BoroBazar Documentation

Introduction

Fastest E-commerce template built with `React`, `NextJS`, `TypeScript`, `React-Query` and `Tailwind CSS`. Its very easy to use, we used `react-query` for data fetching . You can setup your api endpoint's very easily and your frontend team will love using it.

Requirements

- node(12.13.0 or later)
- yarn(version 1)
- editor: Visual Studio Code(recommended)

Tech We Have Used

Tech specification for this template is given below

- React
- NextJs
- TypeScript
- React Query
- Axios
- Tailwind CSS
- Stripe Public Key

Getting Started & Installation

For getting started with the template you have to follow the below procedure. First navigate to the `borobazar` directory.

Step 1 : Configure your env file

Within the project directory you'll find a `.env.local.template` file just rename it as `.env.local`.

** NOTE : ** This file contain `env values` for local development but when you wanna use this template for your needs you need to replace this value with `your own real API endpoint`.

** NOTE : ** To get the map, go to your `.env.local` file and put your google map api key there like `NEXT_PUBLIC_GOOGLE_API_KEY= put your api key`

** NOTE : ** For Stripe Payment Integration, go to your `.env.local` file and put your stripe public api key there like `NEXT_PUBLIC_STRIPE_PUBLIC_KEY= put your stripe public key`

Step 2 : Running the project

Run below command for getting started with this template.

```
# on borobazar directory
$ yarn
$ yarn dev # which will running the template for development
```

If you want to test your production build in local environment then run the below commands.

```
# build for production
yarn build

#start template in production mode
yarn start
```

Folder Structure & Customization

- To setup you site's basic information like **[Logo,Site title,Description, Menus,etc]** go to -> `src/settings/site-settings.ts` file
- To customize tailwind configuration go to -> `tailwind.config.js` file .
- `/public`: To change your `api data, favicon, multi-language assets (images, placeholder)` etc here .
- `/src/components`: This folder contains all the template related ui components .
- `/src/containers`: This folder contains all the common sections related components.
- `/src/contexts`: This folder contains all necessary context for this template . Like `cart, ui` etc.
- `/src/framework`: It's contain all the data fetching related codes. see below for more info.
- `/src/pages`: All the pages created here which is used by nextjs routing mechanism .
- `/src/settings`: To setup your site basic setting like `privacy page, terms page, faq page` etc.
- `/src/styles`: Overwrites some third party packages CSS files and our custom CSS in the tailwind.css file.
- `/src/utils` : This folder contains `hooks, routes, scrolls, local storage` etc.

Multi-Language

We have used next-i18next(https://github.com/isaachinman/next-i18next) package for supporting multi-language.

- `/public/locales`: This folder contains all languages files. If you want to add more languages, please add your language specific folder.

RTL

- `/src/utils/get-direction.ts`: This file contains all RTL related codes. Change it according to your need.

Data Fetching

For this template we didn't provide any actual rest api integration. We have used `react-query` ~~ hook pattern ~~ and fetched data from public json. You will need to edit those hook to integrate your actual API end point. Please go to `framework/basic-rest/` folder for those hooks.

- Creating the hook.
  - We have imported the `Product` type from `@framework/types` = `framework/basic-rest/types` (We have used typescript path aliasing for this. For more info please see our `tsconfig.json` file). Customize it according to your product type.
  - We have built an `axios instance` which called `http`.
  - We have put all ours endpoint at `@framework/utils/api-endpoints` file using constant value. Customize it according to your api endpoints.
  - We have built our `product hook` using `react-query` .

```
import { Product } from '@framework/types';
import http from '@framework/utils/http';
import { API_ENDPOINTS } from '@framework/utils/api-endpoints';
import { useQuery } from 'react-query';

export const fetchProduct = async (_slug: string) => {
  const { data } = await http.get(`${API_ENDPOINTS.PRODUCT}`);
  return data;
};
export const useProductQuery = (slug: string) => {
  return useQuery<Product, Error>([API_ENDPOINTS.PRODUCT, slug], () =>
    fetchProduct(slug)
  );
};
```

For more information about `react-query` please visit React Query.

** NOTE : ** We didn't provide all the endpoints to avoid some unnecessary boiler plate. You will need to customize or build according to your need.

- Using the hook

```
const { data, isLoading, error } = useProductQuery(slug as string);
```

Configuration & Deployment

vercel.com

If you want to host the template in vercel.com then follow the below procedure

- Navigate to `borobazar`
- Put your api endpoint at `vercel.json` file.

- Now run below command

```
vercel
```

NOTE: for deploying to `vercel` using terminal you need to install `vercel-cli` on your machine for more information please visit here

for other hosting provider please follow below url

NextJs Application Deployment