İÇİNDEKİLER

PROJEYI AYAĞA KALDIRMAK

Adım 0:
Ayağa kaldırılacak ortamdan bağımsız olarak projede kullanılan üçüncü parti kütüphanelerin indirilmesi gerekir.

npm install

NPM SCRIPTS

```
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint src --ext js,jsx --report-unused-disable-directives
--max-warnings 0",
    "preview": "vite preview",
    "launchServer": "node bridgeServer.js",
    "custom": "vite build --mode custom",
    "previewCustom": "vite --mode custom",
    "localSetup": "npm run custom && npm-run-all --parallel launchServer
previewCustom"
  },
```

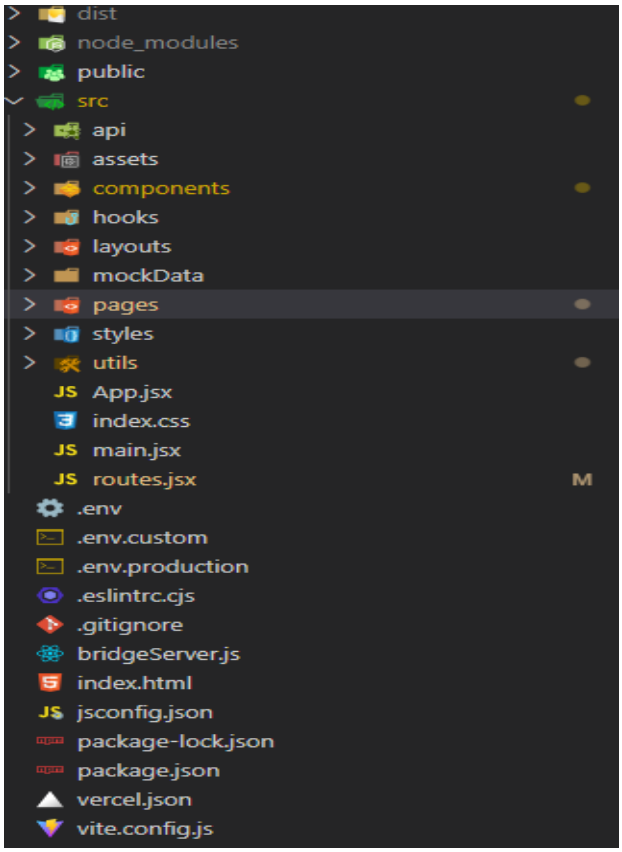
KOMUTLAR

dev: lokal serverı başlatır**.** http://localhost:5173/ adresinden projenin çıktısına ulaşılır.
build: production build oluşturur. Oluşan build **dist** dosyasının altında saklanır.
preview: production buildi lokalde çalıştırır. http://localhost:4173/ üzerinden erişilir.
launchServer: bridgeServer ‘ ı çalıştırır.
custom: *custom* moduna göre build alır**, .env.custom** dosyasındaki verileri kullanır.
previewCustom: *custom* moduna göre alınmış olan buildi lokalde çalıştırır.
localSetup: *custom*, *launchServer* ve *previewCustom* komutlarını tek seferde çalıştırıp, projeyi ayağa kaldırır.

- **localSetup** komutuyla Dev. serverın başarılı bir şekilde ayağa kalkabilmesi için **.env.custom** dosyasındaki VITE_SOURCES_API_URL ve VITE_HEADLINES_API_URL adreslerinin bağlı olunan ağın IPV4 adresi ile değiştirilmesi gerekiyor.

  Bkz: önemli notlar, bridgeServer.js, mockData

KLASÖR YAPISI



dist

- *Production build* sonrası yayına alınmaya hazır dosyalar burda tutulur.

api

- Proje genelinde yapılan HTTP requestler bu klasörün altında ele alınır.

assets

- Proje genelinde kullanılan iconlar bulunur**.**

components

- Proje genelinde kullanılan *reusable* componentler bu klasörün altında bulunur.

hooks

- *Custom hook* lar bu klasörün altında bulunur.

layouts

- *Layout* componentleri bu klasörün altında bulunur

mockData

- *Newsapi'nin* developer sürümünde gün içinde atılabilen requestlerin bir sınırı bulunuyor.Bu sınıra takılmadan geliştirme yapabilmek için gerekli verilerin tutulduğu dosya.

pages

- Bu klasörün altındaki her bir dosya adı aynı isimdeki bir route a karşılık gelir.

  Örnek:
  https://btcturk-case.vercel.app/main adresine gidildiğinde
  src\pages\main\index.jsx componenti çalışır.

- Proje genelinde kullanılmayan, sadece ilgili sayfada kullanılan componentler route adı altındaki /components dosyasının altında bulunur.

  Örnek:
  Sadece anasayfada kullanılan componentler **src\pages\main\components** dizininde bulunur.

styles

- Proje genelinde kullanılan mixinler bu klasörün altında bulunur.

utils

- Proje genelinde kullanılabilecek yardımcı JavaScript fonksiyonları burda bulunur.

  Örnek:
  src\utils\date.js → Tarih ile ilgili işlemler için
  src\utils\string.js → String ile ilgili işlemler için

routes.jsx

- Hangi route için hangi componentin kullanılacağı bu dosyada belirlenir.

.env.[MODE]

- Ortam bazlı değişen ortam değişkenleri bulunur.

bridgeServer.js

Newsapi ücretsiz sunduğu sürümde sadece **localhost** üzerinden atılan isteklere karşılık verdiğinden, aynı ağ üzerindeki diğer cihazlarda proje test edilemiyor. Bu problemin önüne geçebilmek için lokalde çalışan köprü görevi gören bir sunucu oluşturdum.

Tarayıcı üzerinden atılan istekler sırasıyla bu yolu izliyor;

Request : Tarayıcı → lokal sunucu(bridgeServer.js) → newsapi
Response: newsapi → lokal sunucu → tarayıcı

****Bu dosyanın çalışması için projenin *npm run localSetup* komutu ile ayağa kaldırılmış olması gerekiyor.**

BridgeServer.js aynı zamanda bir sunucuya da deploy edildi.

Bu sayede https://btcturk-case.vercel.app/ üzerinden atılan istekler önce deploy edilen sunucuya sonrasında newsapi nin sunucularına ulaşıp sayfanın ilgili verilerle doldurulması sağlandı.

BridgeServer çalışma senaryoları:

1. Günlük kullanım limiti dolmadıysa newsapi den çekilen verileri döner.
2. Limit dolduysa statik olarak tutulan mockData klasörünün altındaki verileri döner.

****Eğer günlük limit dolduysa, haber kaynağında seçilen başlıktan bağımsız olarak listeleme sayfasında src\mockData\headlines.js dizininde bulunan veriler gösterilir.**

KULLANILAN KÜTÜPHANELER

slick: Haberlerin listelendiği sayfadaki slider için kullanıldı.

classNames: Bir elemente koşula bağlı eklenen css classlarının daha rahat bir şekilde organize edilmesi için kullanıldı.

- Örnek için **src\components\FilterButton\FilterButton.jsx** componenti incelenebilir.

tanstack/react-query: Server statenin cachelenip, istenen aralıklarda veri çekilebilmesi için kullanıldı.

react-responsive: Cihazın ekran boyutunu almak için kullanıldı.

- Örnek için **src\hooks\useGetDeviceType.js** incelenebilir.

ÖNEMLİ NOTLAR

Newsapi’nin ücretsiz kullanımındaki kısıtlar

- Günlük kullanım limiti
- Sadece localhost üzerinden yapılan istekleri kabul etmesi

Bu kısıtların önüne geçebilmek için bridgeServer oluşturuldu.

Production URL: https://btcturk-case.vercel.app/