**React Proje Klasör Yapısı Belgelendirmesi**

Bu belgede, "Template" adlı React projesinin klasör yapısının detaylı açıklamasını sunacağım. Bu yapı, **Proje Yapısı Genel Bakışı:**

[**Link**](https://tree.nathanfriend.io/?s=(%27optiJs!(%27fancy!true~fullPath!fwX~trailYgSlash!true~rootDot!fwX)~sOrce!(%27sOrce!%27KmplaKs6deploymFts0docker-compoX.yml0Dockerfile6FvVKstVdevelopmFtVpGiJ6public0asXt5layOtIpage5sas5staticItheme0locwe5ar_AE4*F_US4*tr_TR4C.html6src0asXts0compJFt5basicILayOtIorgYizim0cJfig5Axios3*i183*MFuqxQyO%20can%20addNoNhZmFu%20from%20herZIregex3cJKxt5rtl-cJKxt3page5HomeILogYIPG0rOK5C3*PrivaKROK3*PublicROK3Xrvice5Home0W9B5pG0W9B5AuthI*LogYIW*9*Bs0util5cJtant5*theme7hemZvwue5type5*baeMFu7ypescripttypesNhat%20cOld%20uXd%20Y%20globw0App3Cqx6package.jsJ%27)~versiJ!%271%27)*%20%20%20%2006*2**C.3qx04I*translaK.jsJ05sI6%5Cn*7qQwlN92FdpoYt3B2model.tCYdexFenGroductI0*JonKteN%20tOouQ%20%2F%2FV0.Fv.W2api3XseYinZe%20q.tswal%01wqZYXWVQONKJIGFCB97654320*)

templates/

├── deployments/

│ ├── docker-compose.yml

│ └── Dockerfile

├── env/

│ ├── .env.test

│ ├── .env.development

│ └── .env.production

├── public/

│ ├── assets/

│ │ ├── layout

│ │ ├── pages

│ │ ├── sass

│ │ ├── static

│ │ └── theme

│ ├── locales/

│ │ ├── ar\_AE/

│ │ │ └── translate.json

│ │ ├── en\_US/

│ │ │ └── translate.json

│ │ └── tr\_TR/

│ │ └── translate.json

│ └── index.html

├── src/

│ ├── assets

│ ├── components/

│ │ ├── basic

│ │ ├── Layout

│ │ └── orginizim

│ ├── configs/

│ │ ├── Axios.tsx

│ │ ├── i18.tsx

│ │ ├── Menu.tsx //you can add to the menu from here

│ │ └── regex.tsx

│ ├── contexts/

│ │ └── rtl-context.tsx

│ ├── pages/

│ │ ├── Home

│ │ ├── Login

│ │ └── Product

│ ├── routes/

│ │ ├── index.tsx

│ │ ├── PrivateRoute.tsx

│ │ └── PublicRoute.tsx

│ ├── services/

│ │ ├── Home/

│ │ │ ├── index.api.tsx

│ │ │ ├── index.endpoint.tsx

│ │ │ └── index.model.ts

│ │ ├── product/

│ │ │ ├── index.api.tsx

│ │ │ ├── index.endpoint.tsx

│ │ │ └── index.model.ts

│ │ └── Auth/

│ │ └── Login/

│ │ ├── index.api.tsx

│ │ ├── index.endpoint.tsx

│ │ └── index.model.ts

│ ├── utils/

│ │ ├── contants/

│ │ │ └── theme.ts //all theme values

│ │ └── types/

│ │ └── baeMenu.ts //all typescripttypes that could used in global

│ ├── App.tsx

│ └── index.tsx

└── package.json

Şimdi, her klasör ve dosyanın detaylarına geçelim:

**deployments/**

Bu klasör, dağıtım ile ilgili yapılandırma dosyalarını içerir. İçinde şunlar bulunur:

* **docker-compose.yml**: Docker konteyner düzenlemesi için hizmetlerin, ağların ve hacimlerin tanımlandığı YAML dosyası.
* **Dockerfile**: React uygulaması için Docker görüntüsü oluşturmak için kullanılan yapılandırma dosyası.

**env/**

Bu klasör, ortama özel yapılandırma dosyalarını içerir. İçinde şunlar bulunur:

* **.env.test**: Test ortamına yönelik çevresel değişkenler.
* **.env.development**: Geliştirme ortamına yönelik çevresel değişkenler.
* **.env.production**: Üretim ortamına yönelik çevresel değişkenler.

**public/**

Bu klasör, uygulamanın halka açık varlık ve kaynaklarını içerir. İçinde şunlar bulunur:

* **assets/**: Resimler, videolar vb. gibi statik varlıkların depolanması için bir dizin.
* **locales/**: Uluslararasılaştırma için çeviri dosyalarını içerir.
* **index.html**: React uygulamasına giriş noktası olarak hizmet veren ana HTML dosyası.

**src/**

Bu, React uygulamasının kalbi olan kaynak kodunu içerir. Aşağıdaki klasörleri içerir:

**assets/**

Bu klasör, uygulama için gereken proje özel varlıkları depolamak için kullanılır.

**components/**

Bu klasör, yeniden kullanılabilir tüm React bileşenlerini alt dizinlere göre düzenler:

* **basic/**: Uygulama genelinde yeniden kullanılabilen temel bileşenleri içerir.
* **Layout/**: Uygulamanın genel düzeni ile ilgili bileşenleri içerir.
* **orginizim/**: Projeye özgü bileşenleri düzenler.

**configs/**

Bu klasör, yapılandırma dosyalarını, yardımcı programları ve sabitleri içerir:

* **Axios.tsx**: API istekleri yapmak için kullanılan popüler HTTP istemcisi Axios için yapılandırmayı içerir.
* **i18.tsx**: i18next veya benzeri kütüphaneleri kullanarak çeviriler için uluslararasılaştırma yapılandırmasını içerir.
* **Menu.tsx**: Uygulamanın gezinme menüsünü kolayca özelleştirmek için menü yapılandırmasını içerir.
* **regex.tsx**: Uygulama genelinde kullanılan düzenli ifadeler veya kalıpları içerir.

**contexts/**

Bu klasör, global durum yönetimini kolaylaştıran React bağlam sağlayıcılarını veya kancalarını içerir:

* **rtl-context.tsx**: Sağdan sola dilleri desteklemek için bağlam sağlayıcı içerir.

**pages/**

Bu klasör, uygulamanın bireysel sayfalarının düzenlendiği yerdir:

* **Home/**: Ana sayfayla ilgili bileşenler ve mantığı içerir.
* **Login/**: Giriş sayfasıyla ilgili bileşenler ve mantığı içerir.
* **Product/**: Ürün sayfasıyla ilgili bileşenler ve mantığı içerir.

**routes/**

Bu klasör, yönlendirme ile ilgili bileşenleri içerir:

* **index.tsx**: Tüm uygulama rotalarını merkezi olarak tanımlayan yönlendirme yapılandırmasını içerir.
* **PrivateRoute.tsx**: Kimlik doğrulama gerektiren özel rotalar oluşturmak için bir yüksek düzey bileşeni içerir.
* **PublicRoute.tsx**: Tüm kullanıcılara açık genel rotalar oluşturmak için bir yüksek düzey bileşeni içerir.

**services/**

Bu klasör, hizmetle ilgili dosyaları ve API ile ilgili mantığı içerir:

* **Home/**: Özelliklere göre düzenlenmiş ana sayfanın hizmetleri ile ilgili bileşenleri içerir.
  + **index.api.tsx**: "Home" özelliği için API çağrılarını yöneten servis dosyası.
  + **index.endpoint.tsx**: "Home" API çağrıları için uç nokta yapılandırması.
  + **index.model.ts**: "Home" özelliği için veri modelleri.
* **product/**: Özelliklere göre düzenlenmiş ürün hizmetleri ile ilgili bileşenleri içerir.
* **Auth/**: Özelliklere göre düzenlenmiş kimlik doğrulama hizmetleri ile ilgili bileşenleri içerir.
  + **Login/**: Giriş hizmeti ile ilgili bileşenler ve mantığı içerir.

**utils/**

Bu klasör, doğrudan bileşenlerle veya hizmetlerle ilgisi olmayan yardımcı dosyaları içerir:

* **constants/**: Tema yapılandırmaları gibi sabit değerleri içerir.
* **types/**: Uygulama genelinde kullanılan TypeScript tür tanımlarını içerir.

**App.tsx**

React uygulamasının kökü olarak hizmet veren ana bileşen. Yönlendirmeyi yönetir ve diğer tüm bileşenler için bir konteynır sağlar.

**index.tsx**

Uygulamanın giriş noktası, App bileşenini render eder ve React uygulamasını yapılandırır.

**package.json**

Bağımlılıkları, komut dosyalarını ve diğer proje meta verilerini içeren standart Node.js paket yapılandırma dosyası.

Bu, React proje klasör yapısı belgelendirmesini tamamlar. Modülerlik, düzen ve ölçeklenebilirlik teşvik edilerek, geliştiricilerin işbirliği yapmasını ve uygulamayı zaman içinde sürdürmesini kolaylaştırır.