

## 1.BÖLÜM ALINAN NOTLAR

1) Nuxt js route işlemlerini pages klasörünün düzenine göre gerçekleştirir.Örneğin pages içerisinde products isimli bir klasör olsun bu klasör içerisinde yer alan index.vue dosyasına /products pathi yardımıyla erişebiliriz.

2) Parametre göndermek istersek \_id isimli bir klasör oluşturabiliriz burada \_id ismi değişebilir yani \_listId \_productId gibi isimler alabilir. \_id isimli klasörün içerisinde oluşturduğumuz index.vue dosyası ise pathe parametre geldiğinde ekranda gösterilecek olan dosyadır. \_id klasörünün altında başka bir dosya daha oluştursak örneğin productDetail olsun bu sayfa ise /products/ \_parametreleGelicekId/ productDetail olur bu sayede o sayfa içerisinde de id yi kullanabiliriz.

Bunun dışında id parametresi aldığında gösterilecek dosya için products klasörü içerisinde \_id.vue isimli bir dosya oluştursak bu dosya paramtre geldiğinde çalışacak olan dosyadır. (İsmi \_listId.vue,\_productId.vue gibi isimler olabilir dosyanın isminin değişmesi parametreye erişilirken kullanılacak ismi değiştirir.)

3) İd yi kullanmak içinse template içerisinde \$route.params.id(buradaki isim id klasörünün ismine göre değişir.).Script içerisinde ise this.\$route.params.id ile kullanılabilir.

4) Bunlar dışında export default içerisinde validate() isimli bir fonksyon tanımlayarak belirli şarta bağlı olarak (fonksiyonun true veya false dönmesi) sayfa yerine hata sayfası ekranda gösterilebilir.

5) <nuxt-link> yardımı ile route yönlendirmelerini de dosya hiyerarşisine bağlı kalıcak şekilde gerçekleştirebiliriz. (router-link gibi çalışır.) Ancak <a> etiketinden farkı <nuxt-link> Server tarafından render edilmez.Bu nedenle a etiketi ile yapılan yönlendirmeler sayfayı refresh ederken <nuxt-link> SPA mantığı ile çalışır.

6) Script tarafında bir fonksiyon yardımıyla yönlendirme yapmak istersek ise

this.\$router.push("/link"), this.\$router.push("/link/"+this.\$route.params.id) şeklinde yönlendirmeler yapabiliriz.

7) <nuxt-child>

nuxt-child bizim için child pageslerin parent componente göre nerede gösterileceğini belirler.

Bunu bir örnekle açıklamak gerekirse pages ana dizininde products.vue isimli bir dosya oluşturalım. Ancak products isimli bir klasör pages içerisinde bulunsun.

Bu sayede products.vue dosyasında <nuxt-child> ile beraber bütün products klasörünün altında bulunan pageler için bir layout oluşturma şansımız olur.

Sadece pages klasörünün ana dizininde oluşturulan ve klasörle aynı isimde olan dosyalar için bu işlem çalışır.

**NOT:Pages ana dizininde bulunan index.vue dosyası / path de ana sayfada gösterilir.**

## LAYOUTLAR

-Layoutları kullanmak için ana dizinde layout isimli bir klasör içerisinde bu layoutları tanımlamamız gerekmektedir.

-Layoutları bir page içerisinde kullanmak için parent componentte tanımlamak gerekmektedir bu sayede bütün child componentlere etki etmiş olur.

Tanımlama işlemi;

export default içerisinde layout:"layoutAdı" şeklinde olur.

-layouts klasörü içerisinde error.vue isimli bir layout tanımlarsak bu layout bizim sayfalar hata verdiğinde gösterilecek hata sayfamız olur.

NOT:Global olarak bir css tanımlamak için assets klasörü içerisinde bir css dosyası oluşturulur ve bu css dosyasının pathi nuxt.config.js dosyasında bulunan css altına yazılır. Path tanımlarken “~/assets/css/style.css” gibi bir path kullanılır.

Layoutlarda css ler scope olarak tanımlanmamalıdır.Eğer scope yazılırsa child componentlerle beraber çalışmazlar.

- Component tanımlama işlemleri vue js ile aynıdır.