REAL ESTATE

Create .babelrc:

|  |
| --- |
| { |
|  | "presets": ["next/babel"], |
|  | "plugins": [] |
|  | } |

Dependencies Used:

npm i

1. @chakra-ui/react @emotion/react @emotion/styled (Chakra UI pear dependencies)
2. axios
3. framer-motion
4. mollify (large numbers into human readable form)
5. nprogress (progress bar)
6. react-horizontal-scrolling-menu (for images)
7. react-icons

ChakraUi

<Flex> -> flex

<Box> -> div

ChakraUI:

It is used to avoid writing css expilicitly. To make sure it works fine, we need to wrap our app inside ChakraProvider

import '../styles/globals.css'

import Router from 'next/router'

import Head from 'next/head'

import NProgress from 'nprogress'

import { ChakraProvider} from '@chakra-ui/react'

import Layout from '../components/Layout'

function MyApp({ Component, pageProps }) {

  return(

    <>

    <Head>

    </Head>

    <ChakraProvider>

      <Layout>

          <Component {...pageProps} />

        </Layout>

    </ChakraProvider>

    </>

  )

}

export default MyApp

Using or fetching data from Api:

Create utils folder at root:

import axios from 'axios';

export const baseUrl = 'https://bayut.p.rapidapi.com'

export const fetchApi = async (url) => {

    const {data} = await axios.get(url, {

        headers: {

        'x-rapidapi-host': 'bayut.p.rapidapi.com',

        'x-rapidapi-key': '530cf11fabmsh79e60c0e05c7b69p139c05jsna753fd49ae56'

        }

    })

    return data;

}

These headers are provided by rapidApi. Now we can call fetchApi function anywhere we want in getStaticProps function (as it is nextJs) and pass url and it will fetch data and we can map on that data and extract useful information from it.

Now extracting data from fetchApi function in index.js:

At the bottom,

export async function getStaticProps () {

  const propertyForSale = await fetchApi(`${baseUrl}/properties/list?locationExternalIDs=5002&purpose=for-sale&hitsPerPage=6`);

  const propertyForRent = await fetchApi(`${baseUrl}/properties/list?locationExternalIDs=5002&purpose=for-rent&hitsPerPage=6`);

  return {

    props: {

      propertiesForSale: propertyForSale?.hits,

      propertiesForRent: propertyForRent?.hits,

    }

  }

}

Now we can use these props and map over them