

**Node.js** : Node ,JavaScript kodunun serverda çalıştırılmasından tutun paket yönetimi ve bizim için geliştirme ortamında bir yayın ortamı imkanı sunuyor.

### React Component Mimarisi

Biz uygulamamızı componentlere bölüyoruz.Yani component demek bileşen demektir.React bir single page application yani tek sayfa uygulamasıdır.Aslında bizim tek bir sayfamız var.Bu sayfamız aslında bizim için bir component görevi görecek.Ardından ben bu componentin içerisinde örneğin bir menü kısmım var.Oraya onu bir component olarak oluşturuyorum.Sonra sol menüm var,onun içinde bir component oluşturuyorum.Sonra Menü kısmında bir sisteme giriş sistemden çıkış kısmı oluşturmak istiyorum oda bir component.Yani bunun sınırı yok. Bu şekilde sayfalarımızı yapılandırıyoruz.Mesela A ve B componentimiz var. Ben A componentinden B componentine veri taşıyorum . Ama B componentinden A ya taşıyamıyorum. Buna **One-Binding** deriz.

**Virtual Dom** :

### Componentlerin kendi Yapısı

Componentler bir kaç şekilde oluyor.Bunlar Fonksiyon componentler,class componentler birde React Hooks denilen yapı ile oluşturulan componentler var.

**JSX** :

**Props**:props a neden ihtiyac duyuyoruz?

#### NaviComponent

##### Category List

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

##### Product List

Ben CategoryList ve ProductList in title bilgisini NaviComponentten almak istiyorum.Kısacası amacım bir componentten diğerine data taşımak. Bu durumda karşımıza prop çıkıyor. Props(Property's).Peki bunu nasıl yapıcağız?

App.js

```
function App() {
  return (
    <div>
      <Container> You, 2 hours ago • reactstrap eklendi
      <Row>
        <Navi></Navi>
      </Row>
      <Row>
        <Col xs="3">
          <CategoryList title="Category List"></CategoryList>
        </Col>
        <Col xs="3">
          <ProductList title="Product List"></ProductList>
        </Col>
      </Row>
    </Container>
  </div>
)
```

Sonra ProductList.js e git. Constructor ile props ı ekle.Aslında Constructor a gerek yok . this.props ile de erişilebilir.

```
export default class ProductList extends Component {  
  constructor(props){  
    super(props);  
    state: {}  
  }  
  render() {  
    return (  
      <div>  
        <h3>{this.props.title}</h3>  
      </div>  
    )  
  }  
}
```

**Encapsulation:**Yeni bir eleman geldiği zaman child componente gönderilen değeri bir model gibi düşünüp onun içerisinde herhangi bir yerde değişiklik yapmamak için bu işlemi kullanırız.

**State:**Props gibi veri alışverişinde kullanılıyor.Props bir componentten diğerine taşınan data.Bir componentten diğer componentte bir değişken taşımak istiyorum veya bir fonksiyon props kullanırım.State ise bir componentin datasıdır.Bir coponente özel bir data tutmak istiyorsak onu state ile sağlarız. Örneğin CategoryList in datası Category Componentinin içerisinde lazım.Yanı bu data tek bir data ve tamamen Category componente özel bir data.İşte bu durumda kategori datasını app.js (ana component)ten de taşıyabilirim props ile .Ama yok ben bunu kategori içeriisinde direkt tutmak istiyorum,herhangi bir yerden taşımak istemiyorum.

**Eventler:**örn onClick() eventi

```
<ListGroup>  
  {  
    //map : listenın elemanlarını tek tek döner,döndüğü elemanları yeni bir liste yapar.  
    this.state.categories.map(category =>  
      (  
        <ListGroupItem onClick={this.setState({currentCategory:category.categoryName})} key={category.categoryId}>{category.categoryName}</ListGroupItem>  
      ))  
    }  
  }  
</ListGroup>
```

**setState:**Statede ki herhangi bir nesnenin değiştirilmesini sağlar.

**React Yaşam Döngüsü**

Önce componenetler yerleşiyö(componentDidMount) sonra render lar çalışıyor.

## React Routing:

```
<div>
  <h3>
    {this.props.info.title}-{this.props.currentCategory}
  </h3>
  <Container>
    <Row>
      {
        this.state.products.map((product)=>{
          return(
            <Col key={product.id} md="4">
              <Card top style={{ width: "18rem" }}>

                <CardBody >
                  <CardImg
                    top
                    width="50%"
                    src={product.image}
                    alt="Card image cap"
                  />

                  <CardTitle>{product.title}</CardTitle>
                  <CardText >{product.description}</CardText>
                  <Button >Add to Cart</Button>
                </CardBody>
              </Card>
            </Col>
          )
        })
      }

    </Row>
  </Container>
</div>
```