



React-Session-02

Tarih : 12/11/2020

Konu : Styling in ReactJS

Önemli Noktalar:

1. Nokta:

SASS

Styled Components

2. Nokta:

dependencies

3. Nokta:

eslint

Notlar:

- **React Styling Types**

1. **Inline Styling:** camelCase kullanılıyor. Kullanıcı dövüyorlar 😂
2. **CSS Stylesheet:** normal componentadı.css olarak oluşturup style ayarlarını yazdıktan sonra ana sayfaya import ediyoruz. Import ederken .css yazmayı unutuyoruz.
3. **CSS Modules:** Pek kullanılmaz. SASS da style yapılırken aynı yanlışlıkla aynı isim verilmesi halinde oluşabilecek sorunların önüne geçmek için kullanılır.
import styles from './Card.module.css'; //
değişkene atayarak import ediliyor.

clean code

4. Nokta:

export default

export

5. Nokta:

JSX - { JS }

{ {json} }

style

6. Nokta:

reuseable
component

7. Nokta:

projede paket
kurulumu npm
yada yarn dan
hangisiyle
başladıysa onunla
devam etmeli

ama sen "yarn"
kullan!

8. Nokta:

yarn.lock

9. Nokta:

Version: "4.14.1"
// major-minor-

Kullanımı: {styles.cardWrapper}

Modül kullanılıyorsa değişken tanımlamaların vs.
kolaylık olsun diye genelde camelCase kullanılır.

inspect te incelendiğinde;

class="Card_cardWrapper__tBjr

İçinde bulunduğu component adı: Card

normal class adı : _cardWrapper,

karışıklığı önlemek için random bir değişken:

__tBjr

**sonunda tanımlanan değişken aynı adla başka
bir class tanımlansa bile birbirine
karışmamasını sağlar.**

4. **SASS*****; bilmeden olmaz. Daha fazla kullanılır.
Projede Junior fazla olursa kullanılması sorun
olabilir.

eklemek için güncel version:

\$npm install node-sass@4.14.1

\$yarn add node-sass@4.14.1 ***

5. **Styled Components*****; bir pakettir. Çok sık
kullanılıyor. Bilmeden olmaz. Stilleri component
olarak oluşturmamıza yarar. Card.style.js gibi
JS dosyası olarak tanımlanır. JS dosyası
içerisinde işlem yapılır.

import edilirken best practice **styled** kullanılır.

Proje içerisinde karışıklık olmaması açısından

isim verirken **StyledCardWrapper** gibi

kullanılıyor.

const **StyledCardWrapper = styled.div ` style
buraya yazılır `**

- Takımlar proje öncesi ne kullanılacağına karar verir.
- **devDependencies**; geliştirme aşamasında kullanılan ama canlı projede kullanılmayacak paketler bu şekilde kurulur.
- Kurulumu
\$yarn paketAdı —dev

patch

- **dependencies;** proje için gerekli olan paketler
- **\$yarn install** // tüm proje paketlerinin (dependencies) kurulumu için
- **\$yarn add paketAdı** // paket ekleme
- **eslint:**
- **clean code:**
- **componentleri export** ederken **default** kullanmaz isek import ederken import {Component} from "./....."; olarak yapmamız gerekiyor.
- **componentleri export ederken default** kullandığımızda ise import ederken import Component from "./....."; olarak yapmamız gerekiyor.
- JSX de JS kullanmak için { JS } içerisine almak gerekli
- Style yapmak için JSX için gerekli olan { } içerisine style bir json object olduğu için bir tane daha { JSON } eklenerek yazılır.

```
Camel Case: noticeFirstLetters
Pascal Case: InculdingFirstWord
Snake Case: notice_btwn_words
Kebab Case: emojis-are-useful
```

- **inline style:** her seferinde obje olarak render edilir gereksiz hafıza kaplar. Kullanılması gerekiyorsa Pull Request yapılırken açıklanır. Normalde kullanılmaz.
- box-shadow: x y shadow renk
box-shadow: 2px 2px 5px "red";
- yarn.lock: yüklenip yüklenmeyeceğini takım belirler.
Ama genel olarak yüklenir.

- Version: "4.14.1" // major-minor-patch
major: breaking change
minor: kullanımı etkilemeyecek
^ : major version içerisindeki en son minor ve patch paketlerini kurar. Takibe gerek olmadan kurar. Update edilince sorun olabilir. Genel kullanım ^işareti kaldırılarak manuel olarak güncellenir.
~ : patch in en güncel versionunu otomatik günceller
versionun üzerinde mouse ile birkaç saniye beklersek en güncel version u gösteriyor.
- HTML componentler küçük harfle, custom componentse ilk harfi Büyük harfle yazılır.
- Proje yöneticisi tüm elementler styled component olacak diyebilir.
- Component isimlendirmeleri proje başlangıcında belirlenir.



ÖZET:

