پروژه ی نرم افزار دوره کارشناسی

گزارش شماره ۹ - پیاده سازی پروژه (بخش چهارم)

آریا رادمهر - ۹۷۴۶۳۱۲۵

دکتر سجاد حق زاد کلیدبری

March 1, 2022

ساخت بخش چهارم (پیاده سازی شبکه و API ها)

فایل index.js

در فایل index.js که در دایرکتوری root پروژه قرار دارد و به صورت خودکار به عنوان فایل اجرا کننده ی پروژه صدا زده میشود، حاوی تمامی api های است که از آنها برای ارتباط با شبکه ی بلاکچین استفاده میشود. رویکرد کلی در فایل index.js این است که یک اپلیکیشن توسط فریمورک express با نام app ساخته میشود تا توسط آن متد های get و post برای دریافت و ارسال اطلاعات بر روی شبکه امکان پذیر باشد.

در این فایل تعداد ۴ عدد api استفاده میشود که عبارتند از

۱- پخش کردن بلاک های موجود و ثبت شده در بلاکچین

```
app.get('/api/blocks', (req, res) => { });
```

۲- ساخت و پخش کردن اطلاعات ماین یک بلاک و فرستادن آن به مابقی گره ها

```
app.post('/api/mine', (req, res) => { });
```

۳- ساخت و پخش کردن داده های تراکنش های ثبت شده

```
app.post('/api/transact', (req, res) => { });
```

۴- اطلاع از تراکنش های ثبت شده و آماده به ماین در شبکه

```
app.post('/api/transaction-pool-map', (req, res) => { });
```

و در نهایت توسط متد listen، تمامی گره ها با اجرای برنامه بر روی پورت مورد نظر شروع به فعالیت و اتصال به یکدیگر میکنند.

```
app.listen(PORT, ( ) => { });
```

فایل pubsub.js

در فایل pubsub.js، شروع به ساخت کانال (channel) برای ارسال و دریافت اطلاعات میکنیم.

رویکرد کلی در این فایل استفاده از دیتابیس redis برای ساخت کانال هایی بر بستر شبکه است به گونه ای که هر زمان گره ای به شبکه اضافه شود، میتواند با مابقی گره ها از طریق این کانال ها در ارتباط باشند. آنها وقتی اطلاعاتی را روی این کانال ها ارسال کنند، هر گره ای که عضو این کانال ها باشد (subscribe کرده باشد) میتواند آن اطلاعات را دریافت کند. مزیت استفاده از این رویکرد این است که گره ها بدون دانستن آدرس گره ی فرستنده، اطلاعات را دریافت میکنند و همچنین گره ی فرستنده بدون دانستن آدرس گره های گیرنده، اطلاعات را تنها بر روی کانال به اشتراک میگذارد.

```
Const redis = require('redis');

Const CHANNELS = {
    BLOCKCHAIN,
    TRANSACTIONS
}
Class PubSub {
    constructor( );

    handleMessage( channel, message );
    subscribeToChannels( );
    publish( channel, message );
    broadcastChain( );
    broadcastTransactions( transaction );
}
```