <View style={styles.container}>
9        <Text>Open up App.js to start working on your app!</Text>
10       <StatusBar style="auto" />
11     </View>
12   );
13 }
14
15 const styles = StyleSheet.create({
16   container: {
17     flex: 1,
18     backgroundColor: '#fff',
19     alignItems: 'center',
20     justifyContent: 'center',
21   },
22 });
23
```

```
JS rn-cli.config.js JS global.js JS App.js X
JS App.js > App
1 import { StatusBar } from 'expo-status-bar';
2 import { StyleSheet, Text, View } from 'react-native';
3 import './global';
4 import Web3 from 'web3';
5 import { useEffect } from 'react';
6
7 export default function App() {
8
9   //초기에 한번 함수 동작
10  useEffect(() => {
11    const web3 = new Web3();
12
13  }, [])
14
15  return (
16    <View style={styles.container}>
17      <Text>Open up App.js to start working on your app!</Text>
18      <StatusBar style="auto" />
19    </View>
20  );
21 }
22
23 const styles = StyleSheet.create({
24   container: {
25     flex: 1,
26     backgroundColor: '#fff',
27     alignItems: 'center',
28     justifyContent: 'center',
29   },
30 });
31
```



고를리로 바꿀것

노드를 연결 시켜주는 연결점

메타마스크로 연결하는게 아닌 노드에 직접 키가 필요한게 아닌 고를리라는 네트워크에 직접연결

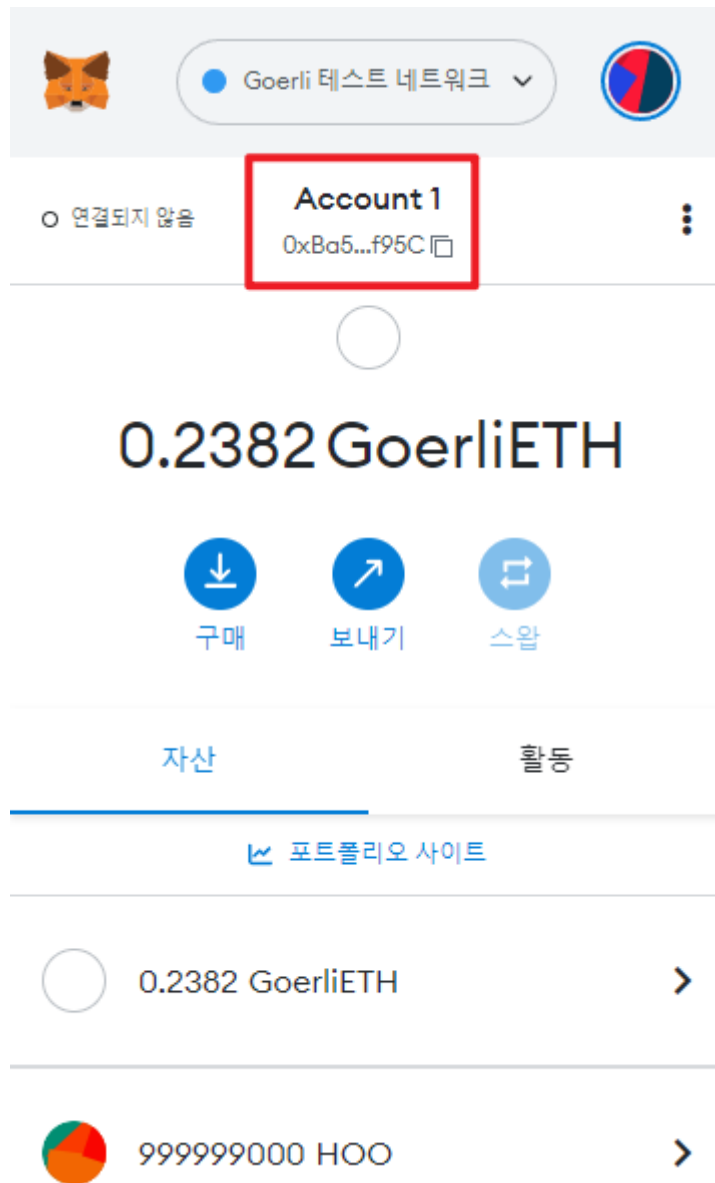
<https://mainnet.infura.io/v3/a303bfee1e1a4c838c84064f8683c0ef>

```
web3.setProvider(new web3.providers.HttpProvider('https://mainnet.infura.io/v3/a303bfbee1e1a4c838c84064f8683c0ef'));
web3.eth.getBlock('latest').then(console.log)
},[])
```

```
03>npm start
```

[illegible]

```
JS rn-cli.config.js JS global.js JS App.js
JS App.js > App > useEffect() callback
1 import { StatusBar } from 'expo-status-bar';
2 import { StyleSheet, Text, View } from 'react-native';
3 import './global';
4 import Web3 from 'web3';
5 import { useEffect, useState } from 'react';
6
7 export default function App() {
8   //자신의 계정 잔고조회
9   const [web3, setWeb3] = useState();
10  const [balance, setBalance] = useState();
11
12  //초기에 한번 함수 동작
13  useEffect(() => {
14    const web3 = new Web3();
15    web3.setProvider(new web3.providers.HttpProvider('https://goerli.infura.io/v3/a303bfee1e1a4c838d...'));
16    setWeb3(web3);
17    getBalance();
18  }, []);
19  //동시화 async await
20  const getBalance = async () => {
21    let balance = await web3.eth.getBalance("0xBa502C767f3F5c25C26e8626e57A906b5141f95C");
22    console.log(balance);
23  };
24
25  return (
26    <View style={styles.container}>
27      <Text>Open up App.js to start working on your app!</Text>
28
29    <StatusBar style="auto" />
```

```

export default function App() {
  //자신의 계정 잔고조회
  const [web3, setWeb3] = useState();
  const [balance, setBalance] = useState();

  //초기에 한번 함수 동작
  useEffect(() => {
    const web3 = new Web3();
    web3.setProvider(new web3.providers.HttpProvider('https://goerli.infura.io/v3/a303bfee1e1a4c838c84064f8683c0ef'));
    setWeb3(web3);
    getBalance();
  }, []);
  //동시화 async await
  const getBalance = async () => {
    let balance = await web3.eth.getBalance("0xBa502C767f3F5c25C26e8626e57A906b5141f95C");
    balance = web3.utils.fromWei(balance, 'ether');
    console.log(balance);
    setBalance(balance);
  };

  return (
    <View style={styles.container}>
      <Text>내 잔고 {balance}</Text>

      <StatusBar style="auto" />
    </View>
  );
}

```