

## موارد پروژه

این پروژه با استفاده از زبان TypeScript و NodeJS و ReactJS ایجاد شده است.

به دلیل ارور "Request timed out." برای اتصال به وب سوکت بایننس در سمت سرور، مجبور به ایجاد فایل fake-binance-api.ts جهت ایجاد وب سوکت مشابه api بایننس شده.

به دلیل نیاز برنامه به یک دیتابیس real time و زمان کم برای انجام پروژه، در این برنامه از پایگاه داده‌ای استفاده نشده.

## توضیحات پروژه

/server

ابتدا با دستور "npm run start-api" فایل binance-fake-api.ts توسط ts-node اجرا شده و پس از دریافت پیام "fake api run in port: '4433'" میتوان با استفاده از آدرس "ws://localhost:4433" به وب سوکت مذکور دسترسی داشت. این وب سوکت داده‌های موجود در فایل fake-api.json در مسیر utils/data/ در یک interval در هر یک ثانیه ارسال میکند.

با اجرای دستور "npm start" فایل index.ts اجرا شده و پس از ایمپورت‌های مورد نیاز، تابع wsServer که در مسیر utils/ws-server.ts/ صادر شده است را اجرا میکند. wsServer با دریافت مقدار address به صورت کلایت به آدرس دریافت شده (که میتواند آدرس ایجاد شده در فایل binance-fake-api.ts و یا آدرس api بایننس باشد) متصل شده و مسیج‌های ارسال شده را دریافت کند. پس از دریافت مسیج‌ها، میتوان داده‌های آن را در پایگاه داده ذخیره کرد. همچنین داده‌ها دریافت شده با استفاده از wsConnection داده‌ها را به وب سوکت ایجاد شده با wsSr ارسال میکند. تابع wsServer با دریافت مقدار port که به صورت پیشفرض 4343 مقداردهی شده، وب سوکتی ایجاد و داده‌های دریافتی از وب سوکت متصل شده را در وب سوکت ایجاد شده با آدرس "ws://localhost:4343" ارسال میکند.

پس از اجرای wsServer، مقدار دهی اولیه برنامه انجام شده، پوشه public به صورت static مقداردهی و مسیریهای برنامه با استفاده از router صادر شده از فایل routes/routes.ts/ معرفی میشود. در آخر برنامه در پورت 80 اجرا میشود.

در فایل `routes.ts`، برای اجرای درخواست GET به مسیر `"/`، فایل `index.html` در پوشه `views` را ارسال میکند. همچنین برای درخواست GET به مسیر `"/json/fake-data/"`، داده‌های موجود در فایل `fake-api.json` را به کاربر پاسخ میدهد.

`/client`

در فایل `index.tsx` برنامه `react` ایجاد شده و کامپوننت `App` را اجرا میکند. کامپوننت `App` با استفاده از پکیج `mui`، تب‌های مورد نیاز را در صفحه ایجاد و آن‌ها را کنترل میکند.

در تب اول `"my server ws"`، کامپوننت `WsChart` اجرا و به آدرس وب سوکت ساخته شده در فایل `server/index.ts` متصل شده و داده‌های دریافتی را نمایش میدهد.

تب `"binance api"`، کامپوننت `BinanceWs` را اجرا و مقادیر دریافتی را نمایش میدهد.

تب `"feach data"`، کامپوننت `FeachChart` را اجرا کرده و با درخواست GET کاربر، داده‌ها را از مسیر `"http://localhost/json/fake-data"` دریافت و نمایش میدهد.

کامپوننت `Charts`، با دریافت مقدار `data`، با استفاده از پکیج `react-apexcharts` داده‌ها را در چرات نمایش میدهد.