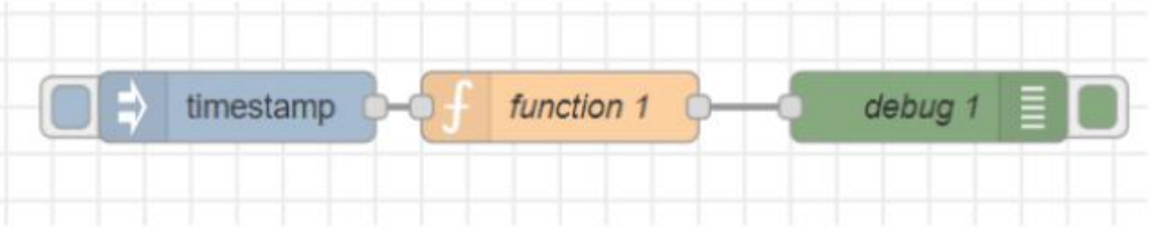
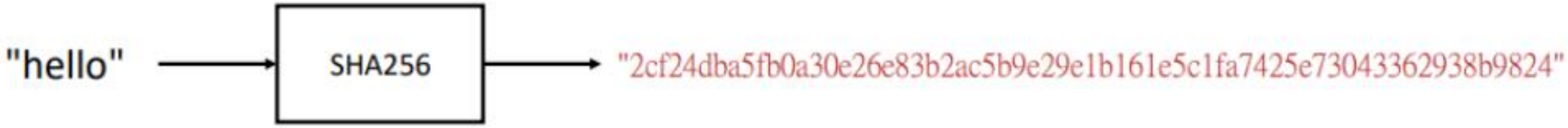
物聯網實務 HW14

電機碩一 11278008 林佳慧

日期:2023/12/20

Exercise 14-1

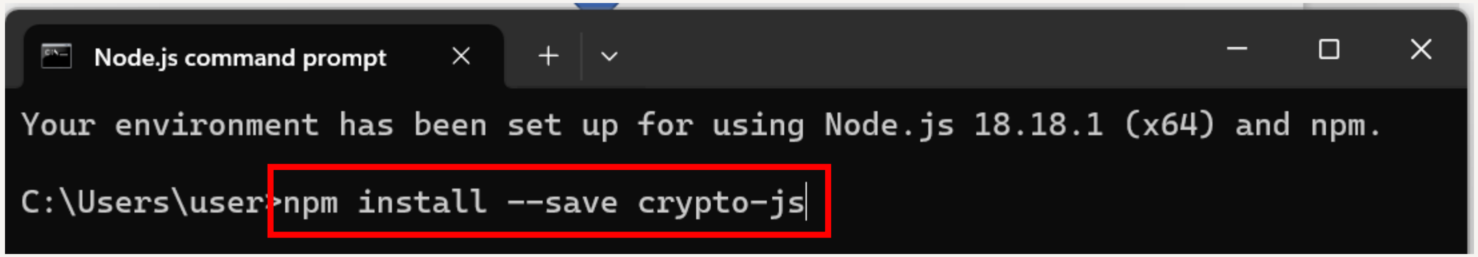




"hello" → SHA256 → "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824"

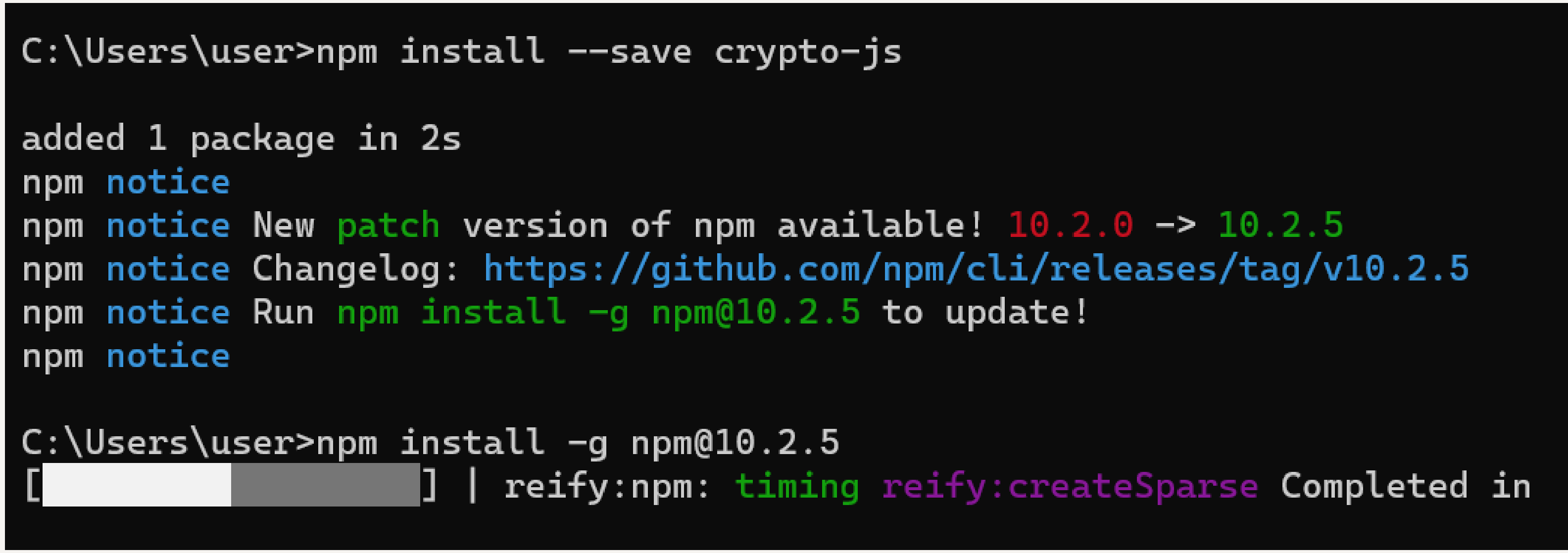
Install crypto-js

npm install –save crypto-js



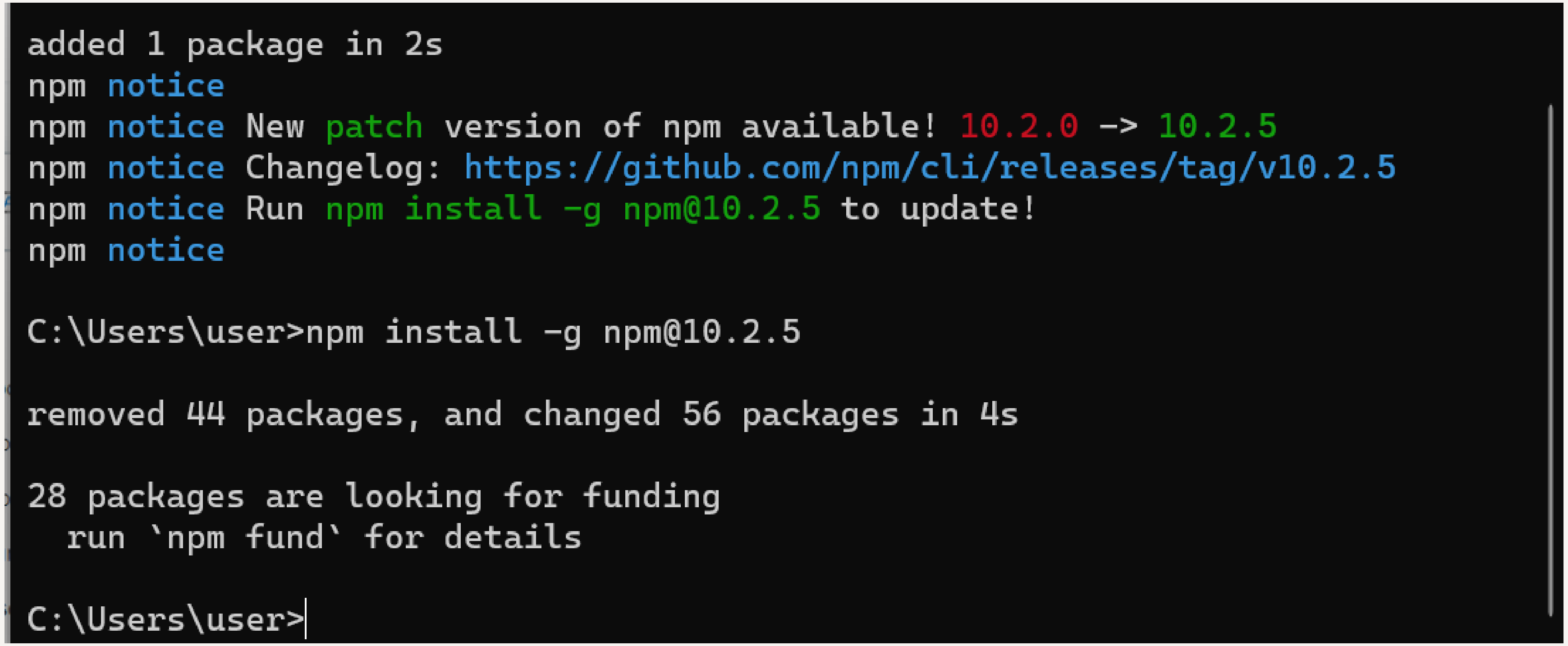
```
Your environment has been set up for using Node.js 18.18.1 (x64) and npm.

C:\Users\user>npm install --save crypto-js
```

Use “npm install –g npm@10.2.5 “ to update



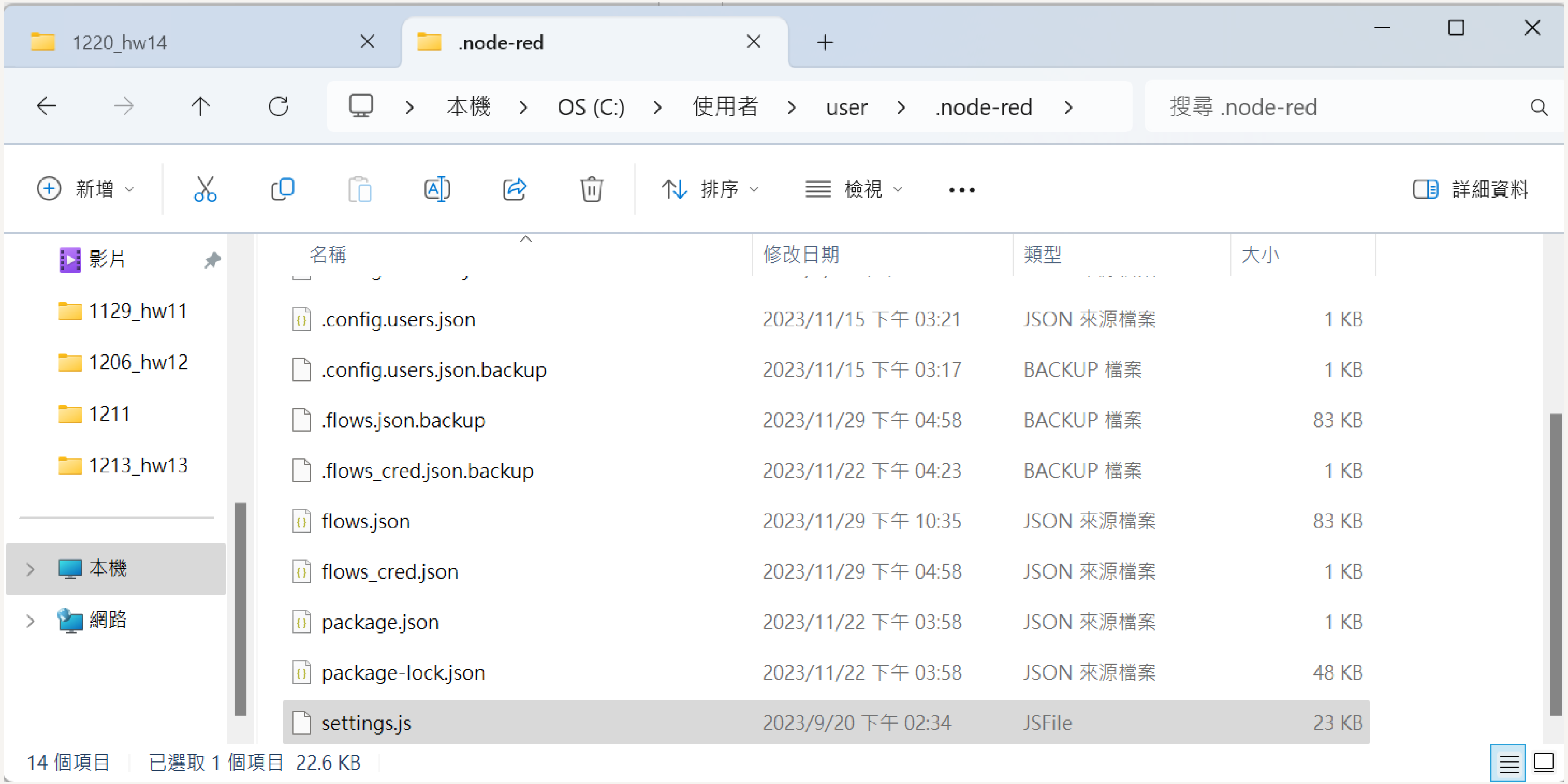
```
C:\Users\user>npm install --save crypto-js

added 1 package in 2s
npm notice
npm notice New patch version of npm available! 10.2.0 -> 10.2.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice

C:\Users\user>npm install -g npm@10.2.5
[                    ] | reify:npm: timing reify:createSparse Completed in
```

Done



```
added 1 package in 2s
npm notice
npm notice New patch version of npm available! 10.2.0 -> 10.2.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice

C:\Users\user>npm install -g npm@10.2.5

removed 44 packages, and changed 56 packages in 4s

28 packages are looking for funding
  run `npm fund` for details

C:\Users\user>
```

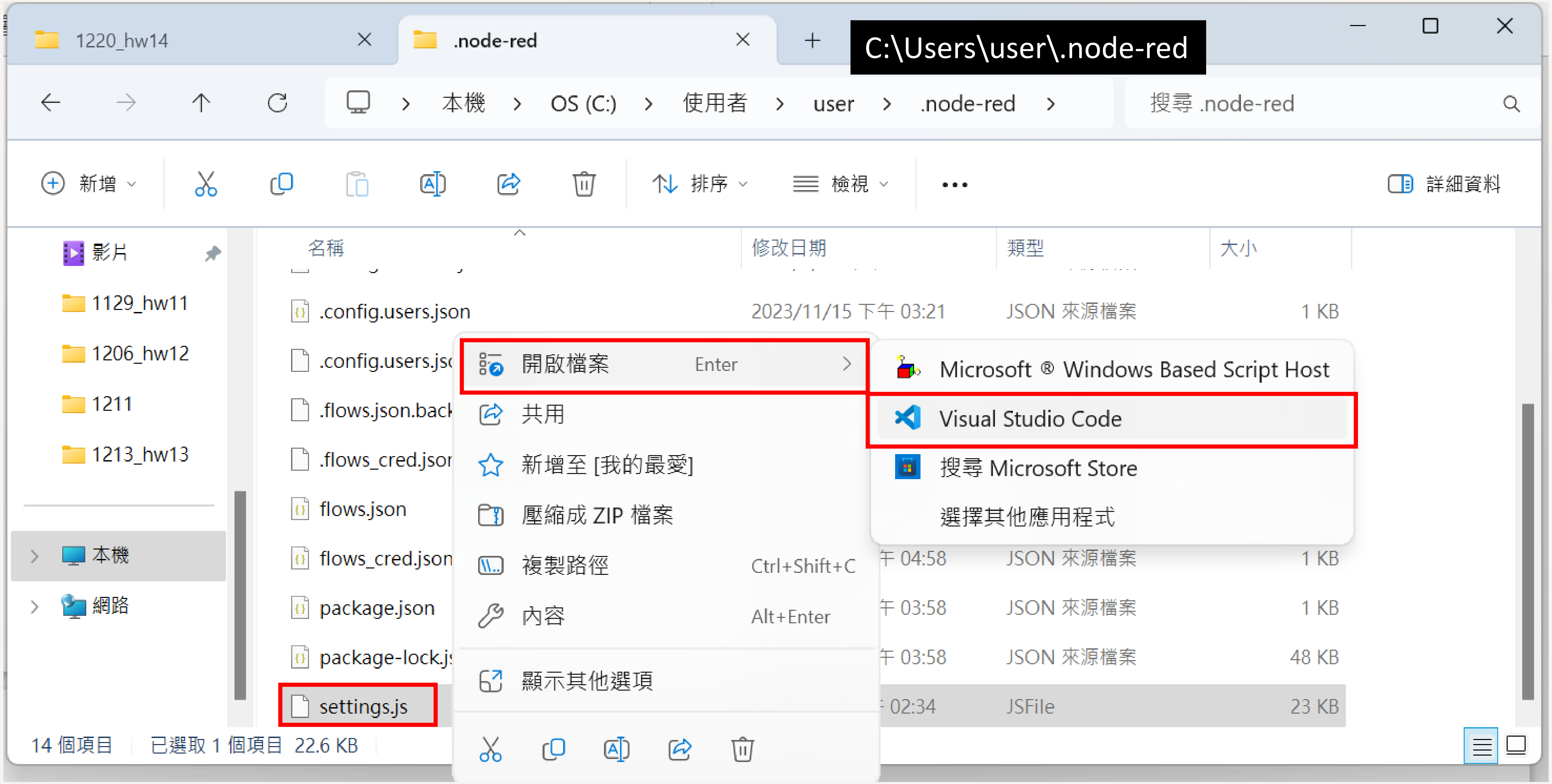
Edit settings.js



名稱	修改日期	類型	大小
.config.users.json	2023/11/15 下午 03:21	JSON 來源檔案	1 KB
.config.users.json.backup	2023/11/15 下午 03:17	BACKUP 檔案	1 KB
.flows.json.backup	2023/11/29 下午 04:58	BACKUP 檔案	83 KB
.flows_cred.json.backup	2023/11/22 下午 04:23	BACKUP 檔案	1 KB
flows.json	2023/11/29 下午 10:35	JSON 來源檔案	83 KB
flows_cred.json	2023/11/29 下午 04:58	JSON 來源檔案	1 KB
package.json	2023/11/22 下午 03:58	JSON 來源檔案	1 KB
package-lock.json	2023/11/22 下午 03:58	JSON 來源檔案	48 KB
settings.js	2023/9/20 下午 02:34	JSFile	23 KB

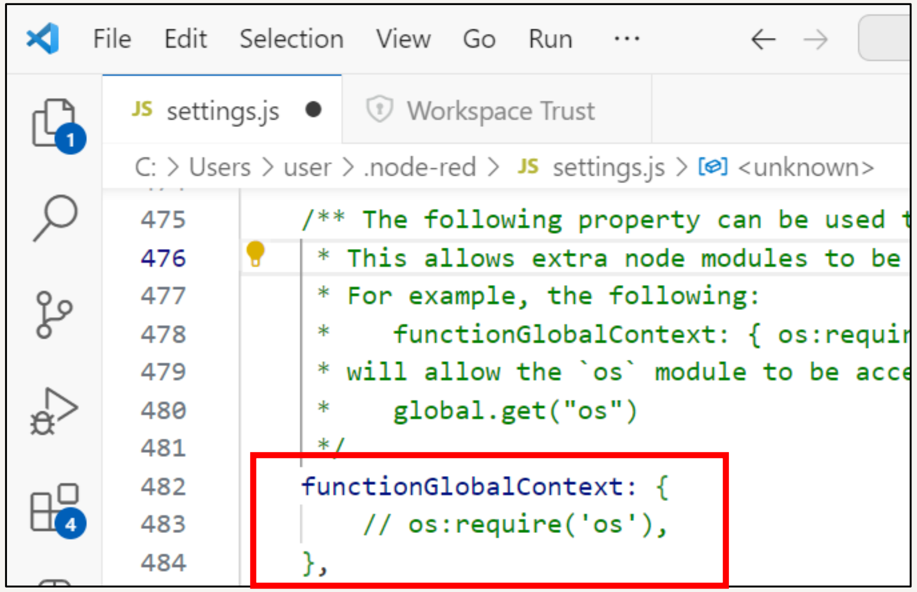
14 個項目　已選取 1 個項目 22.6 KB

Edit settings.js

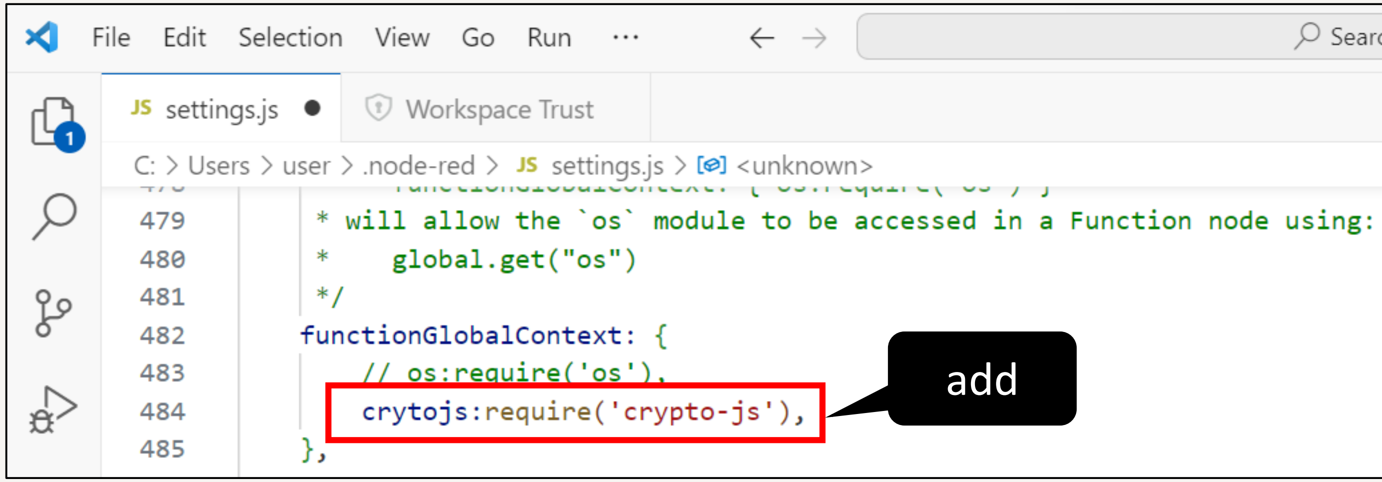


C:\Users\user\.node-red

Edit settings.js



```
/** The following property can be used t
 * This allows extra node modules to be
 * For example, the following:
 *    functionGlobalContext: { os:requir
 * will allow the `os` module to be acce
 *    global.get("os")
 */
functionGlobalContext: {
    // os:require('os'),
},
```

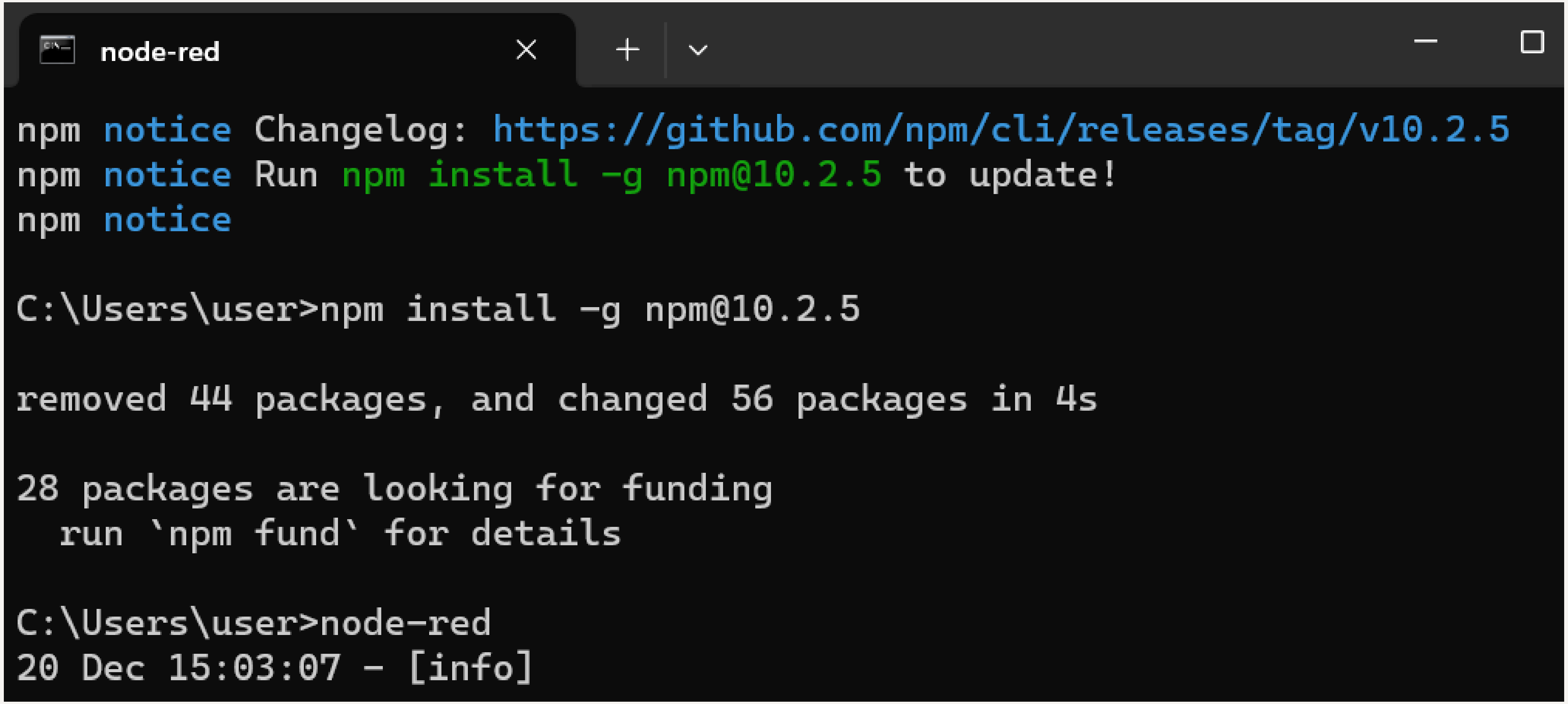


```
 * will allow the `os` module to be accessed in a Function node using:
 *    global.get("os")
 */
functionGlobalContext: {
    // os:require('os'),
    crytojs:require('crypto-js'),
},
```

add

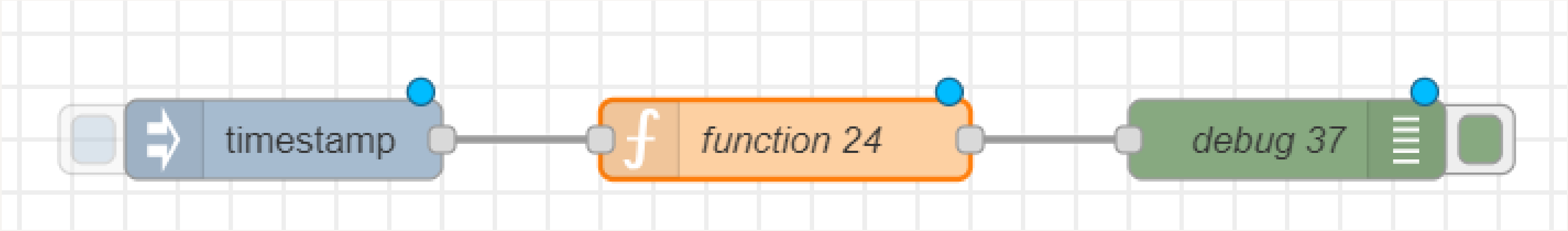
Save

Run node-red



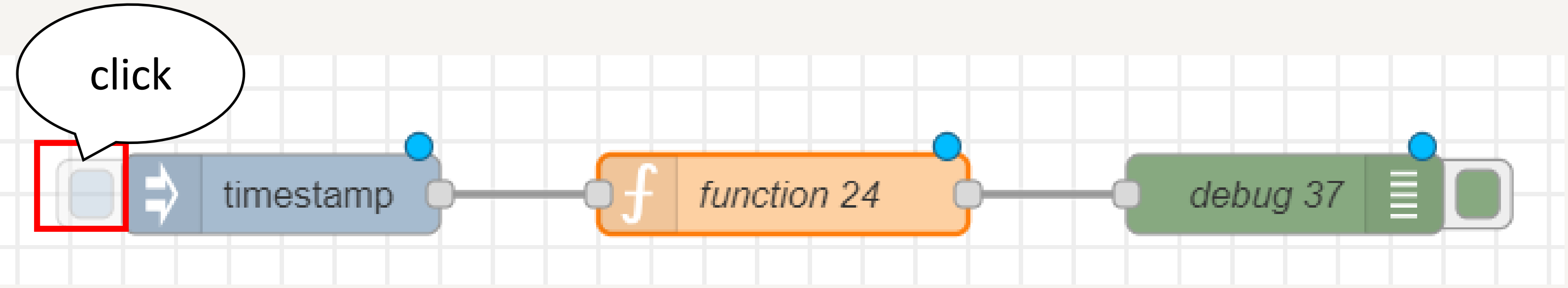
```
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice

C:\Users\user>npm install -g npm@10.2.5

removed 44 packages, and changed 56 packages in 4s

28 packages are looking for funding
  run `npm fund` for details

C:\Users\user>node-red
20 Dec 15:03:07 - [info]
```

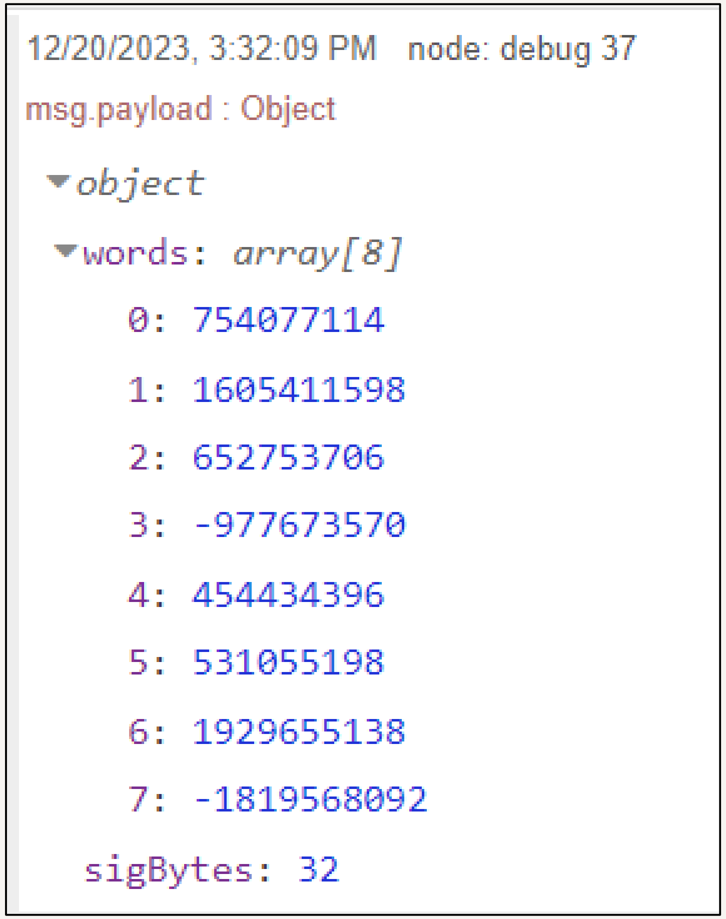
Build a Flow



```
var cryptojs = context.global.cryptojs;
var Hash = cryptojs.SHA256("hello");
msg.payload=Hash;
return msg;
```

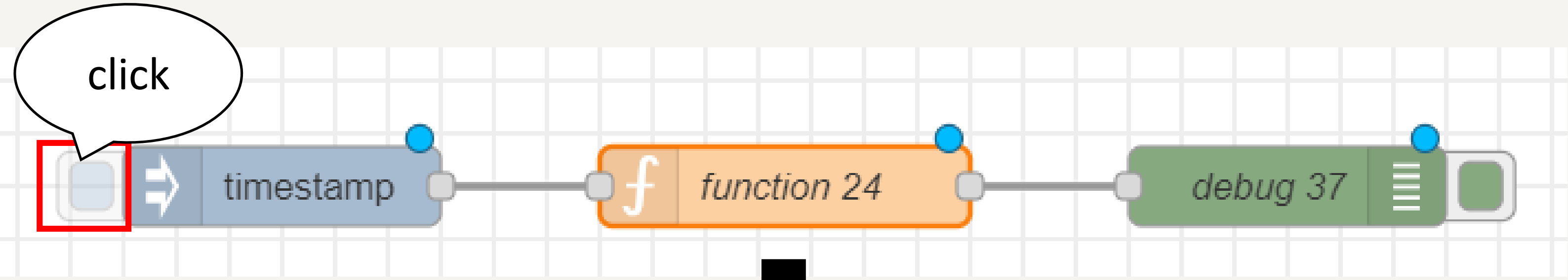
Trigger



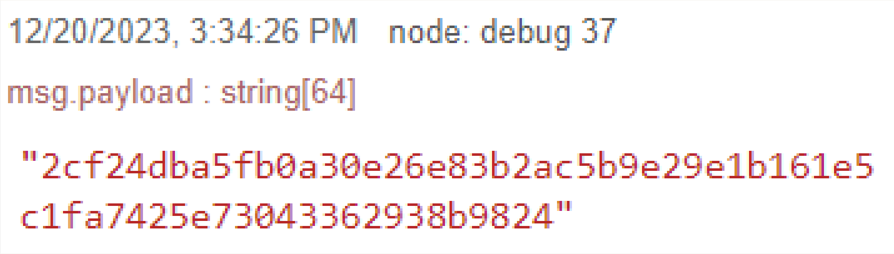


```
12/20/2023, 3:32:09 PM   node: debug 37
msg.payload : Object
 ▼object
  ▼words: array[8]
      0: 754077114
      1: 1605411598
      2: 652753706
      3: -977673570
      4: 454434396
      5: 531055198
      6: 1929655138
      7: -1819568092
    sigBytes: 32
```

JavaScript Object toString()

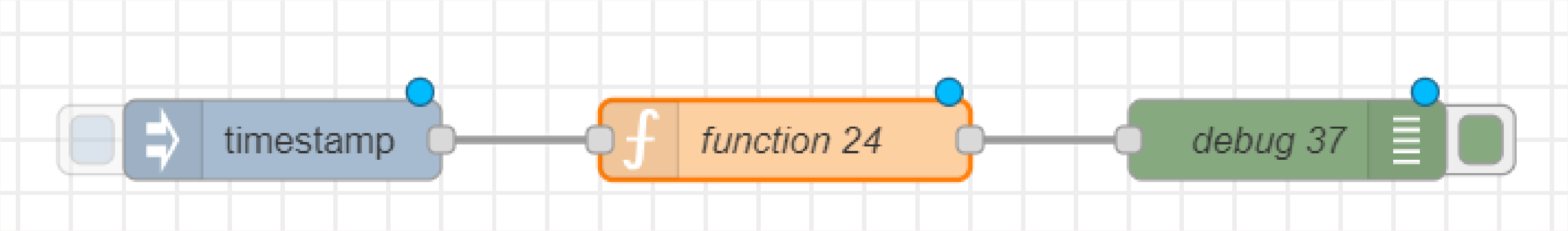


```
var cryptojs = context.global.cryptojs;
var Hash = cryptojs.SHA256("hello");
msg.payload=Hash.toString();
return msg;
```

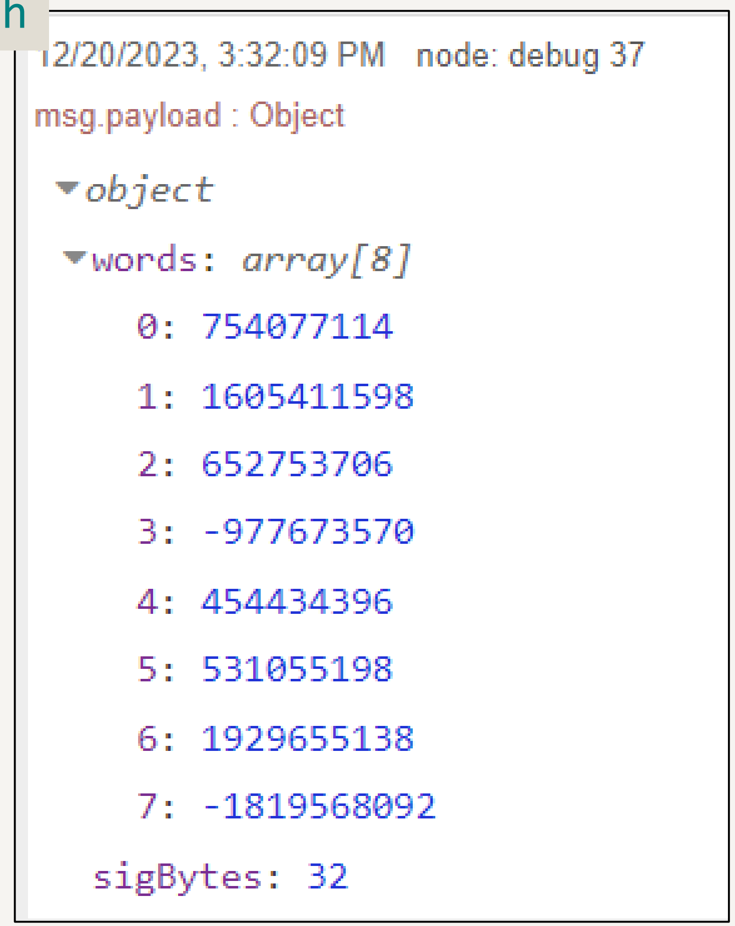


```
12/20/2023, 3:34:26 PM   node: debug 37
msg.payload : string[64]
"2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5
c1fa7425e73043362938b9824"
```

JavaScript Object toString()

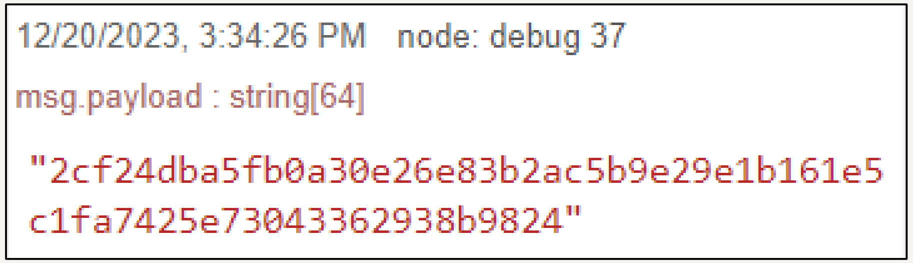


Hash



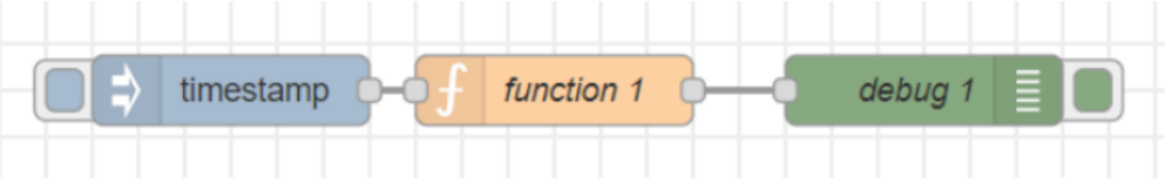
```
12/20/2023, 3:32:09 PM   node: debug 37
msg.payload : Object
 ▾object
  ▾words: array[8]
      0: 754077114
      1: 1605411598
      2: 652753706
      3: -977673570
      4: 454434396
      5: 531055198
      6: 1929655138
      7: -1819568092
    sigBytes: 32
```

`Hash.toString();`



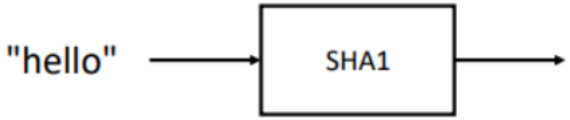
```
12/20/2023, 3:34:26 PM   node: debug 37
msg.payload : string[64]
 "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5
 c1fa7425e73043362938b9824"
```

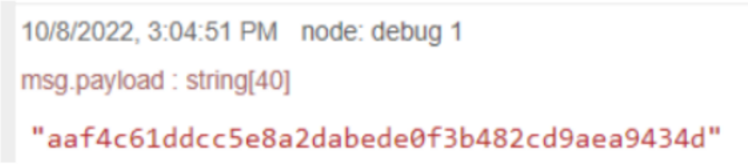
Homework 14-1



- Try SHA1

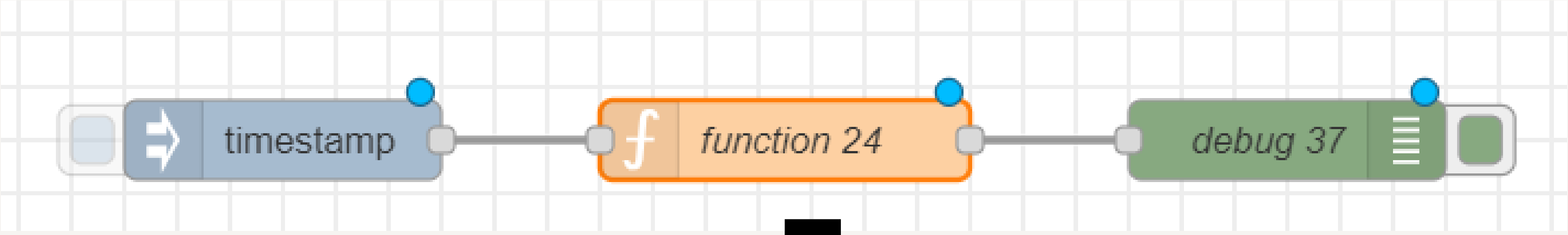
```
var Hash = cryptojs.SHA1("hello");
```



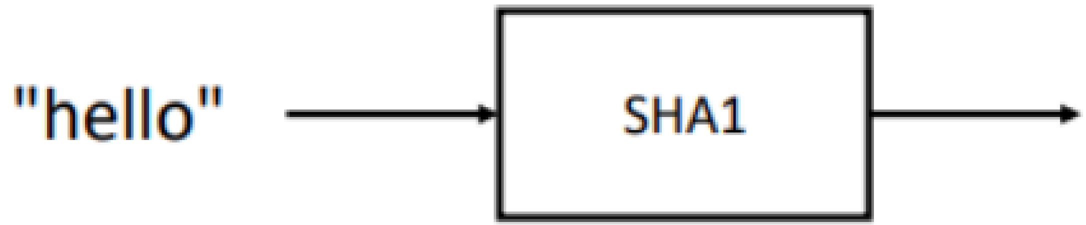


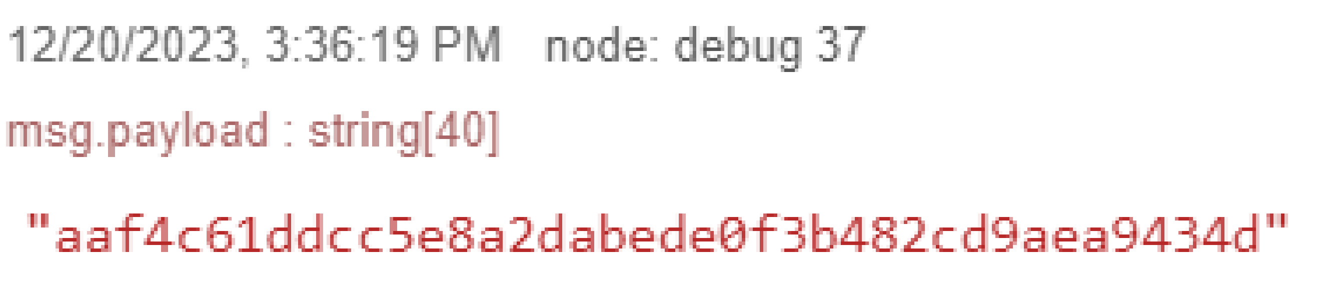
```
10/8/2022, 3:04:51 PM   node: debug 1
msg.payload : string[40]
 "aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d"
```

```
var Hash = cryptojs.SHA1("hello");
```



```
var cryptojs = context.global.cryptojs;
var Hash = cryptojs.SHA1("hello");
msg.payload=Hash.toString();
return msg;
```

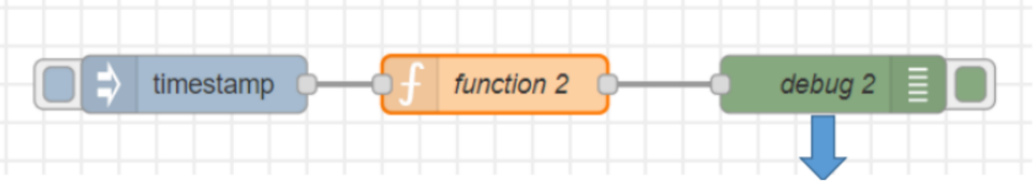




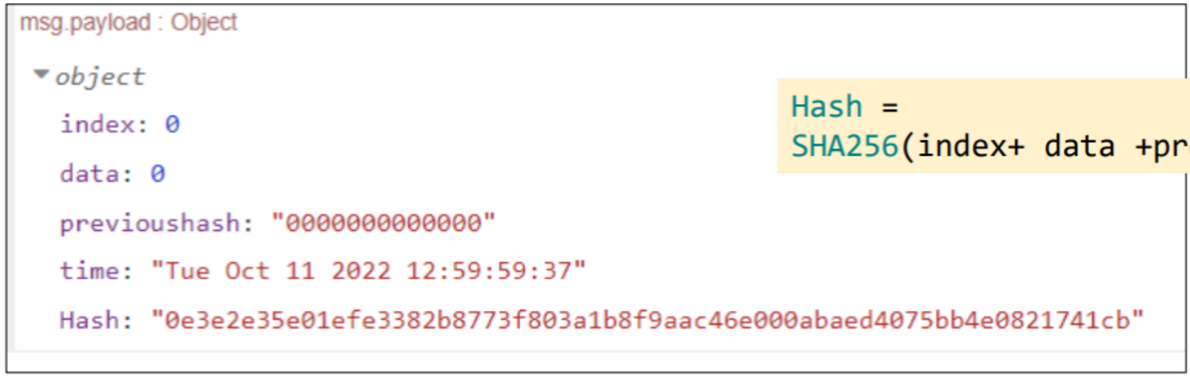
```
12/20/2023, 3:36:19 PM   node: debug 37
msg.payload : string[40]
"aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d"
```

Exercise 14-2

- Create an object

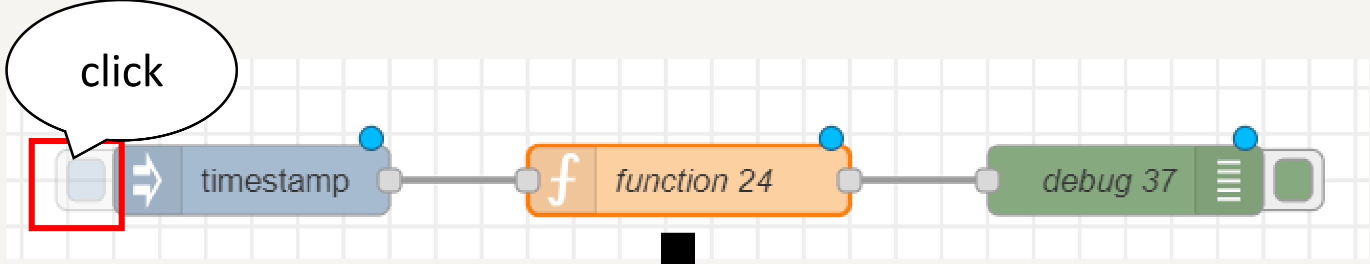


index
data
previoushash
time
Hash

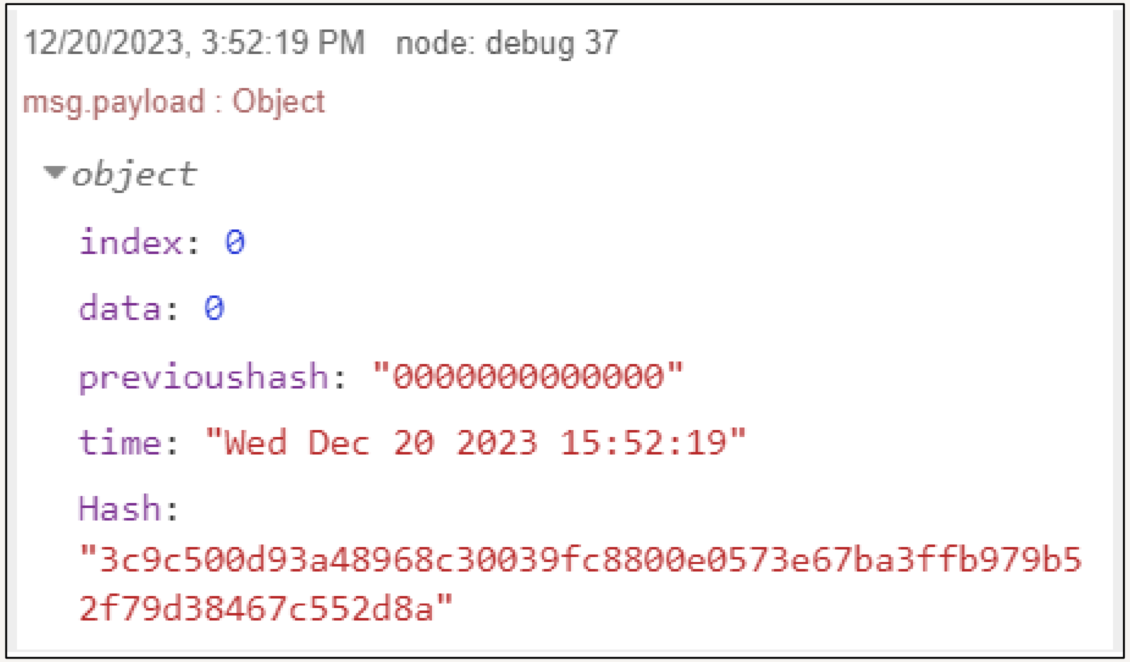


```
msg.payload : Object
▾object
  index: 0
  data: 0
  previoushash: "0000000000000"
  time: "Tue Oct 11 2022 12:59:59:37"
  Hash: "0e3e2e35e01efe3382b8773f803a1b8f9aac46e000abaed4075bb4e0821741cb"
```

```
Hash =
SHA256(index+ data +previoushash + time);
```

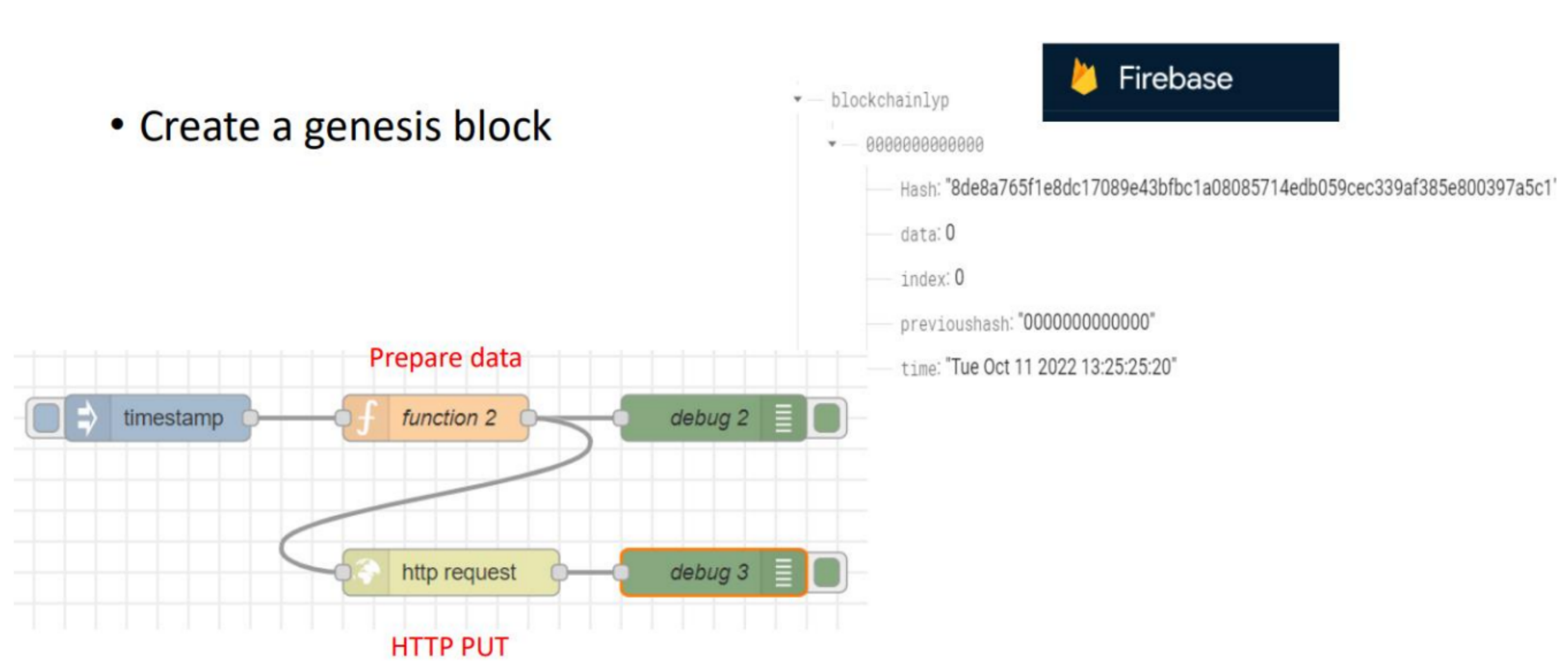


```
var cryptojs = context.global.cryptojs;
let data =0;
let previoushash="0000000000000";
const d = new Date();
var timestamp = d.getTime();
var time = d.toDateString()+" " +
d.getHours()+":" + d.getMinutes()+":" +
d.getSeconds();
var index = 0;

msg.url="";

var Hash = cryptojs.SHA256(previoushash + index
+ data + time);
msg.payload= {"index":index, "data":data,
"previoushash":previoushash,"time":time, "Hash":
Hash.toString()};
return msg;
```



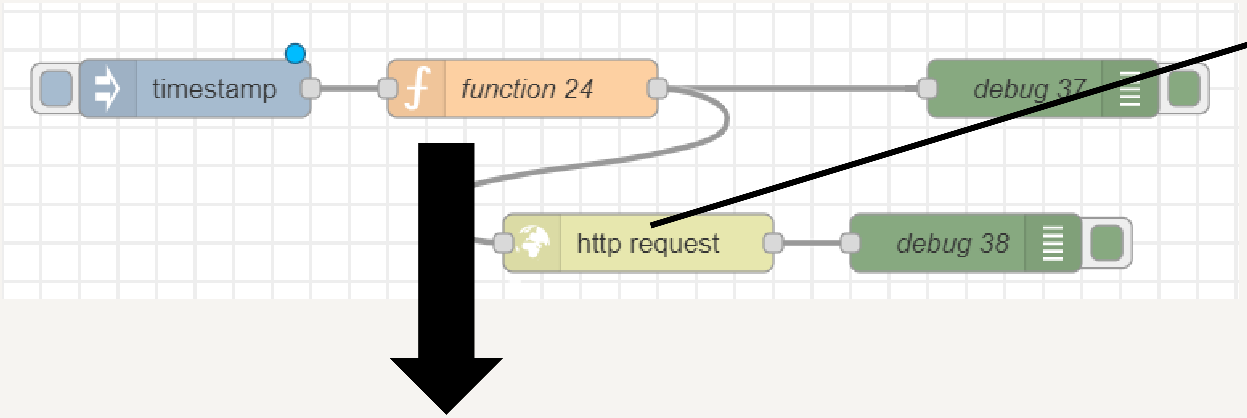
```
12/20/2023, 3:52:19 PM   node: debug 37
msg.payload : Object
▼object
  index: 0
  data: 0
  previoushash: "0000000000000"
  time: "Wed Dec 20 2023 15:52:19"
  Hash:
  "3c9c500d93a48968c30039fc8800e0573e67ba3ffb979b5
  2f79d38467c552d8a"
```

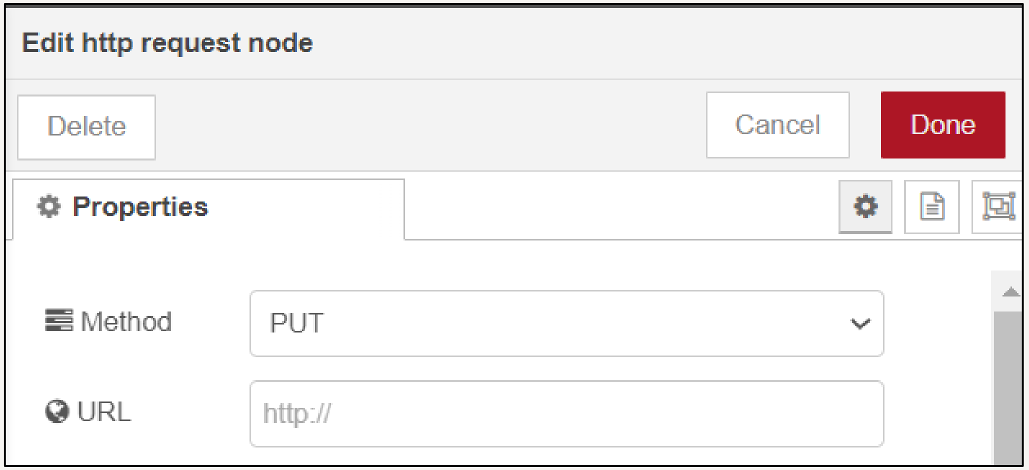
Exercise 14-3



- Create a genesis block

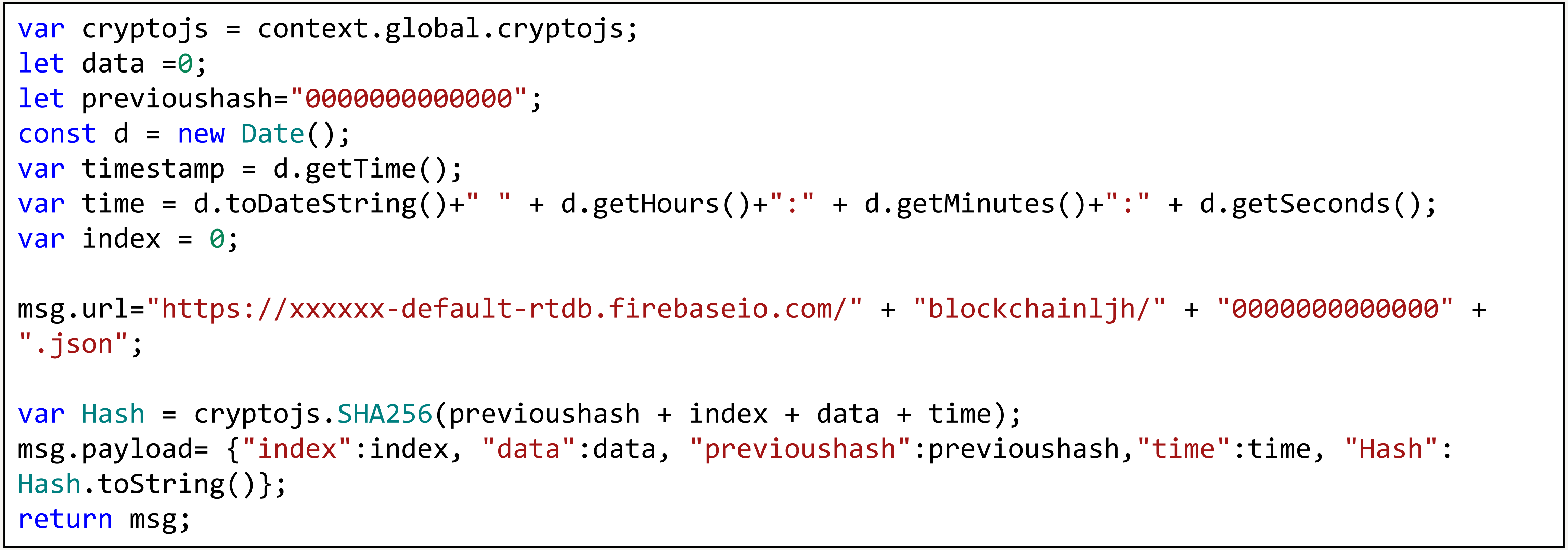
```
▾ — blockchainlyp
    ▾ — 0000000000000
          — Hash: "8de8a765f1e8dc17089e43bfbc1a08085714edb059cec339af385e800397a5c1"
          — data: 0
          — index: 0
          — previoushash: "0000000000000"
          — time: "Tue Oct 11 2022 13:25:25:20"
```





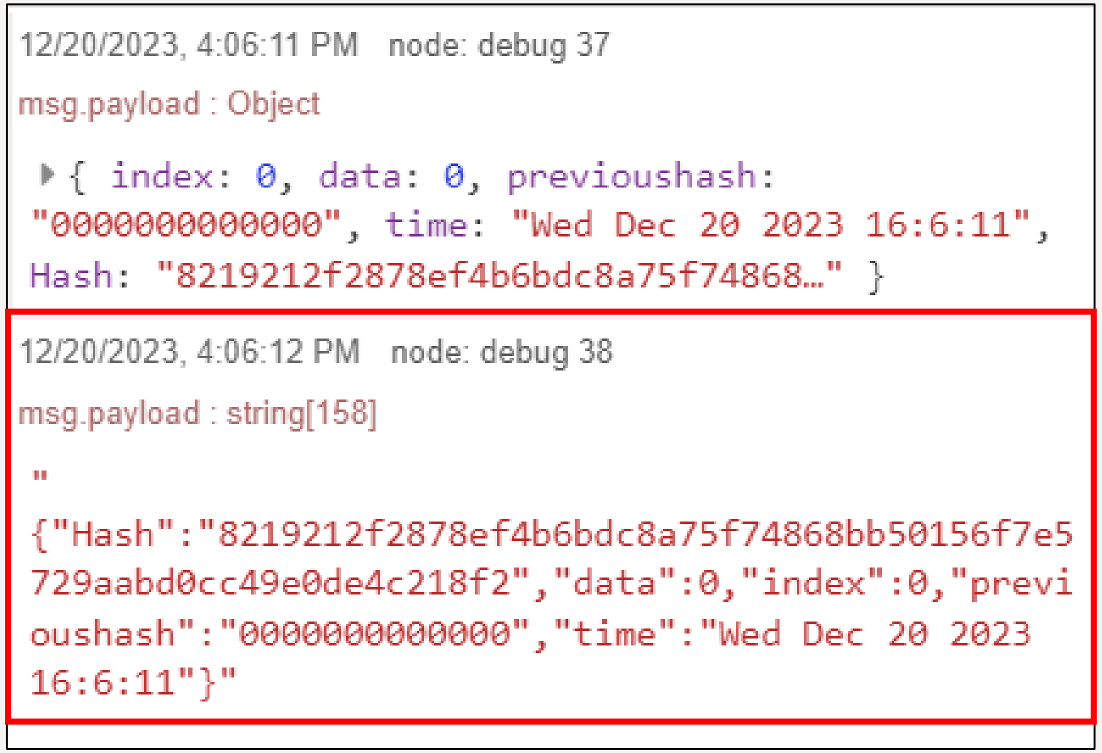
Method PUT

URL http://

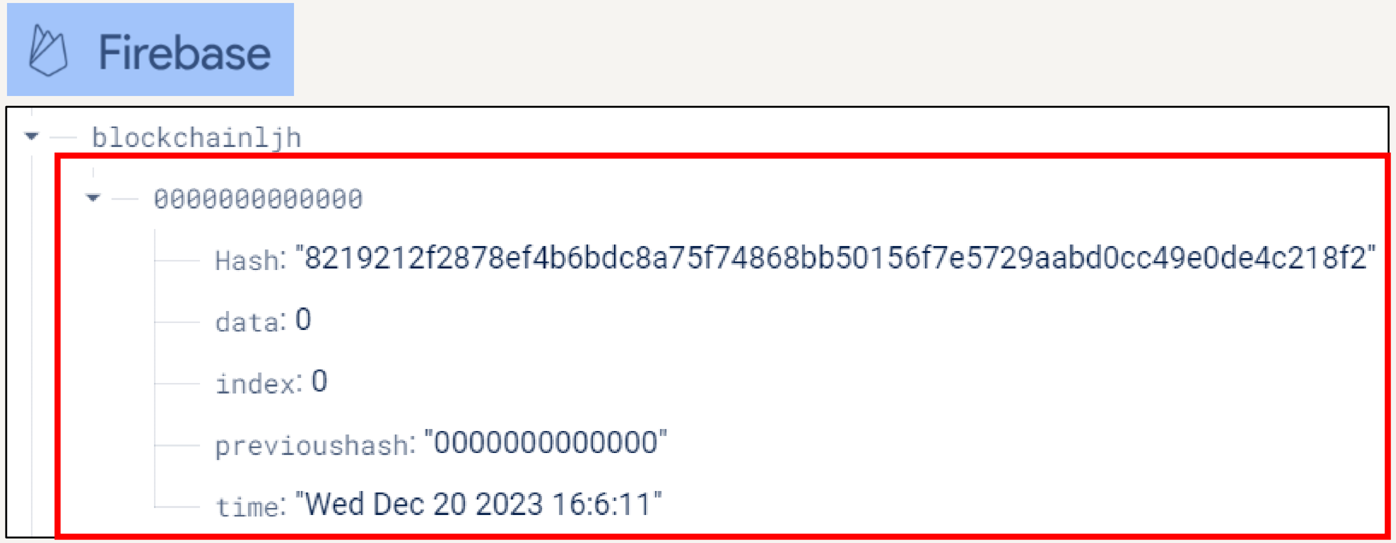


```
var cryptojs = context.global.cryptojs;
let data =0;
let previoushash="0000000000000";
const d = new Date();
var timestamp = d.getTime();
var time = d.toDateString()+" " + d.getHours()+":" + d.getMinutes()+":" + d.getSeconds();
var index = 0;

msg.url="https://xxxxxx-default-rtdb.firebaseio.com/" + "blockchainljh/" + "0000000000000" +
".json";

var Hash = cryptojs.SHA256(previoushash + index + data + time);
msg.payload= {"index":index, "data":data, "previoushash":previoushash,"time":time, "Hash":
Hash.toString()};
return msg;
```

Check new data



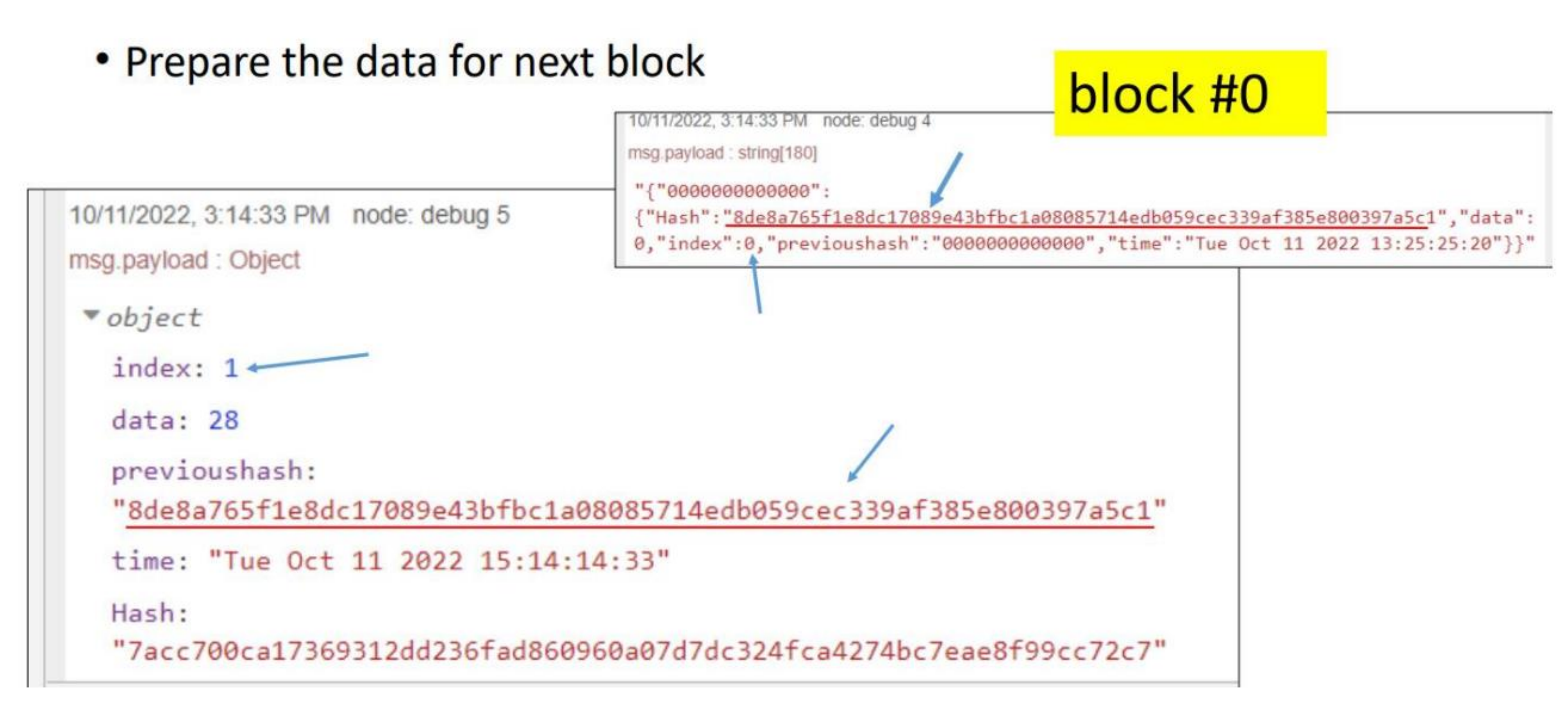
```
12/20/2023, 4:06:11 PM   node: debug 37
msg.payload : Object
▸{ index: 0, data: 0, previoushash:
"0000000000000", time: "Wed Dec 20 2023 16:6:11",
Hash: "8219212f2878ef4b6bdc8a75f74868…" }

12/20/2023, 4:06:12 PM   node: debug 38
msg.payload : string[158]
"
{"Hash":"8219212f2878ef4b6bdc8a75f74868bb50156f7e5
729aabd0cc49e0de4c218f2","data":0,"index":0,"previ
oushash":"0000000000000","time":"Wed Dec 20 2023
16:6:11"}"
```



Firebase

```
▾ blockchainljh
   ▾ 0000000000000
        Hash: "8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49e0de4c218f2"
        data: 0
        index: 0
        previoushash: "0000000000000"
        time: "Wed Dec 20 2023 16:6:11"
```

Exercise 14-4



- Prepare the data for next block

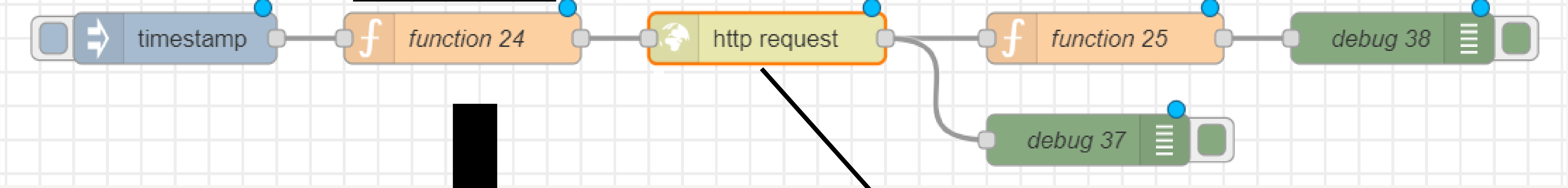
block #0

```
10/11/2022, 3:14:33 PM   node: debug 4
msg.payload : string[180]
"{"0000000000000":
{"Hash":"8de8a765f1e8dc17089e43bfbc1a08085714edb059cec339af385e800397a5c1","data":
0,"index":0,"previoushash":"0000000000000","time":"Tue Oct 11 2022 13:25:25:20"}}"
```

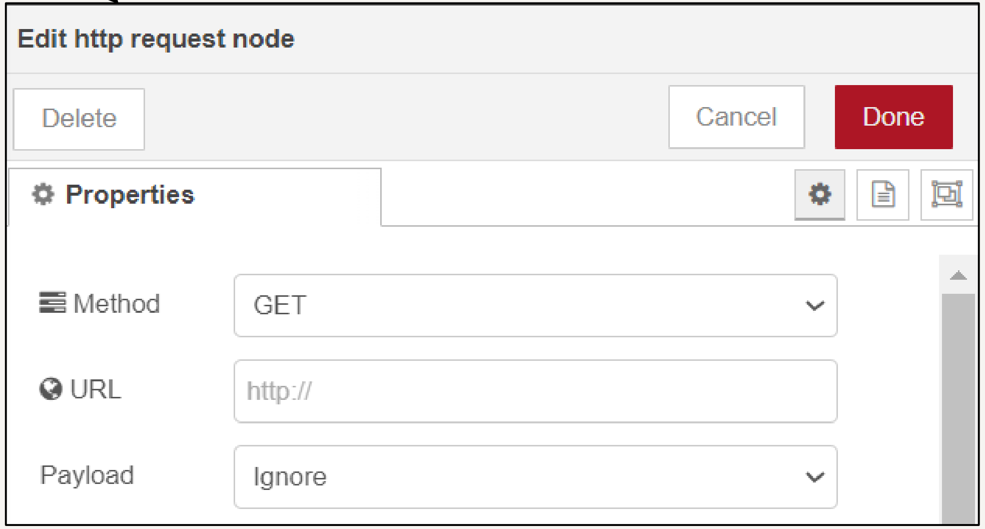
```
10/11/2022, 3:14:33 PM   node: debug 5
msg.payload : Object
▾object
  index: 1
  data: 28
  previoushash:
  "8de8a765f1e8dc17089e43bfbc1a08085714edb059cec339af385e800397a5c1"
  time: "Tue Oct 11 2022 15:14:14:33"
  Hash:
  "7acc700ca17369312dd236fad860960a07d7dc324fca4274bc7eae8f99cc72c7"
```

Set URL

HTTP GET



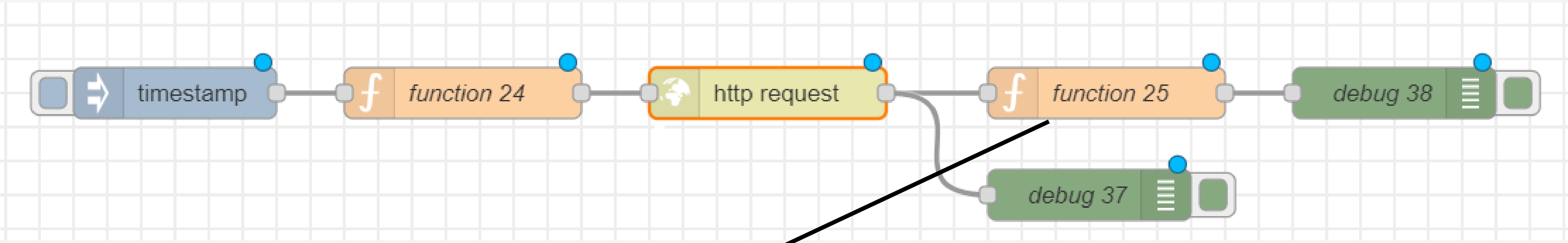
```
msg.url="https://aiotdemo-f60d9-default-
rtdb.firebaseio.com/" + "blockchainljh.json";
return msg;
```

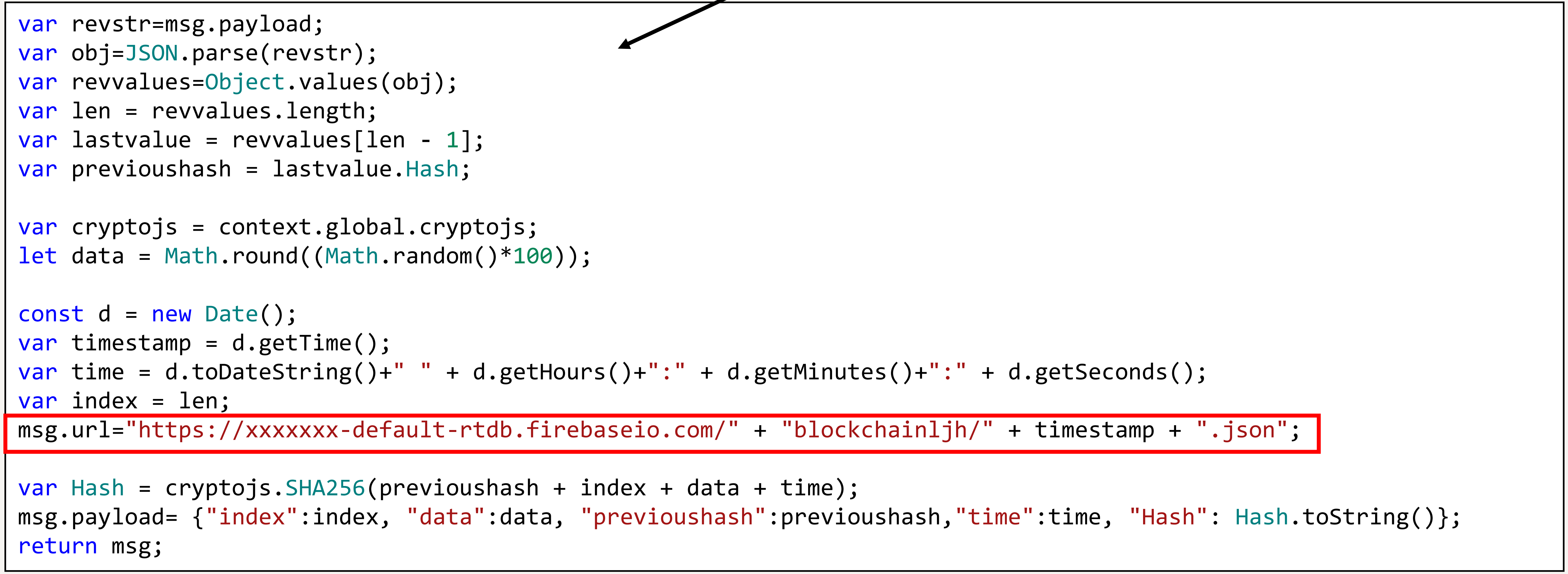


Method: GET

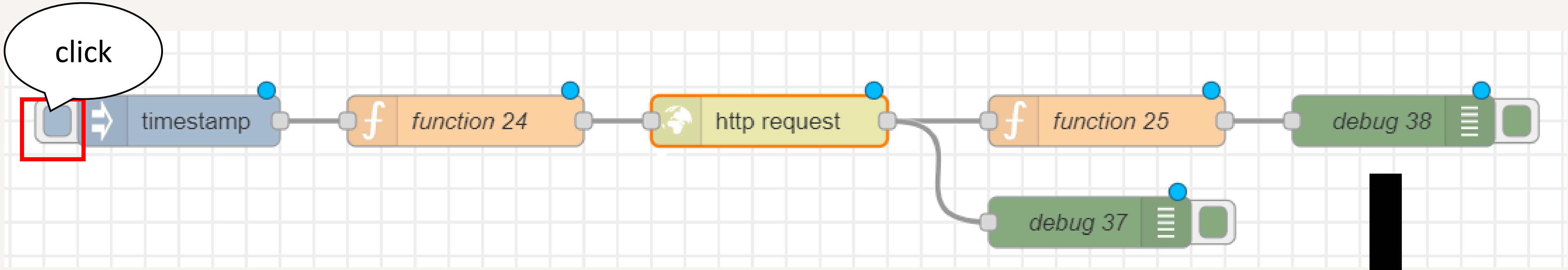
URL: http://

Payload: Ignore

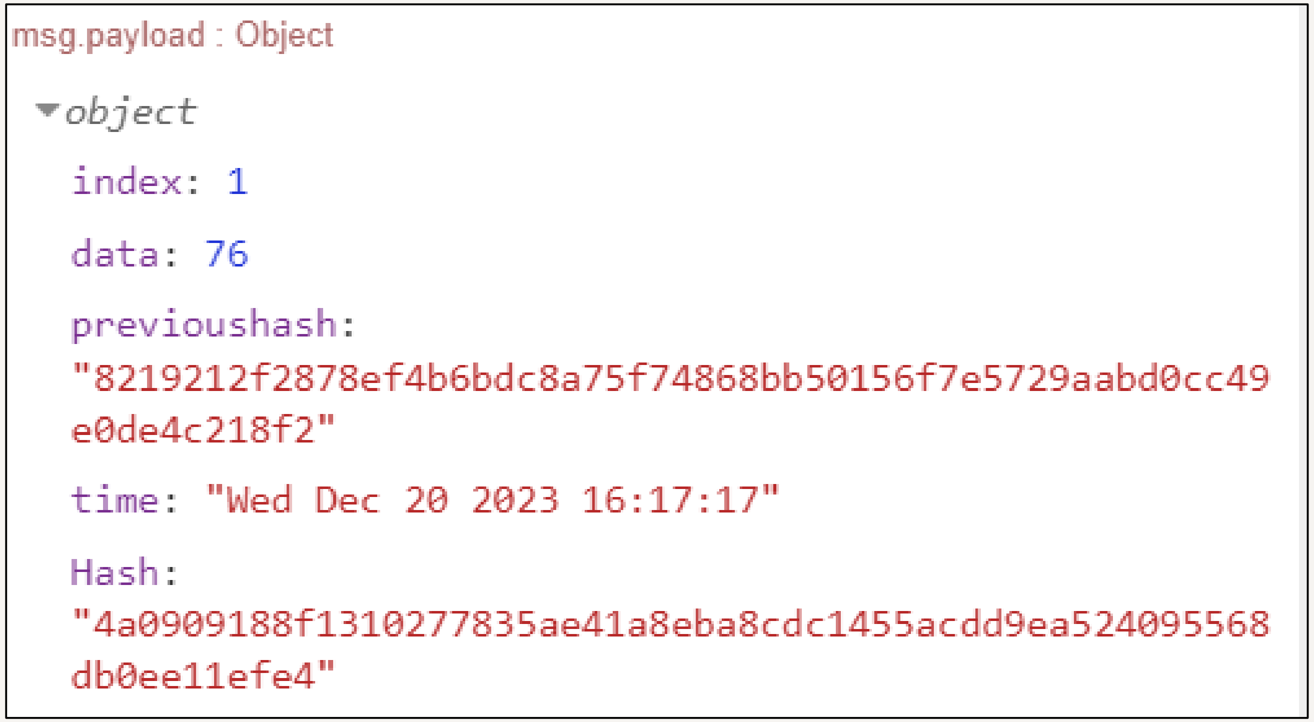




```
var revstr=msg.payload;
var obj=JSON.parse(revstr);
var revvalues=Object.values(obj);
var len = revvalues.length;
var lastvalue = revvalues[len - 1];
var previoushash = lastvalue.Hash;

var cryptojs = context.global.cryptojs;
let data = Math.round((Math.random()*100));

const d = new Date();
var timestamp = d.getTime();
var time = d.toDateString()+" " + d.getHours()+":" + d.getMinutes()+":" + d.getSeconds();
var index = len;
msg.url="https://xxxxxxx-default-rtdb.firebaseio.com/" + "blockchainljh/" + timestamp + ".json";

var Hash = cryptojs.SHA256(previoushash + index + data + time);
msg.payload= {"index":index, "data":data, "previoushash":previoushash,"time":time, "Hash": Hash.toString()};
return msg;
```



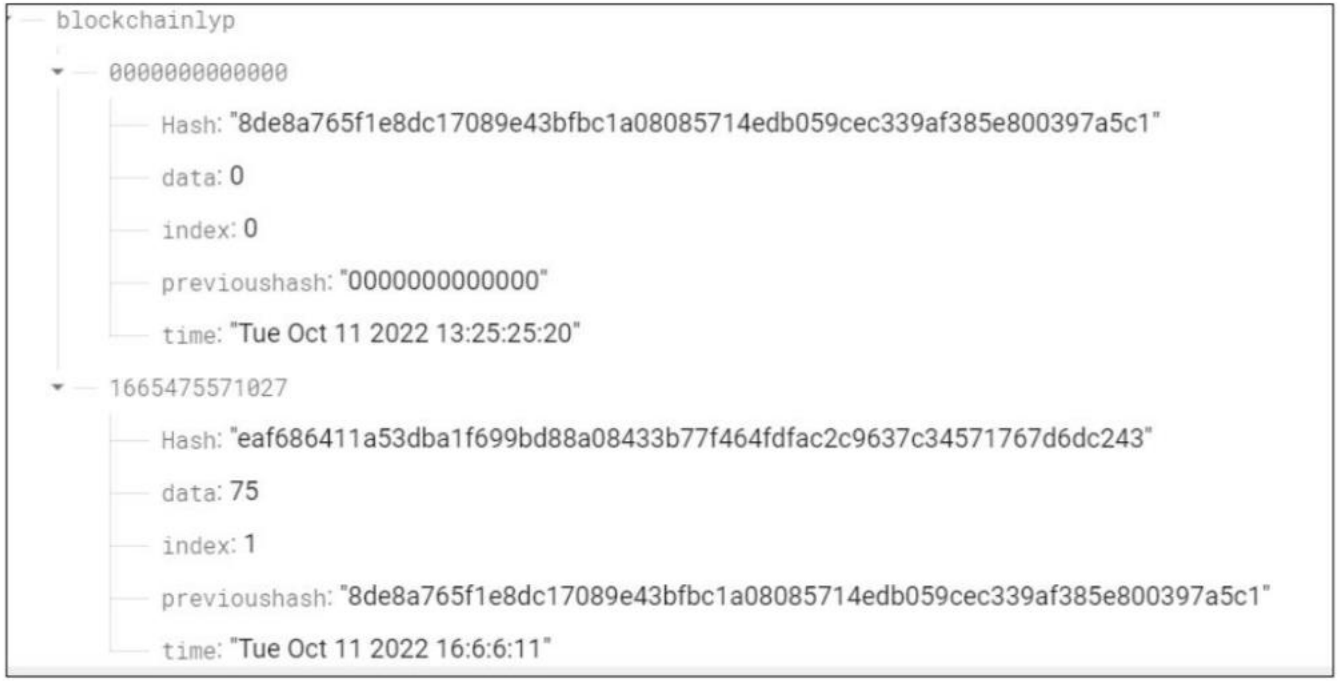
click



```
msg.payload : Object

▾object
  index: 1
  data: 76
  previoushash:
  "8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49
  e0de4c218f2"
  time: "Wed Dec 20 2023 16:17:17"
  Hash:
  "4a0909188f1310277835ae41a8eba8cdc1455acdd9ea524095568
  db0ee11efe4"
```

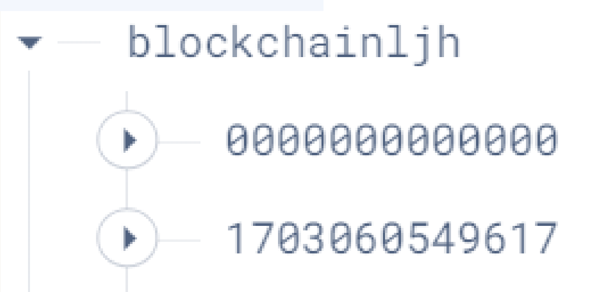
Exercise 14-5

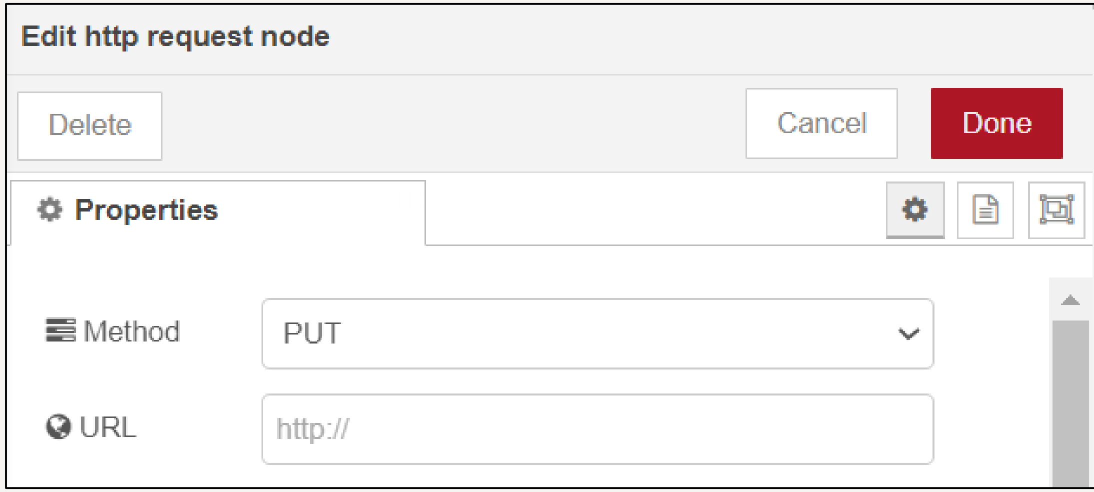
Write a block to your database.



```
blockchainlyp
  0000000000000
    Hash: "8de8a765f1e8dc17089e43bfbc1a08085714edb059cec339af385e800397a5c1"
    data: 0
    index: 0
    previoushash: "0000000000000"
    time: "Tue Oct 11 2022 13:25:25:20"
  1665475571027
    Hash: "eaf686411a53dba1f699bd88a08433b77f464fdfac2c9637c34571767d6dc243"
    data: 75
    index: 1
    previoushash: "8de8a765f1e8dc17089e43bfbc1a08085714edb059cec339af385e800397a5c1"
    time: "Tue Oct 11 2022 16:6:6:11"
```

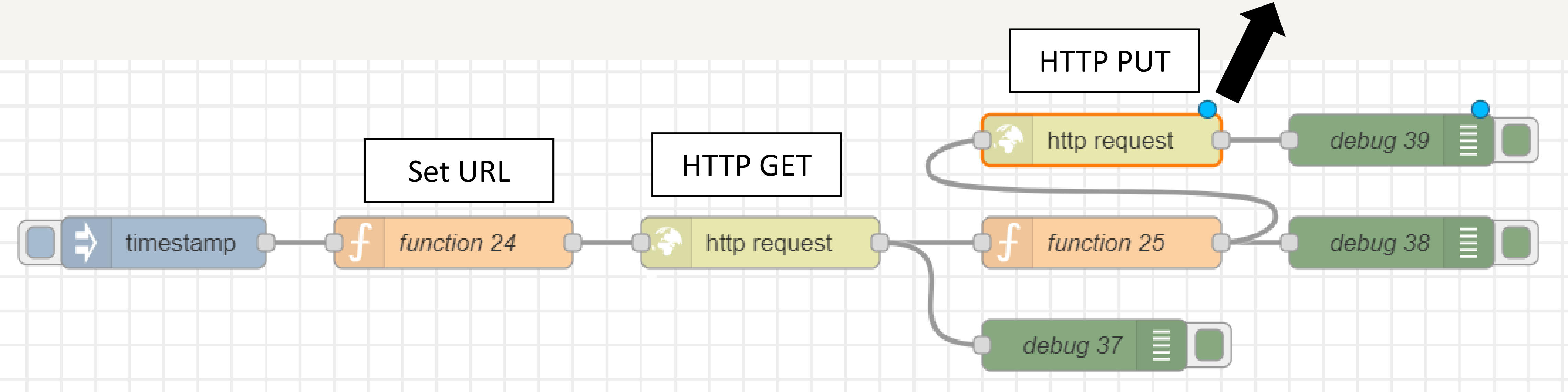
Firebase


blockchainljh
- 0000000000000
- 1703060549617


Edit http request node

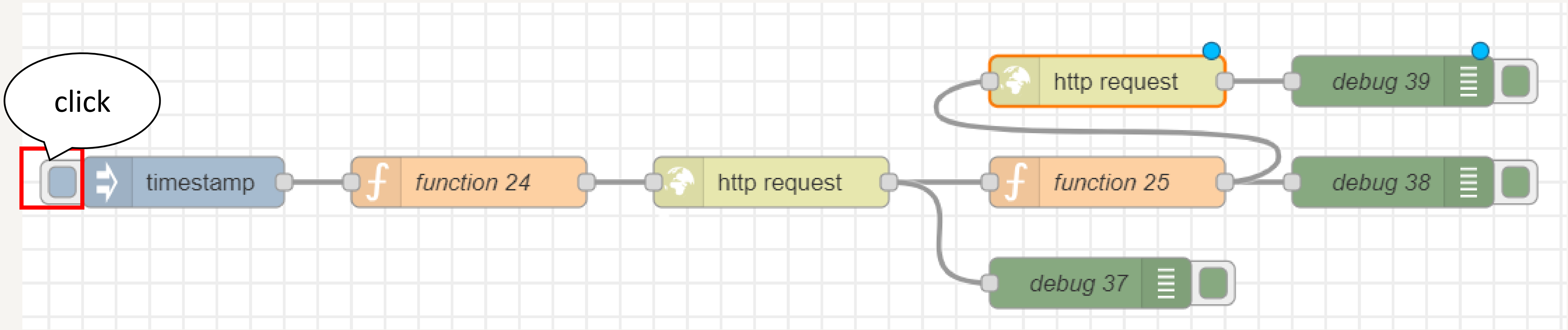
Method: PUT

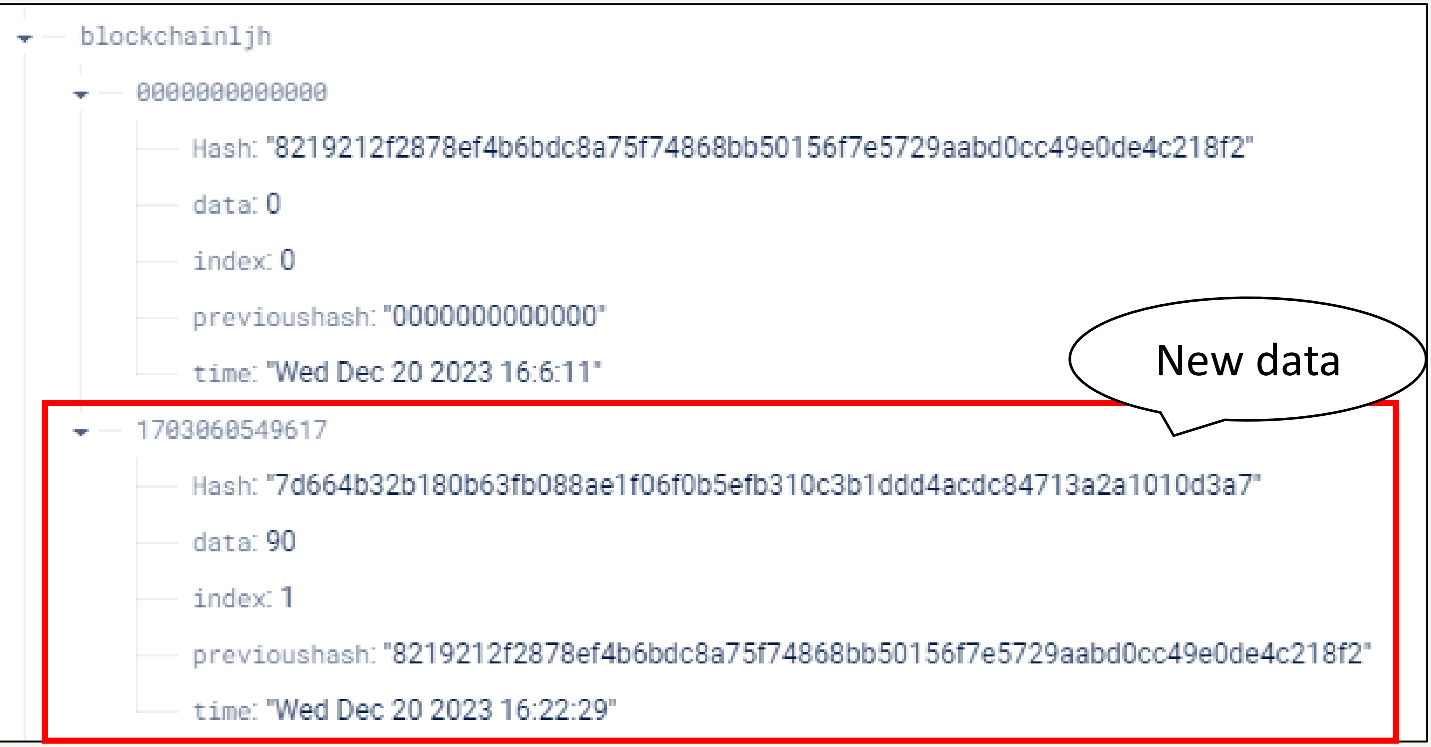
URL: http://


Set URL

HTTP GET

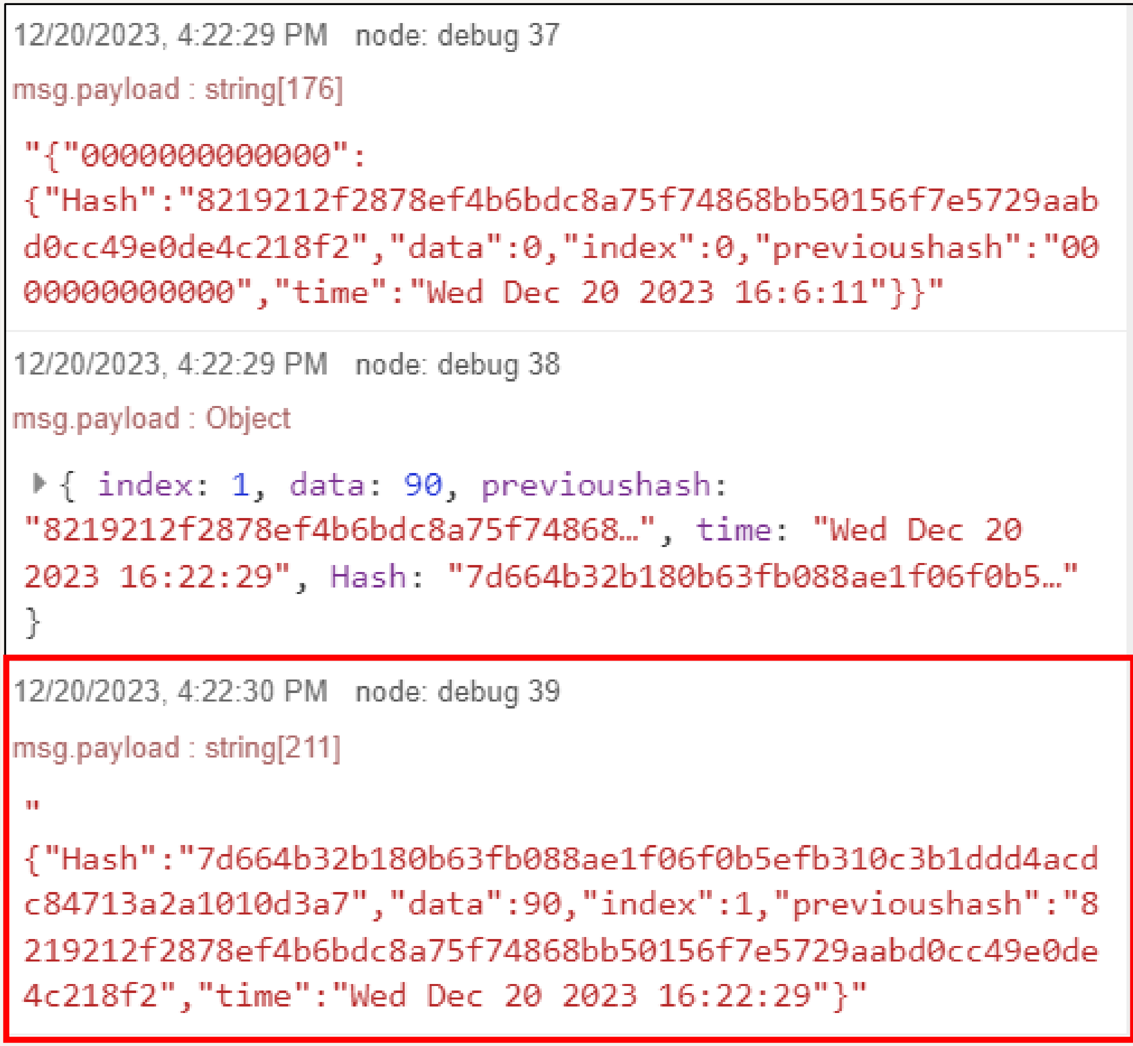
HTTP PUT


click
timestamp
function 24
http request
http request
function 25
debug 39
debug 38
debug 37

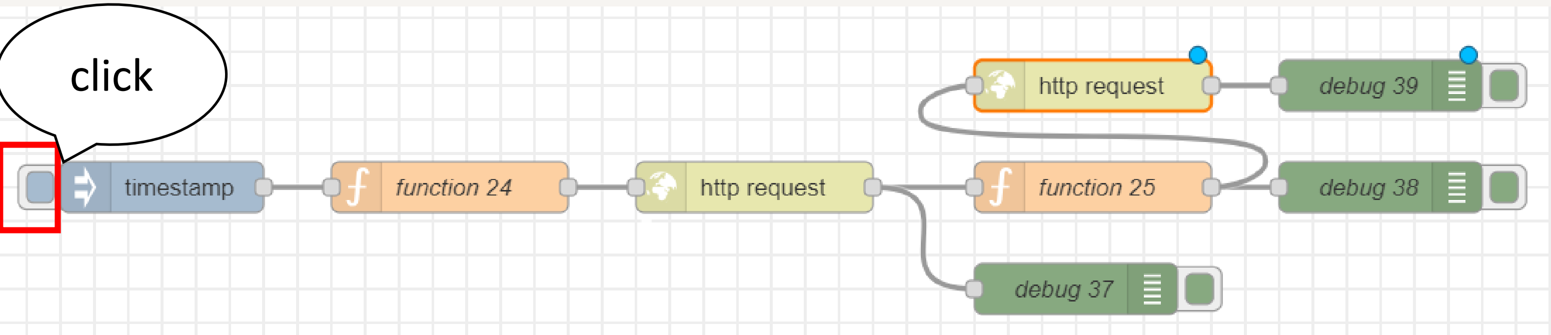


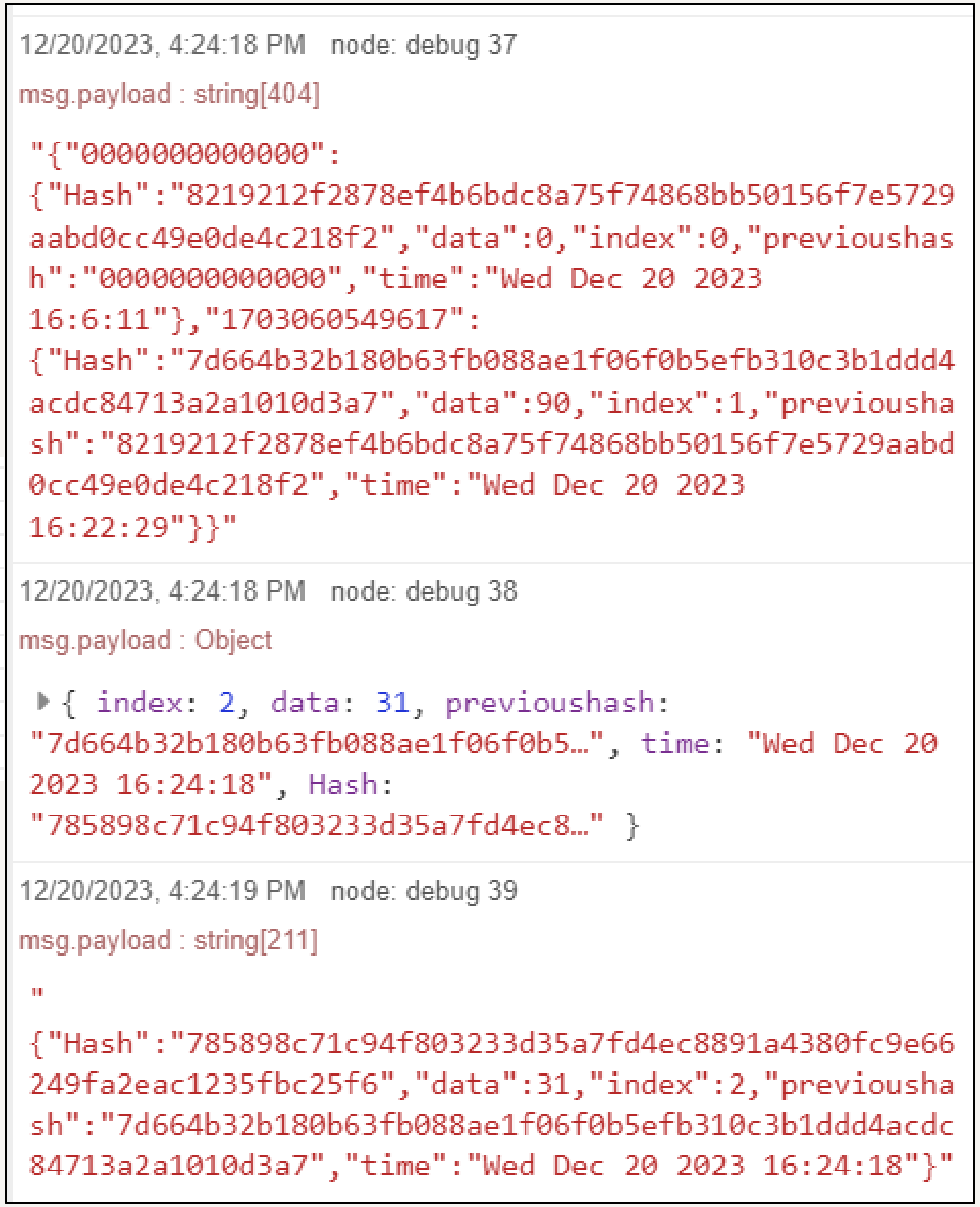
New data

```
▾ — blockchainljh
    ▾ — 0000000000000
          — Hash: "8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49e0de4c218f2"
          — data: 0
          — index: 0
          — previoushash: "0000000000000"
          — time: "Wed Dec 20 2023 16:6:11"
    ▾ — 1703060549617
          — Hash: "7d664b32b180b63fb088ae1f06f0b5efb310c3b1ddd4acdc84713a2a1010d3a7"
          — data: 90
          — index: 1
          — previoushash: "8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49e0de4c218f2"
          — time: "Wed Dec 20 2023 16:22:29"
```

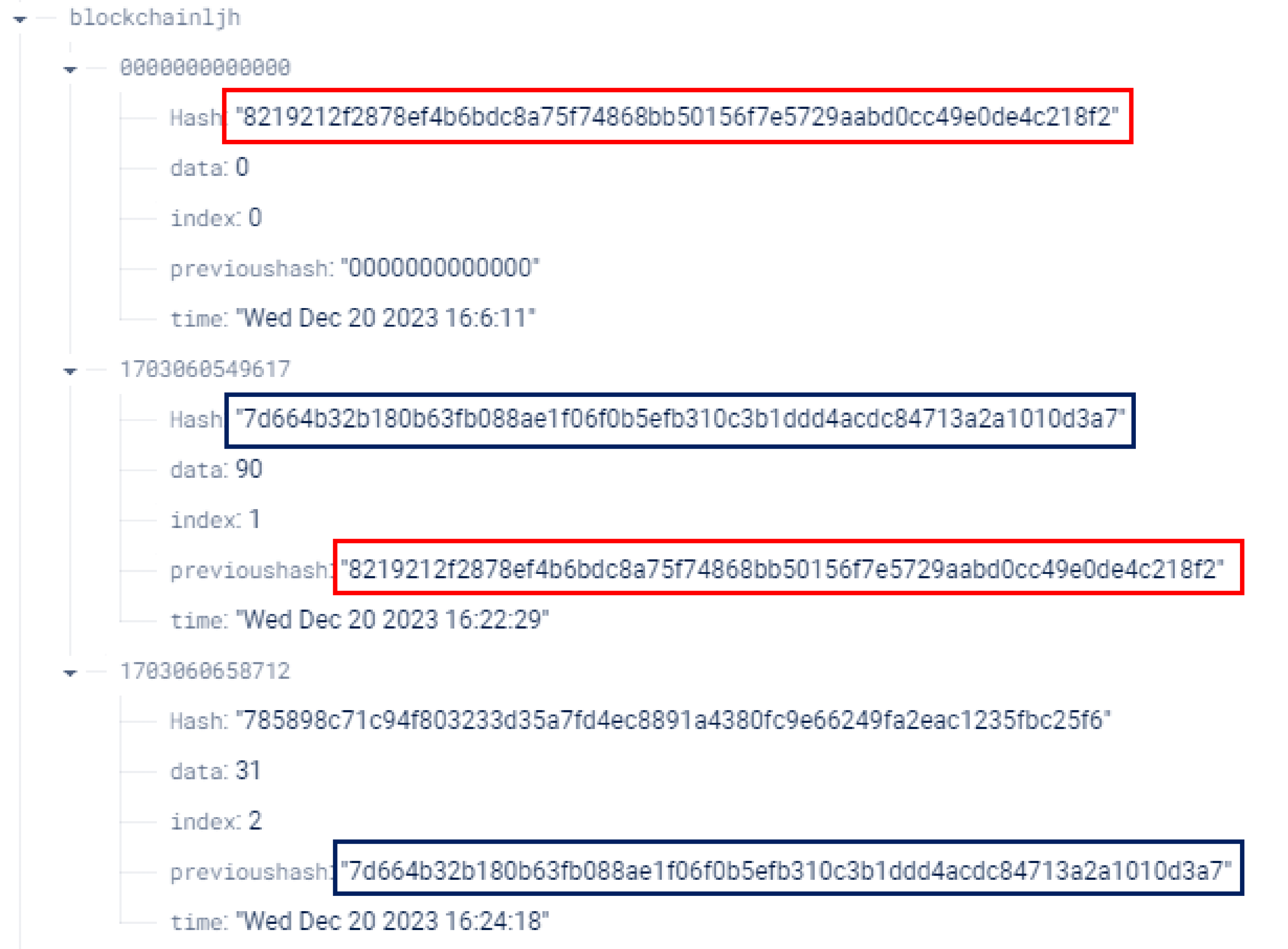


```
12/20/2023, 4:22:29 PM   node: debug 37
msg.payload : string[176]
"{"0000000000000":
{"Hash":"8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aab
d0cc49e0de4c218f2","data":0,"index":0,"previoushash":"00
00000000000","time":"Wed Dec 20 2023 16:6:11"}}"

12/20/2023, 4:22:29 PM   node: debug 38
msg.payload : Object
▸{ index: 1, data: 90, previoushash:
"8219212f2878ef4b6bdc8a75f74868…", time: "Wed Dec 20
2023 16:22:29", Hash: "7d664b32b180b63fb088ae1f06f0b5…"
}

12/20/2023, 4:22:30 PM   node: debug 39
msg.payload : string[211]
"
{"Hash":"7d664b32b180b63fb088ae1f06f0b5efb310c3b1ddd4acd
c84713a2a1010d3a7","data":90,"index":1,"previoushash":"8
219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49e0de
4c218f2","time":"Wed Dec 20 2023 16:22:29"}"
```





```
12/20/2023, 4:24:18 PM   node: debug 37
msg.payload : string[404]
"{"0000000000000":
{"Hash":"8219212f2878ef4b6bdc8a75f74868bb50156f7e5729
aabd0cc49e0de4c218f2","data":0,"index":0,"previoushas
h":"0000000000000","time":"Wed Dec 20 2023
16:6:11"},"1703060549617":
{"Hash":"7d664b32b180b63fb088ae1f06f0b5efb310c3b1ddd4
acdc84713a2a1010d3a7","data":90,"index":1,"previousha
sh":"8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd
0cc49e0de4c218f2","time":"Wed Dec 20 2023
16:22:29"}}"

12/20/2023, 4:24:18 PM   node: debug 38
msg.payload : Object
▸{ index: 2, data: 31, previoushash:
"7d664b32b180b63fb088ae1f06f0b5…", time: "Wed Dec 20
2023 16:24:18", Hash:
"785898c71c94f803233d35a7fd4ec8…" }

12/20/2023, 4:24:19 PM   node: debug 39
msg.payload : string[211]
"
{"Hash":"785898c71c94f803233d35a7fd4ec8891a4380fc9e66
249fa2eac1235fbc25f6","data":31,"index":2,"previousha
sh":"7d664b32b180b63fb088ae1f06f0b5efb310c3b1ddd4acdc
84713a2a1010d3a7","time":"Wed Dec 20 2023 16:24:18"}"
```



```
blockchainljh
  0000000000000
    Hash: "8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49e0de4c218f2"
    data: 0
    index: 0
    previoushash: "0000000000000"
    time: "Wed Dec 20 2023 16:6:11"
  1703060549617
    Hash: "7d664b32b180b63fb088ae1f06f0b5efb310c3b1ddd4acdc84713a2a1010d3a7"
    data: 90
    index: 1
    previoushash: "8219212f2878ef4b6bdc8a75f74868bb50156f7e5729aabd0cc49e0de4c218f2"
    time: "Wed Dec 20 2023 16:22:29"
  1703060658712
    Hash: "785898c71c94f803233d35a7fd4ec8891a4380fc9e66249fa2eac1235fbc25f6"
    data: 31
    index: 2
    previoushash: "7d664b32b180b63fb088ae1f06f0b5efb310c3b1ddd4acdc84713a2a1010d3a7"
    time: "Wed Dec 20 2023 16:24:18"
```