PickBazar Documentation

Introduction

Fastest E-commerce template built with React, NextJS, TypeScript, Apollo, React-Query, NestJs & Tailwind. Its very easy to use, you can build your schema very easily. GraphQL playground makes it's own documentation, and for rest we use swagger, your frontend team will love using it.

Demo Links

1. PickBazar Shop GraphQL
2. PickBazar Shop REST
3. PickBazar Admin GraphQL
4. PickBazar Admin REST
5. PickBazar API GraphQL Playground
6. PickBazar API REST Playground

Tech We Have Used

We have used monorepo folder structure with Yarn Workspace. In our template we have three different part Admin Dashboard, Shop and API. Tech specification for specific part is given below

Admin Dashboard (GraphQL)

- NextJs
- Apollo
- GraphQL Let
- Tailwind
- Typescript
- React Hook Form

Admin Dashboard (REST)

- NextJs
- React-Query
- Axios
- Tailwind
- Typescript
- React Hook Form

Shop (GraphQL)

- NextJs
- Apollo
- GraphQL Let
- Typescript
- Tailwind
- Stripe Integration
- React Hook Form

Shop (REST)

- NextJs
- React-Query
- Axios
- Typescript
- Tailwind
- Stripe Integration
- React Hook Form

API

- NestJs GraphQL
- NestJs REST
- Class Transformer
- Class Validator

Getting Started & Installation

For getting started with the template you have to follow the below procedure. First navigate to the `root` directory. Then run below command for getting started with specific part.

```
# on root directory
yarn
```

API (NestJs GraphQL)

1. Go to `/api/graphql` directory and run below command.

```
# on api/graphql directory
yarn
yarn start:dev
```

NOTE : This will start the server at `http://localhost:4000` for graphql api and you can access it from browser at `http://localhost:4000/graphql`(for graphql playground).

API (NestJs REST)

1. Go to `/api/rest` directory and run below command.

```
# on api/rest directory
yarn
yarn start:dev
```

NOTE : This will start the server at `http://localhost:5000/api/` for rest api and you can access it from browser at `http://localhost:5000/docs`(for rest swagger doc).

Admin(GraphQL)

For starting the admin dashboard part with corresponding api data run below commands.

1. Go to `/admin/graphql` folder.
2. Copy the contents of `.env.template` into a new file called `.env`
3. Run below command to start the server.

```
# on root directory
yarn dev:admin-gql
# or
# on admin/graphql directory
yarn dev
```

Admin(REST)

For starting the admin dashboard part with corresponding api data run below commands.

1. Go to `/admin/rest` folder.
2. Copy the contents of `.env.template` into a new file called `.env`
3. Run below command to start the server.

```
# on root directory
yarn dev:admin-rest
# or
# on admin/rest directory
yarn dev
```

Shop

For starting the shop part with corresponding api run below commands.

1. Go to `/shop` folder.
2. Copy the contents of `.env.template` into a new file called `.env`
3. Fill the `.env` file with your values like, `NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=, SECRET=, GOOGLE_CLIENT_ID=, GOOGLE_CLIENT_SECRET=`
4. SET `FRAMEWORK_PROVIDER='yourProjectType'` in `.env` file. ex: if you want to use rest then `FRAMEWORK_PROVIDER='rest'` or `FRAMEWORK_PROVIDER='graphql'`
5. SET `tsconfig.json`. if you want to use `rest version` then copy the content from `tsconfig.rest.json` to `tsconfig.json` file or if you want to use `graphql version` then copy the content from `tsconfig.graphql.json` to `tsconfig.json` file.
6. Run below command to start the server.

```
# on root directory

yarn dev:shop-gql
yarn dev:shop-rest

# or
# on shop directory
yarn dev:rest
yarn dev:gql
```

NOTE : API must be running for the above commands to work.

NOTE : `.env` file must be filled with your values.

Only For Windows Users

```
- Go to your specific project root and find `.graphql-let.yml`
- Replace the env variable `${NEXT_PUBLIC_GRAPHQL_API_ENDPOINT}` in schema
field manually
- Provide your API url in the schema field.
- Also change the dev command at `package.json`
- Go to `shop/package.json` file and change the scripts:

"codegen": "node -r dotenv/config $(yarn bin)/graphql-let",
to
"codegen": "graphql-let",
```

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

Admin(GraphQL)

1. Go to `/api/graphql` folder.
2. Run `yarn build`
3. Run `yarn start:prod`
4. Go to `/admin/graphql` folder.
5. Run `yarn build`
6. Run `yarn start`

NOTE : API must be running for the above commands to work.

NOTE : `.env` file must be filled with your values.

Admin(REST)

1. Go to `/api/rest` folder.
2. Run `yarn build`
3. Run `yarn start:prod`
4. Go to `/admin/rest` folder.
5. Run `yarn build`
6. Run `yarn start`

NOTE : API must be running for the above commands to work.

NOTE : `.env` file must be filled with your values.

Shop(GraphQL)

1. Go to `/api/graphql` folder.
2. Run `yarn build`
3. Run `yarn start:prod`
4. Go to `/shop` folder.
5. Run `yarn build:gql`
6. Run `yarn start`

NOTE : API must be running for the above commands to work.

NOTE : `.env` file must be filled with your values.

Shop(REST)

1. Go to `/api/rest` folder.
2. Run `yarn build`
3. Run `yarn start:prod`

4. Go to `/shop` folder.
5. Run `yarn build:rest`
6. Run `yarn start`

NOTE : API must be running for the above commands to work.

NOTE : `.env` file must be filled with your values.

Folder Structure & Customization

`/admin` : In this portion all the admin dashboard related coding and functions.

`/shop` : All the shop related coding and functions.

`/api` : API related code for both graphql and rest.

Configuration & Deployment

API

- Navigate to `/api/{yourStack}`
- put your domain url in `src/common/constants.ts` file(if your stack is rest)
- Now host any where like any nodejs application.

vercel.com

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

Admin and Shop

- After deploying the api you will get the api endpoint url. Put that url in the `/yourProject/yourStack/.env` ex: `/admin/graphql/.env`
- also need to put it within `vercel.json` .

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT={put_your_api_url_here}/graphql;
NEXT_PUBLIC_REST_API_ENDPOINT={put_your_api_url_here}/api/;
```

- Now run below command(Within the project folder)

```
vercel
```

```
vercel
```

NOTE: for deploying to `vercel` 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