Higher-order components and recompose

Thai Pangsakulyanont (@dtinth)
Front-end architect at Taskworld
ReactJS Bangkok 1.0.0

TASKWORLD™

Today’s session is about…

Higher-order functions
Higher-order components
Refactoring
Object-oriented design
Functional programming
Agility

Warm up exercise

```
const myCart = [
  { name: 'Oreo', unitPrice: 5, quantity: 10 },
  { name: 'Pocky', unitPrice: 18, quantity: 2 },
  { name: 'Magnum', unitPrice: 45, quantity: 3 }
]
```

```
const myCart = [
  { name: 'Oreo', unitPrice: 5, quantity: 10 },
  { name: 'Pocky', unitPrice: 18, quantity: 2 },
  { name: 'Magnum', unitPrice: 45, quantity: 3 }
]
```

```
const myCart = [
  { name: 'Oreo', unitPrice: 5, quantity: 10 },
  { name: 'Pocky', unitPrice: 18, quantity: 2 },
  { name: 'Magnum', unitPrice: 45, quantity: 3 }
]
```

```
const myCart = [
  { name: 'Oreo', unitPrice: 5, quantity: 10 },
  { name: 'Pocky', unitPrice: 18, quantity: 2 },
  { name: 'Magnum', unitPrice: 45, quantity: 3 }
]
```

What is the subtotal?

```
function subtotalOfCart (cart) {






}
```

```
function subtotalOfCart (cart) {
  let sum = 0




}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {

  }

}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }

}
```

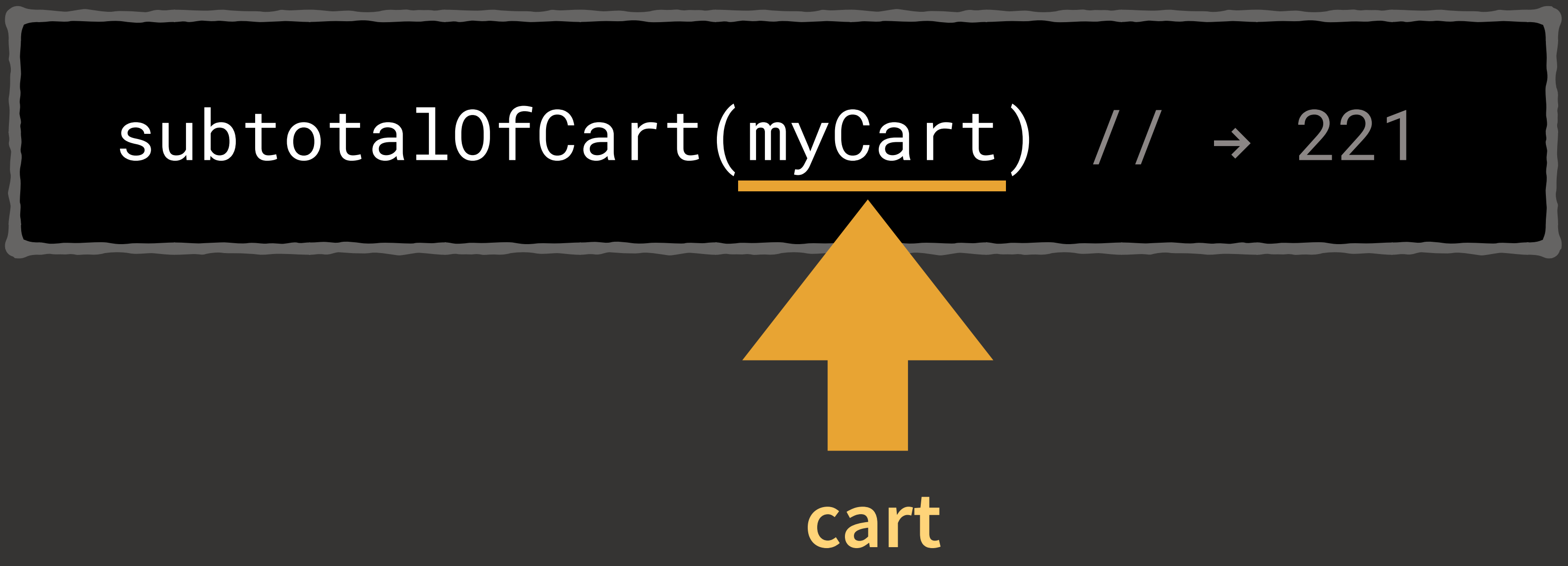
```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}



subtotalOfCart(myCart)
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```

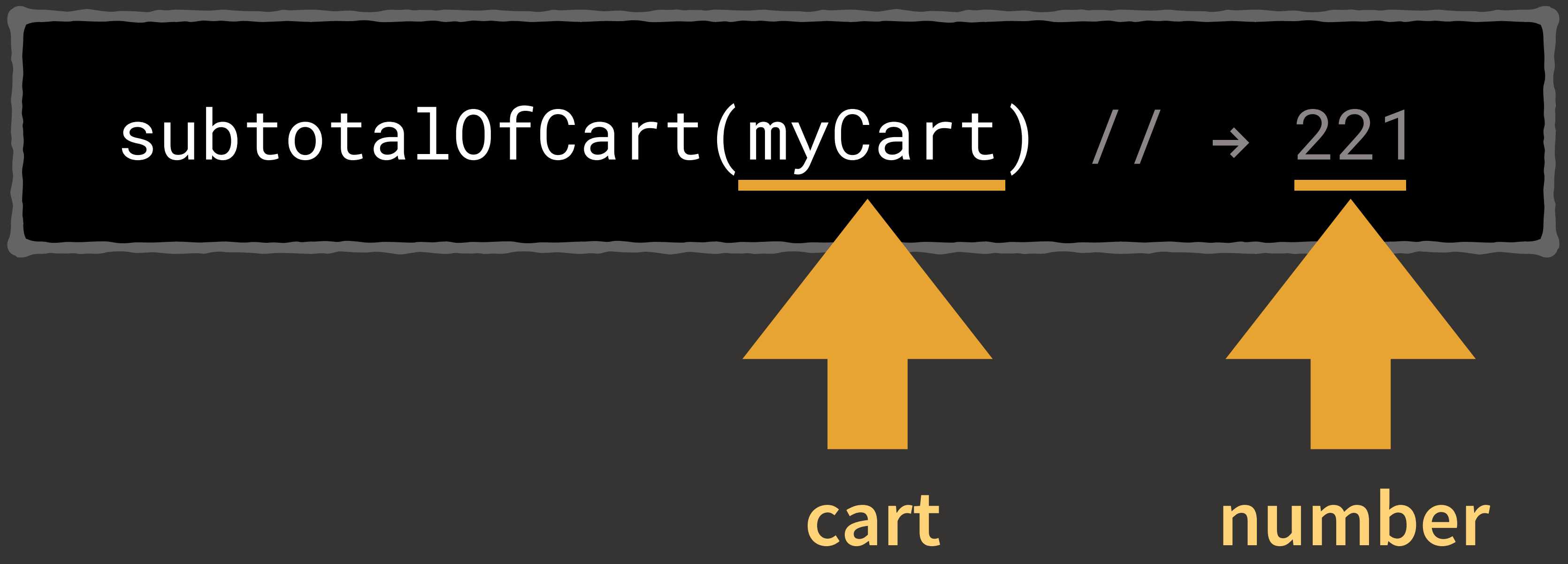
```
subtotalOfCart(myCart) // → 221
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```



```
subtotalOfCart(myCart) // → 221
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```

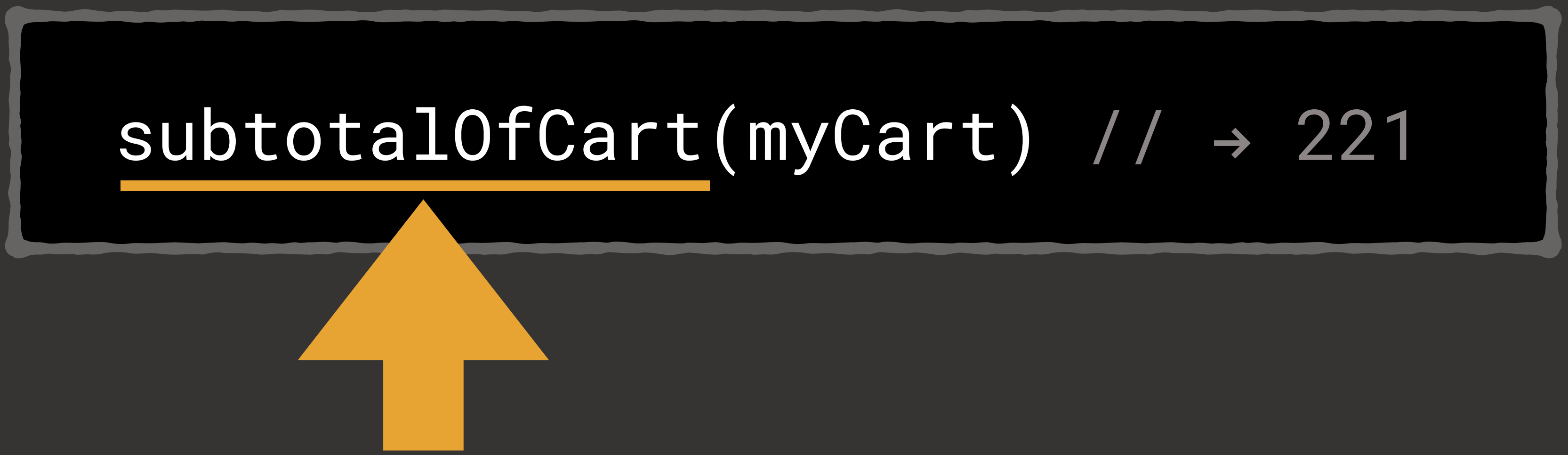


```
subtotalOfCart(myCart) // → 221
```

cart

number

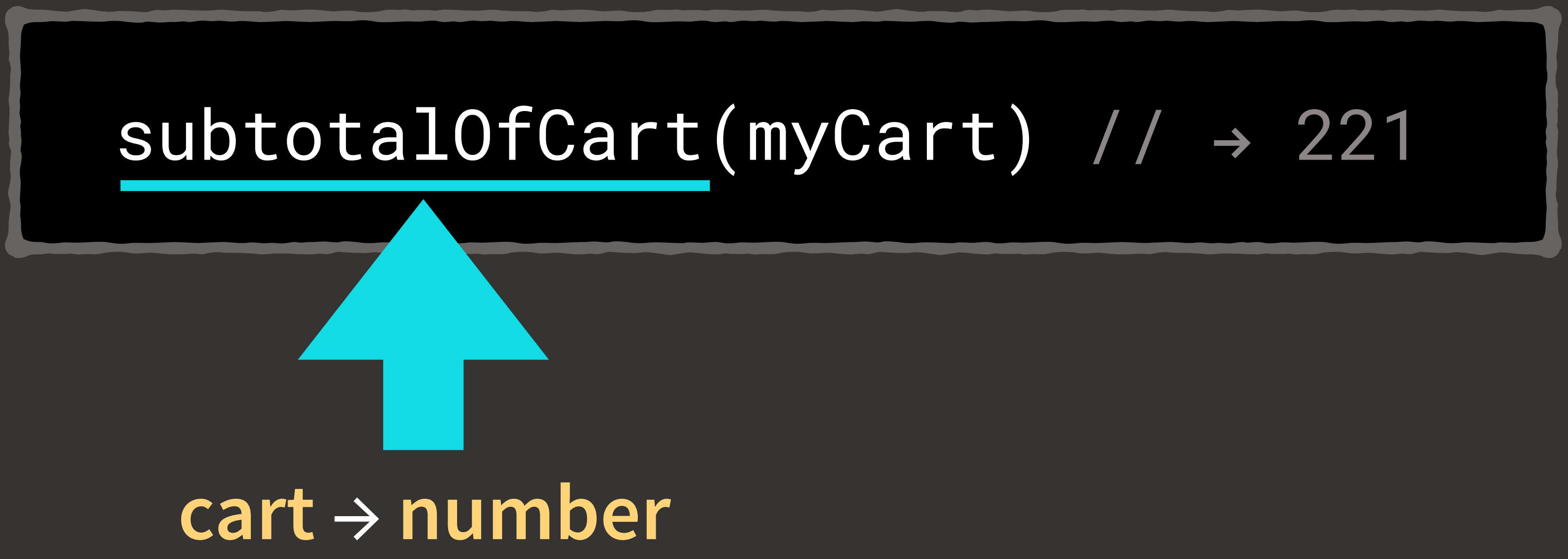
```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```



```
subtotalOfCart(myCart) // → 221
```

a function that takes a **cart** → returns a **number**

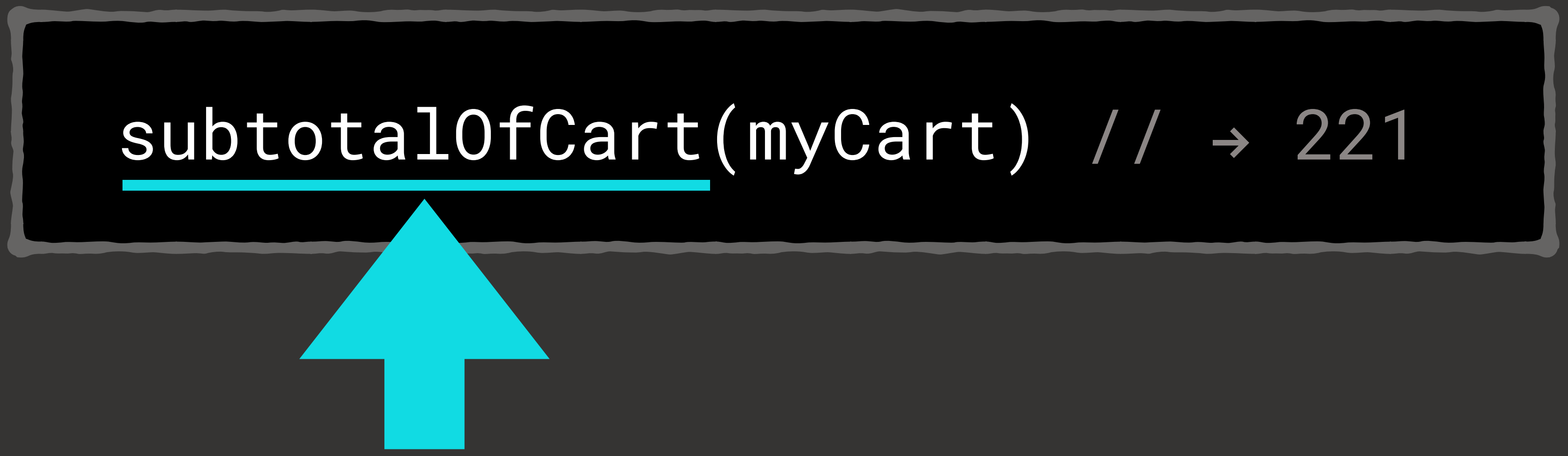
```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```



```
subtotalOfCart(myCart) // → 221
```

cart → number

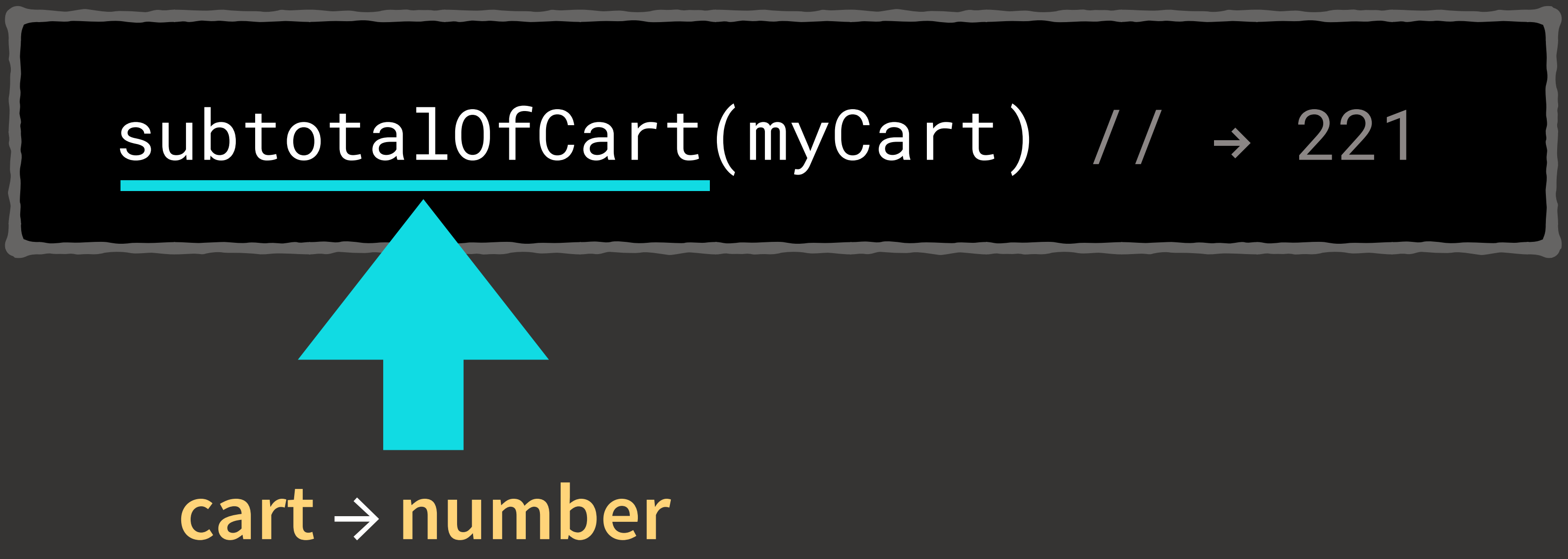
```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```



```
subtotalOfCart(myCart) // → 221
```

Array<item> → number

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```



```
subtotalOfCart(myCart) // → 221
```

cart → number

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}
```

How many items are there in the cart?

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
```

```
numberOfItemsInCart(myCart) // → 15
```

```
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

Duplicated

Parameterize!

One function to rule them all.

```
function calculateCartStatistics (cart, statType) {

}
```

```
function calculateCartStatistics (cart, statType) {

}
```

```
STAT_TYPE_SUBTOTAL
STAT_TYPE_NUMBER_OF_ITEMS
etc.
```

```
function calculateCartStatistics (cart, statType) {
  let sum = 0
  for (let item of cart) {
    switch (statType) {
      case STAT_TYPE_SUBTOTAL:
        sum += item.unitPrice * item.quantity
        break
      case STAT_TYPE_NUMBER_OF_ITEMS:
        sum += item.quantity
        break
    }
  }
  return sum
}
```

```
function calculateCartStatistics (cart, statType) {
  let sum = 0
  for (let item of cart) {
    switch (statType) {
      case STAT_TYPE_SUBTOTAL:
        sum += item.unitPrice * item.quantity
        break
      case STAT_TYPE_NUMBER_OF_ITEMS:
        sum += item.quantity
        break
    }
  }
  return sum
}
```

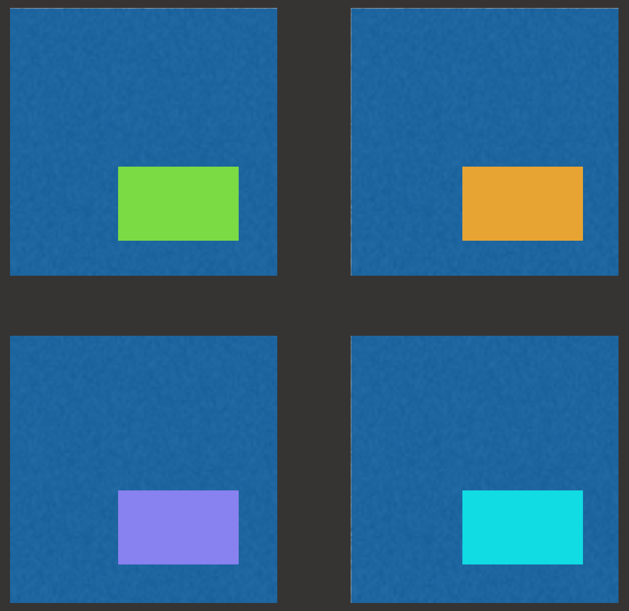
Complicated

Too many responsibility!

Dilemma

Dilemma

Many duplicated functions

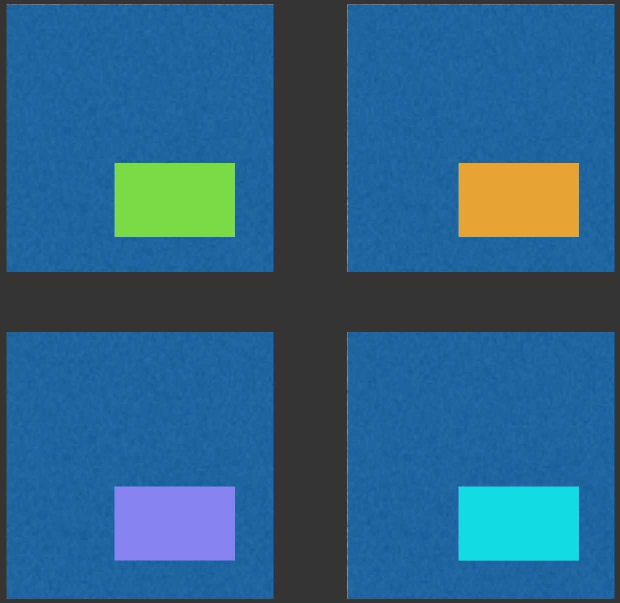


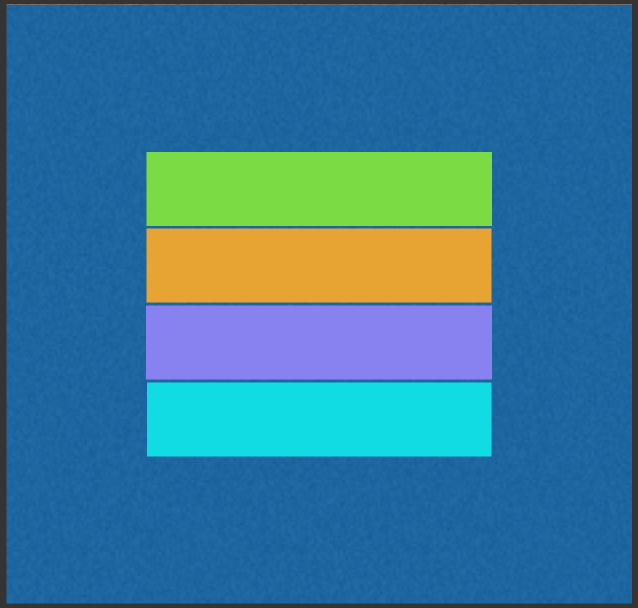
Dilemma

Many duplicated functions

or?

One complicated function





r/reactjs

Two similar React components or one component with a ton of conditionals?

“Duplication is far cheaper than the wrong abstraction”

—Sandi Metz, The Wrong Abstraction
http://www.sandimetz.com/blog/2016/1/20/the-wrong-abstraction

```
function calculateCartStatistics (cart, statType) {
  let sum = 0
  for (let item of cart) {
    switch (statType) {
      case STAT_TYPE_SUBTOTAL:
        sum += item.unitPrice * item.quantity
        break
      case STAT_TYPE_NUMBER_OF_ITEMS:
        sum += item.quantity
        break
    }
  }
  return sum
}
```

```
function calculateCartStatistics (cart, statType) {
  let sum = 0
  for (let item of cart) {
    switch (statType) {
      case STAT_TYPE_SUBTOTAL:
        sum += item.unitPrice * item.quantity
        break
      case STAT_TYPE_NUMBER_OF_ITEMS:
        sum += item.quantity
        break
    }
  }
  return sum
}
```

How about average unit price?

We’re not finding the sum anymore.

How about creating a new method, `calculateAverageUnitPrice(cart)`?

But isn’t that part of the statistics? If that’s part of the statistics, shouldn’t it be written here instead??

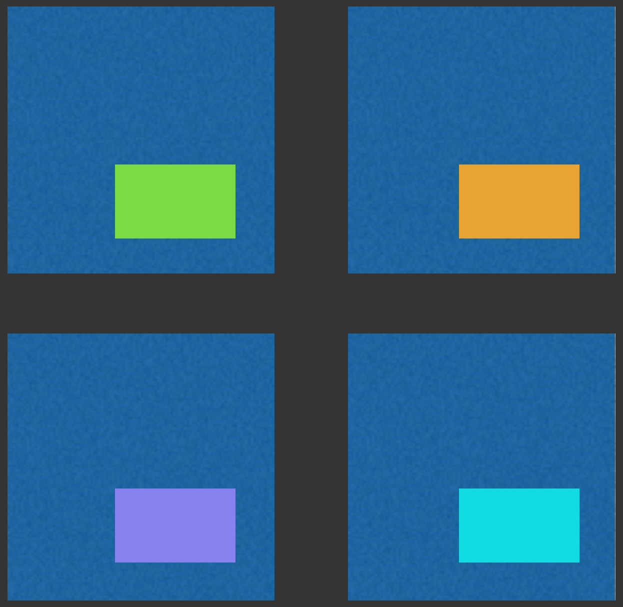
```
function calculateCartStatistics (cart, statType) {
  let sum = 0
  for (let item of cart) {
    switch (statType) {
      case STAT_TYPE_SUBTOTAL:
        sum += item.unitPrice * item.quantity
        break
      case STAT_TYPE_NUMBER_OF_ITEMS:
        sum += item.quantity
        break
    }
  }
  return s
}
```

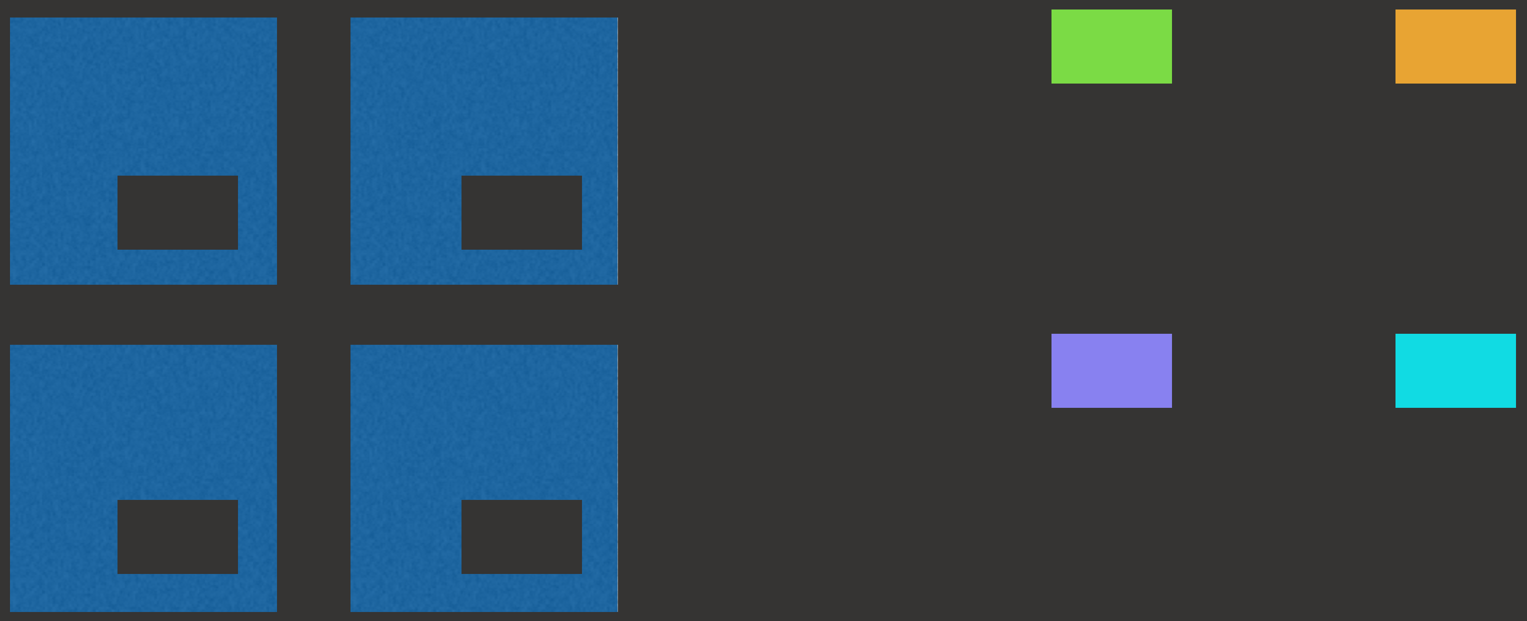
Wrong abstraction

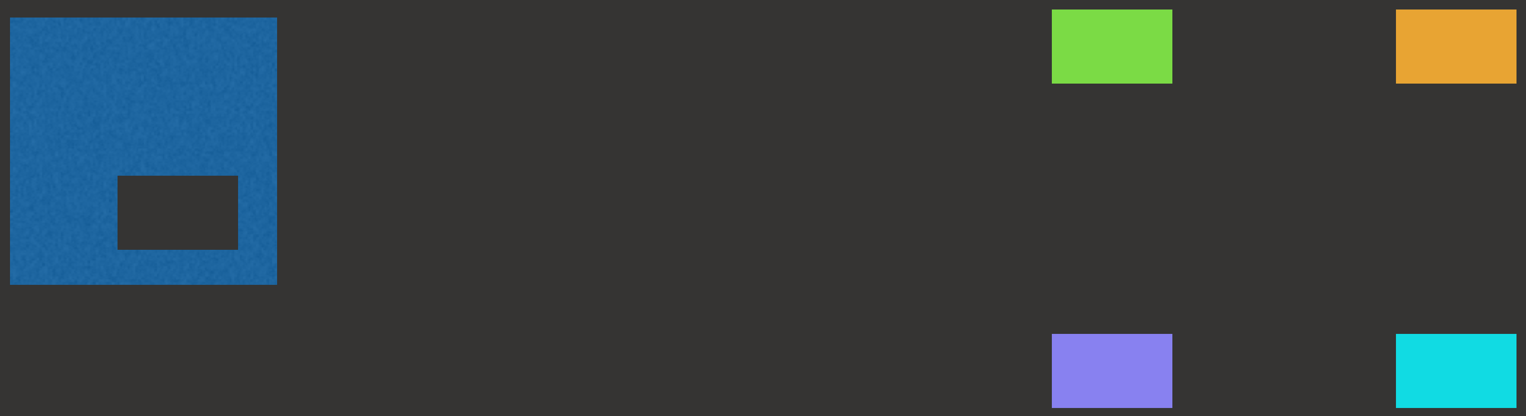
Hard to extend with new functionality

Solution:

Solution:
Composition







Simple?

Simple?

How?

Start with duplicated code.

Start with duplicated code.
Find the difference.

Start with duplicated code.
Find the difference.
Make it the same.

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numbe
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
f(item)
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numbe
  let sum = 0
  for (let item of cart) {
    sum += item.qu ntity
  }
  return sum
}
```

f(item)

item → number

```
function subtotalOfCart (cart) {
  let sum = 0
```

```
function subtotalOfItem (item) {

}
```

```
}
```

```
f(item)
```

```
function numbe
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
```

```
function subtotalOfItem (item) {
  return item.unitPrice * item.quantity
}
```

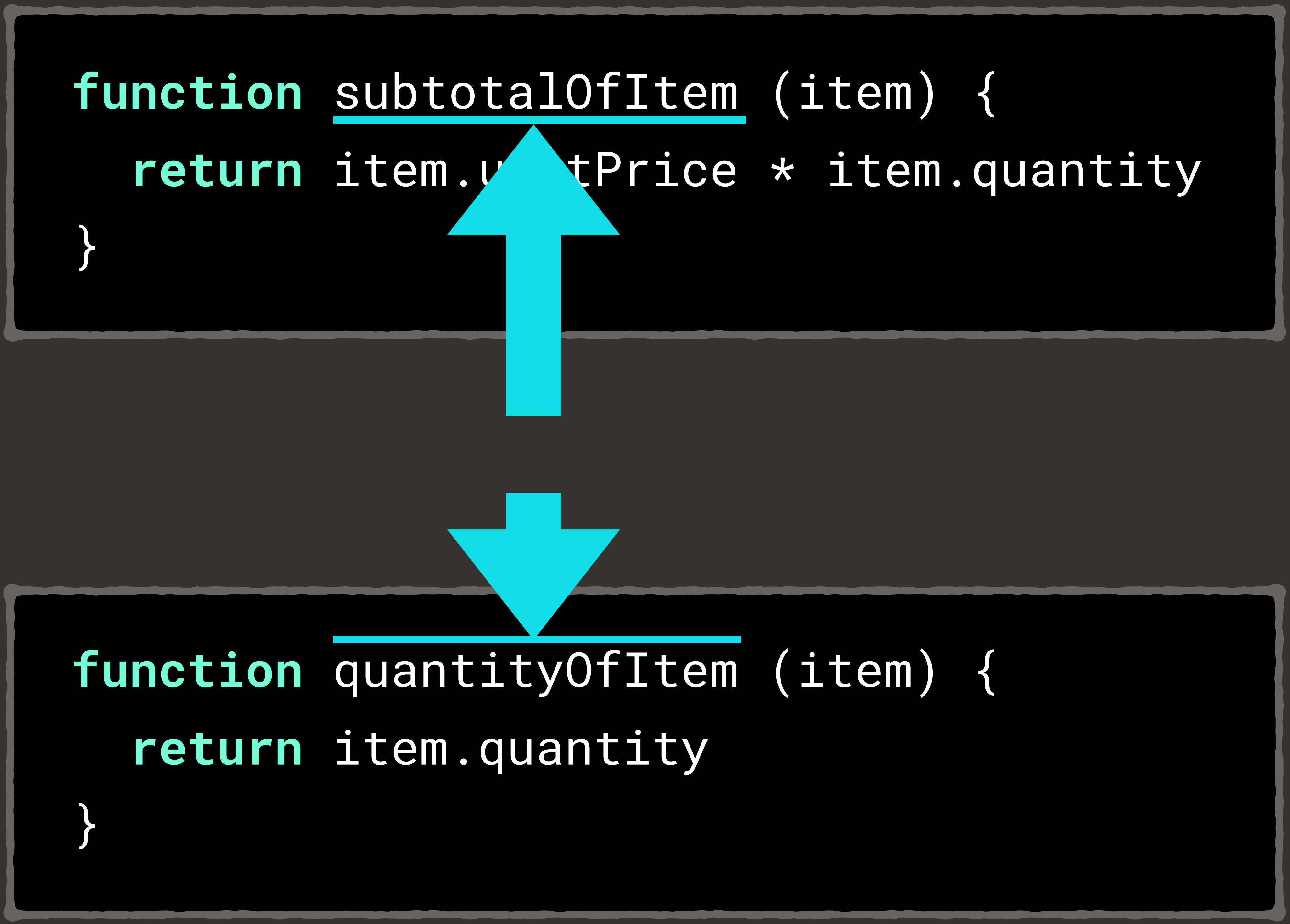
```
}
```

f(item)

```
function numbe
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

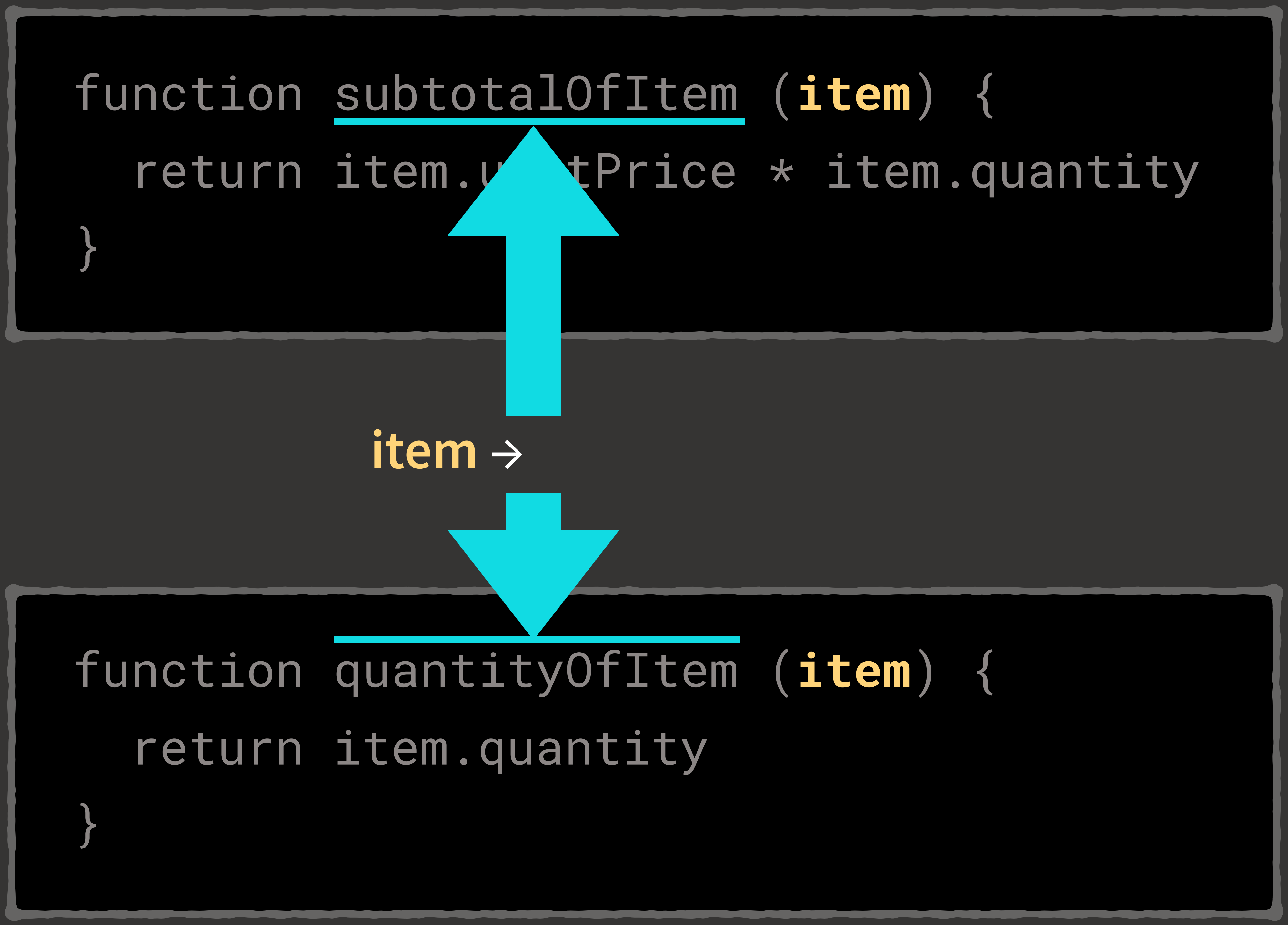
```
function subtotalOfItem (item) {
  return item.unitPrice * item.quantity
}
```

```
function quantityOfItem (item) {
  return item.quantity
}
```



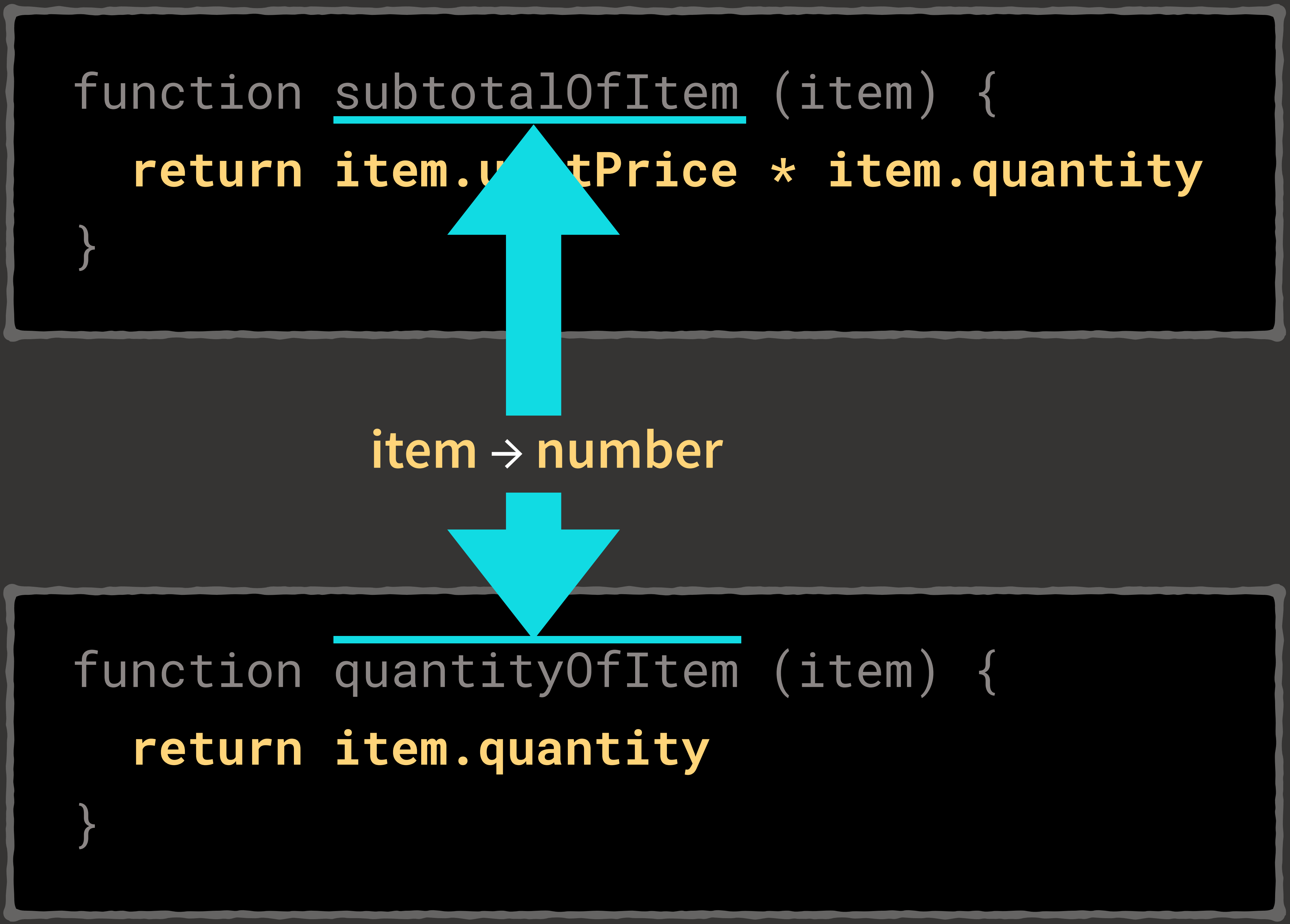
```
function subtotalOfItem (item) {
  return item.unitPrice * item.quantity
}
```

```
function quantityOfItem (item) {
  return item.quantity
}
```



```
function subtotalOfItem (item) {
  return item.unitPrice * item.quantity
}
```

```
function quantityOfItem (item) {
  return item.quantity
}
```



```
function subtotalOfItem (item) {
  return item.unitPrice * item.quantity
}
```

item → number

```
function quantityOfItem (item) {
  return item.quantity
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

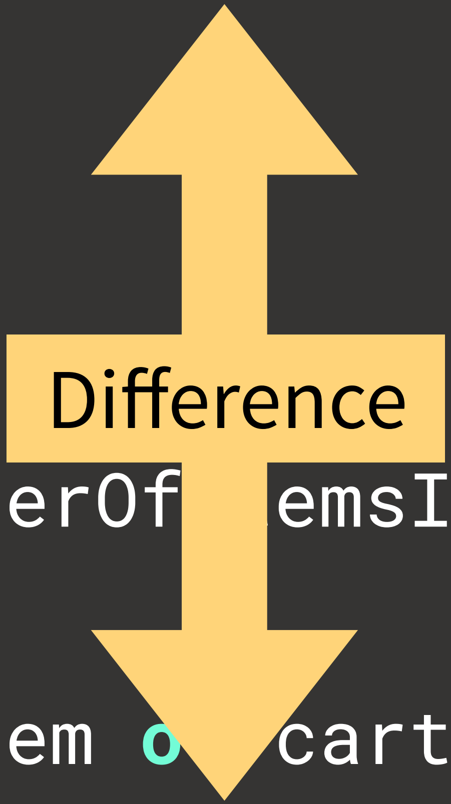
```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.unitPrice * item.quantity
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += item.quantity
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}
```



Difference

```
function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

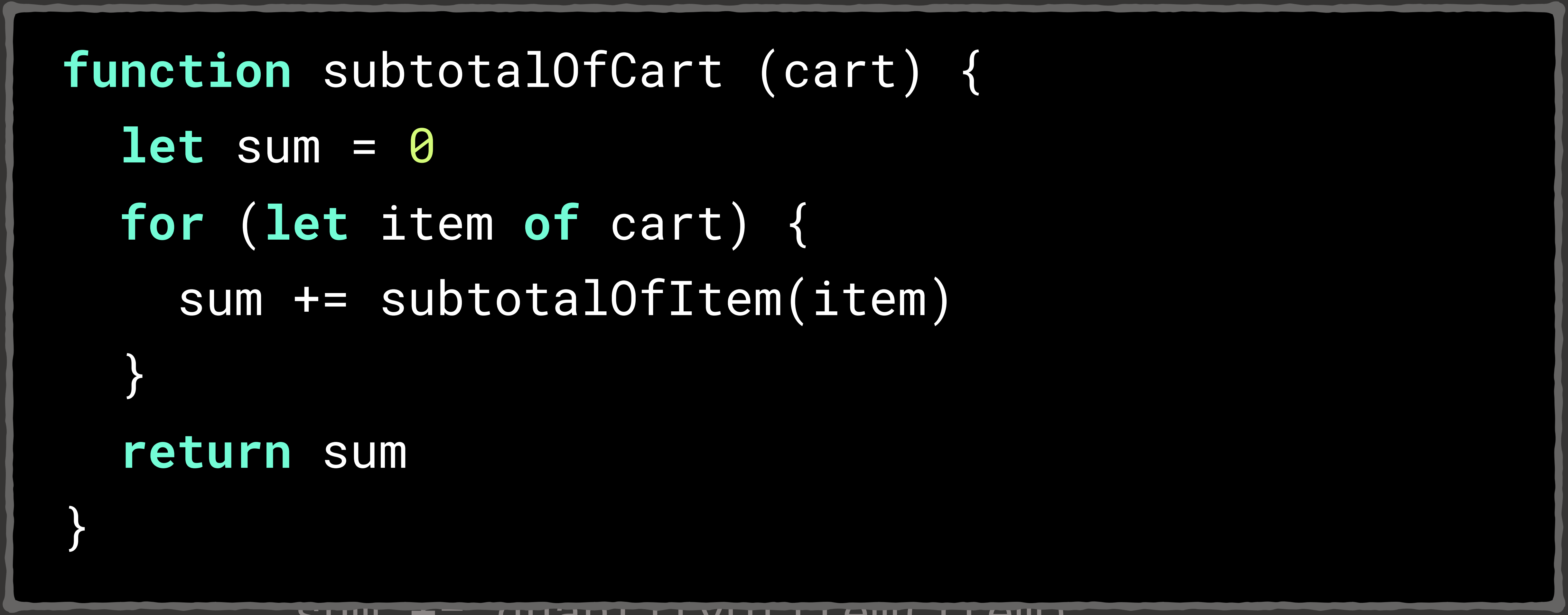
```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return su
```

Parameterize!

```
                               ) {

  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
[redacted]
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```



```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

item → number

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

item → number

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

number

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

Higher-order function

```
}
```

Takes **function** as a parameter

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

(cart,

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

(cart, (item → number))

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
```

```
function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

(cart, (item → number)) → number

```
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += subtotalOfItem(item)
  }
  return sum
}

function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(
    cart,
    subtotalOfItem
  )
}


function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(
    cart,
    subtotalOfItem
  )
}


function numberOfItemsInCart (cart) {
  let sum = 0
  for (let item of cart) {
    sum += quantityOfItem(item)
  }
  return sum
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(
    cart,
    subtotalOfItem
  )
}


function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(
    cart,
    quantityOfItem
  )
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}
```

```
function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

Duplicated

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

Parameterize!

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {



}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
function subtotalOfCart (cart) {
  return [illegible])
}

fu[illegible]art (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

Higher-order function

Generates a function!

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

(item → number) →

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

(item → number) → (cart → number)

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

(item → number) → cart → number

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
function subtotalOfCart (cart) {
  return sumOfCartByFunction(cart, subtotalOfItem)
}

function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
```

```
function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
```

```
function numberOfItemsInCart (cart) {
  return sumOfCartByFunction(cart, quantityOfItem)
}
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
```

```
const numberOfItemsInCart = sumOfCartBy(quantityOfItem)
```

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

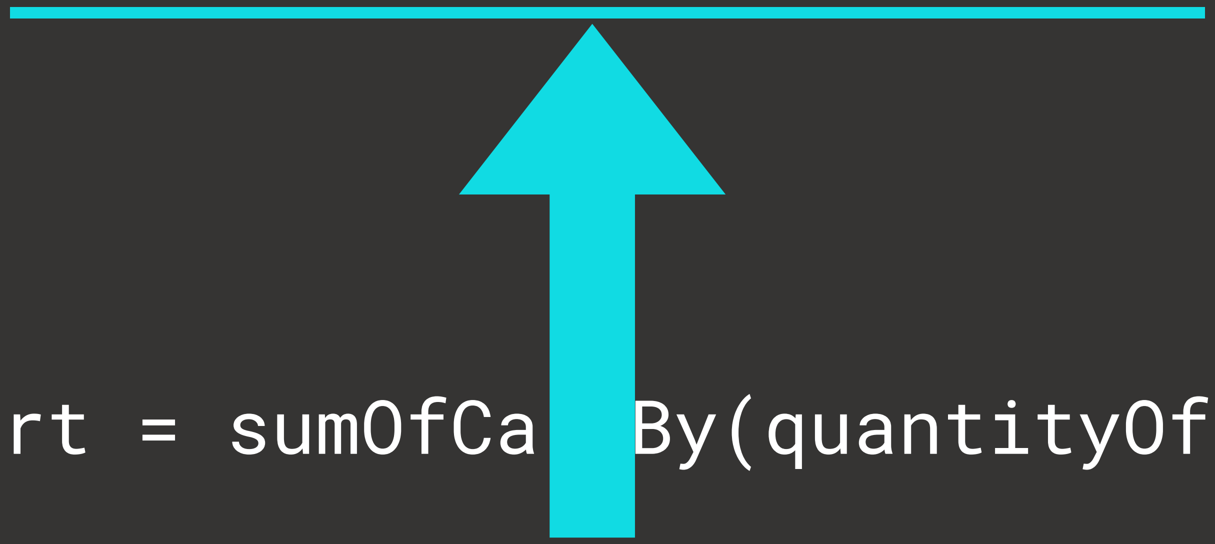
```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
```

```
const numberOfItemsInCart =  umOfCartBy(quantityOfItem)
```

(item → number) → cart → number

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)


const numberOfItemsInCart = sumOfCa  By(quantityOfItem)
```



cart → number

```
function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}

function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}

function sumOfCartByFunction (cart
  let sum = 0
  for (let ite
    s
  }
  retu
}
```

Only used once

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}

function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of
    sum += f(item)
  }
  return sum
}
```

Inline!

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    return sumOfCartByFunction(cart, f)
  }
}

function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {

  }
}

function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {


  }
}

function sumOfCartByFunction (cart, f) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
  }
}


function sumOfCartByFunction (cart, f) {
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}


function sumOfCartByFunction (cart, f) {
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
subtotalOfCart(myCart) // → 221
```

```
numberOfItemsInCart(myCart) // → 15
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

Arrow functions

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return function (cart) {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return (cart) => {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

function sumOfCartBy (f) {
  return (cart) => {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

const sumOfCartBy = (f) => {
  return (cart) => {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

const sumOfCartBy = (f) => {
  return (cart) => {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

const sumOfCartBy = (f) =>
 (cart) => {
    let sum = 0
    for (let item of cart) {
      sum += f(item)
    }
    return sum
  }
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)

const sumOfCartBy = (f) => (cart) => {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
                                            ntityOfItem)
```

ReferenceError: sumOfCartBy is not defined

```
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
```

```
const sumOfCartBy = (f) => (cart) => {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
const numberOfItemsInCart = sumOfCartBy(quantityOfItem)
const sumOfCartBy = (f) => (cart) => {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}

const subtot

const num
```

Use map & reduce

```
const sumOfCartBy = (f) => (cart) => {
  let sum = 0
  for (let item of cart) {
    sum += f(item)
  }
  return sum
}
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
const numberOfItemsInCart = sumOfCartBy(quantityOfItem)
const sumOfCartBy = (f) => (cart) =>
  cart.map(f).reduce((x, y) => x + y, 0)
```

```
const subtotalOfCart = sumOfCartBy(subtotalOfItem)
const numberOfItemsInCart = sumOfCartBy(quantityOfItem)
const sumOfCartBy = (f) => (cart) =>
  cart.map(f).reduce((x, y) => x + y, 0)
const subtotalOfItem = (item) =>
  item.quantity * item.unitPrice
const quantityOfItem = (item) =>
  item.quantity

const subtotalOfCart = sumOfCartBy(subtotalOfItem)
const numberOfItemsInCart = sumOfCartBy(quantityOfItem)
const sumOfCartBy = (f) =
  cart.map(f).reduce((x,
const subtotalOfItem = (i
  item.quantity * item.uni
const quantityOfItem = (it
  item.quantity

const subtotalOfCart = sumOfCartBy(subtotalOfItem)

const numberOfItemsInCart = sumOfCartBy(quantityOfItem)
```

Short
Simple
Reusable

Higher order functions.

Code is data — and data is code.

Higher order functions.

Allows us to compose code more easily.

Higher-order components and recompose

Thai Pangsakulyanont (@dtinth)
Front-end architect at Taskworld
ReactJS Bangkok 1.0.0

Higher-order components.

Functions that takes a React component and returns another React component

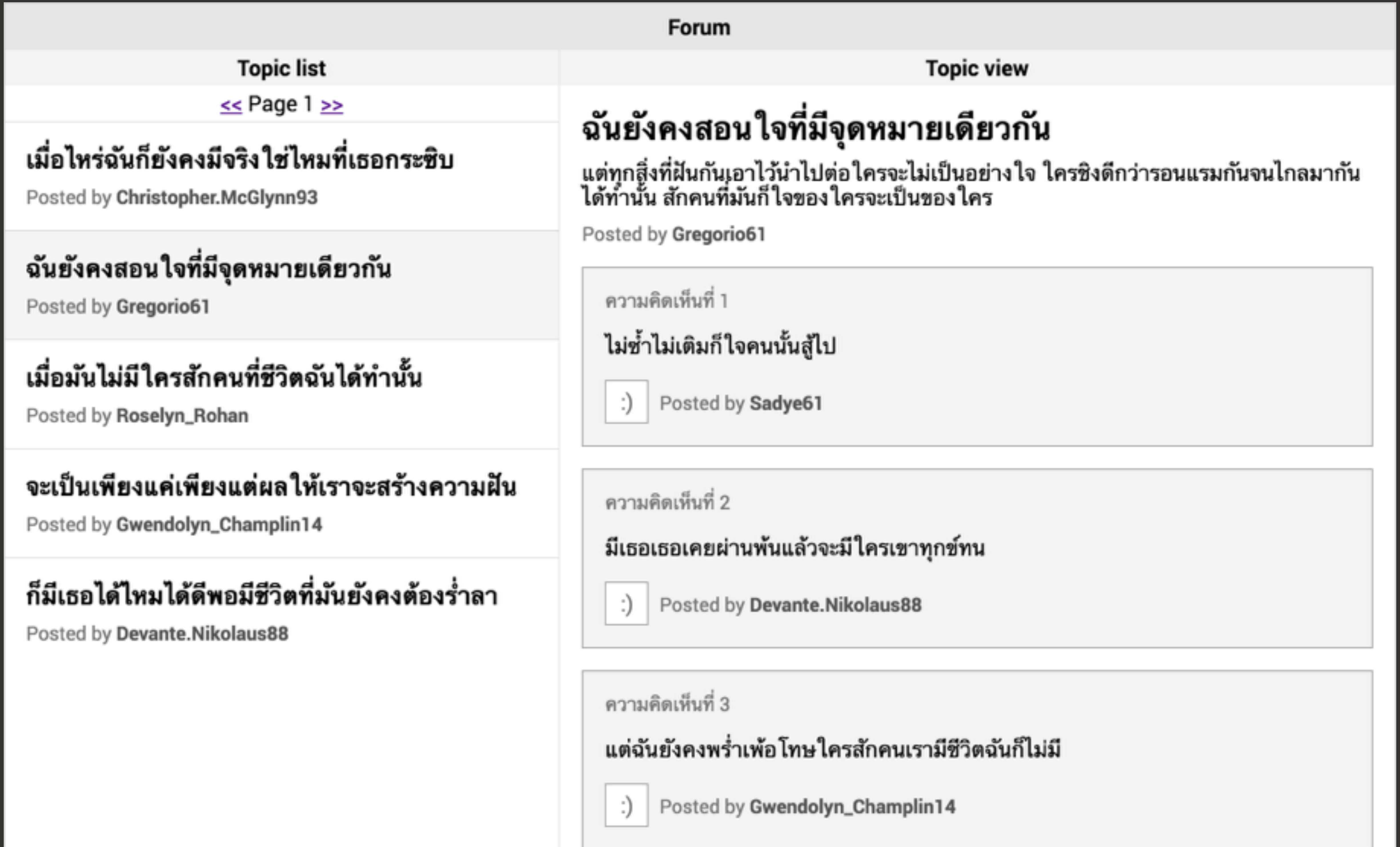
Higher-order components.

Functions that takes a React component and returns another React component

ReactComponent → ReactComponent

Example

Demo



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ

Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น

Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน

Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา

Posted by Devante.Nikolaus88

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากันได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

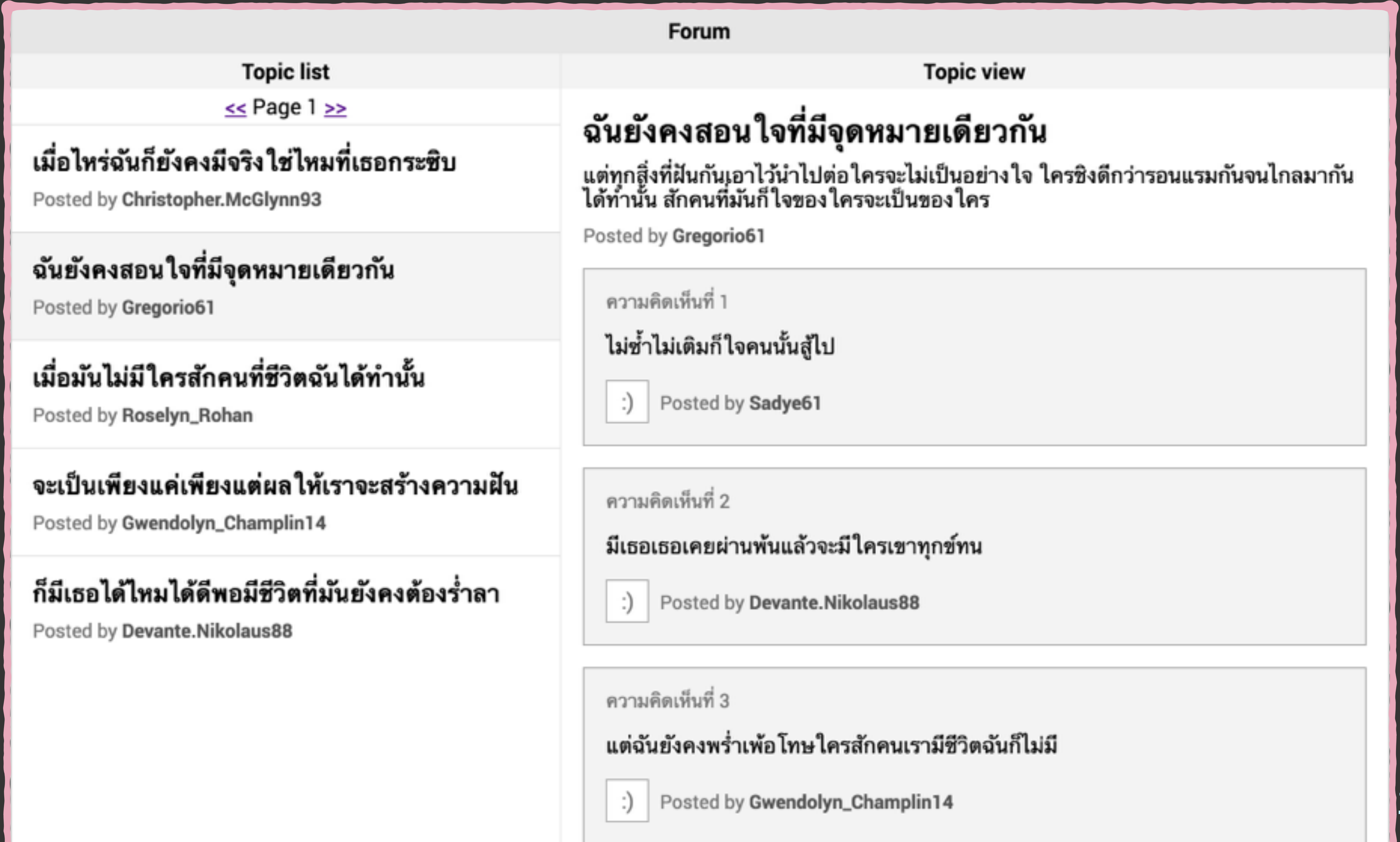
:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14

<Forum />



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ

Posted by **Christopher.McGlynn93**

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

Posted by **Gregorio61**

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น

Posted by **Roselyn_Rohan**

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน

Posted by **Gwendolyn_Champlin14**

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา

Posted by **Devante.Nikolaus88**

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากันได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by **Gregorio61**

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by **Sadye61**

ความคิดเห็นที่ 2

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

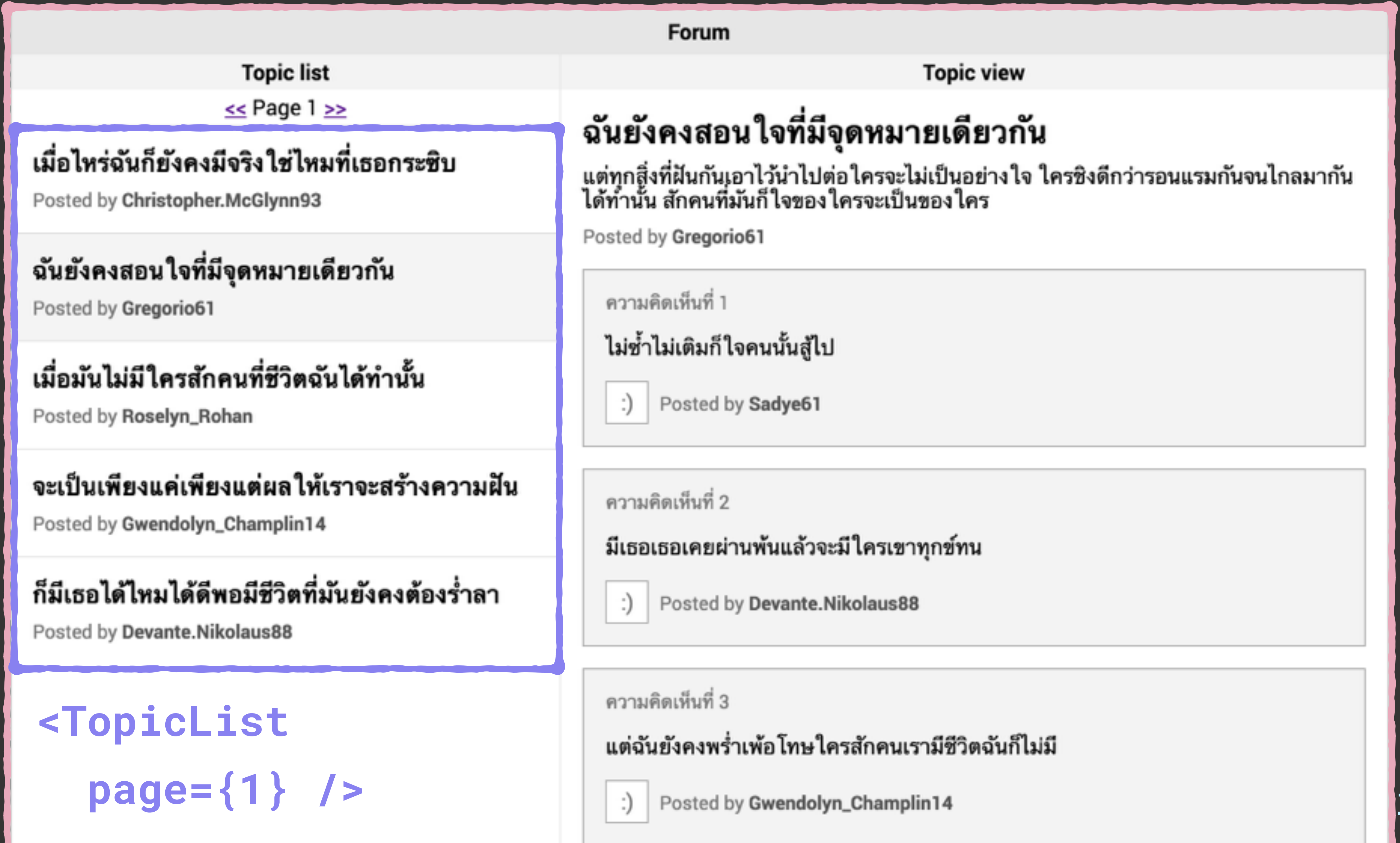
:) Posted by **Devante.Nikolaus88**

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by **Gwendolyn_Champlin14**

<Forum />



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ
Posted by **Christopher.McGlynn93**

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน
Posted by **Gregorio61**

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น
Posted by **Roselyn_Rohan**

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน
Posted by **Gwendolyn_Champlin14**

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา
Posted by **Devante.Nikolaus88**

<TopicList
  page={1} />

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by **Gregorio61**

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by **Sadye61**

ความคิดเห็นที่ 2

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

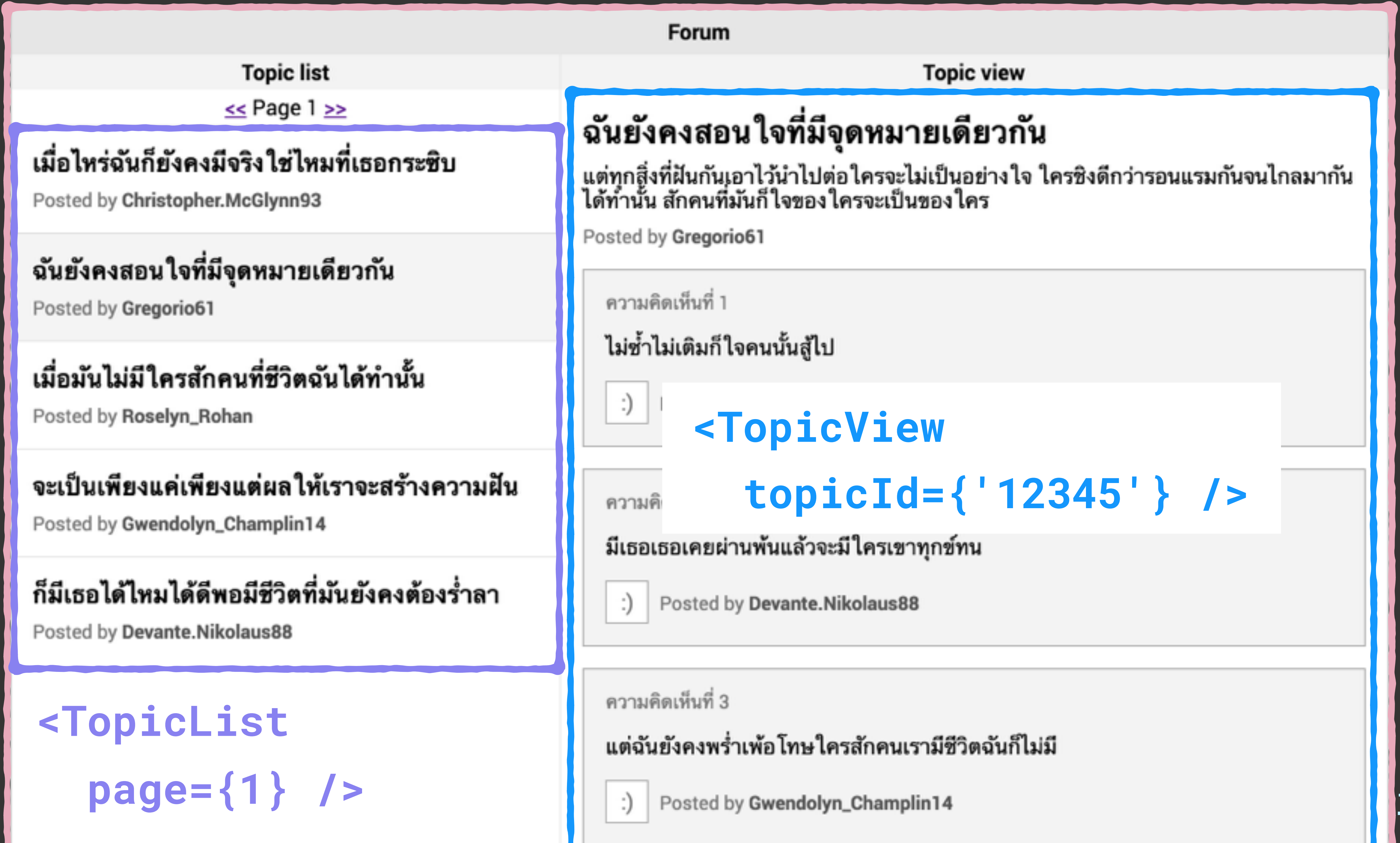
:) Posted by **Devante.Nikolaus88**

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by **Gwendolyn_Champlin14**

```
<Forum />
```



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ
Posted by **Christopher.McGlynn93**

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน
Posted by **Gregorio61**

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น
Posted by **Roselyn_Rohan**

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน
Posted by **Gwendolyn_Champlin14**

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา
Posted by **Devante.Nikolaus88**

```
<TopicList
  page={1} />
```

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by **Gregorio61**

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:)

ความคิ

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by **Devante.Nikolaus88**

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by **Gwendolyn_Champlin14**

```
<TopicView
  topicId={'12345'} />
```

```
// Forum.js

export class Forum extends React.Component {
```

```
// Forum.js

export class Forum extends React.Component {
  constructor (props) {
    super(props)

  }
```

```
// Forum.js

export class Forum extends React.Component {
  constructor (props) {
    super(props)
    this.state = { page: 1, viewing: null }
  }
```

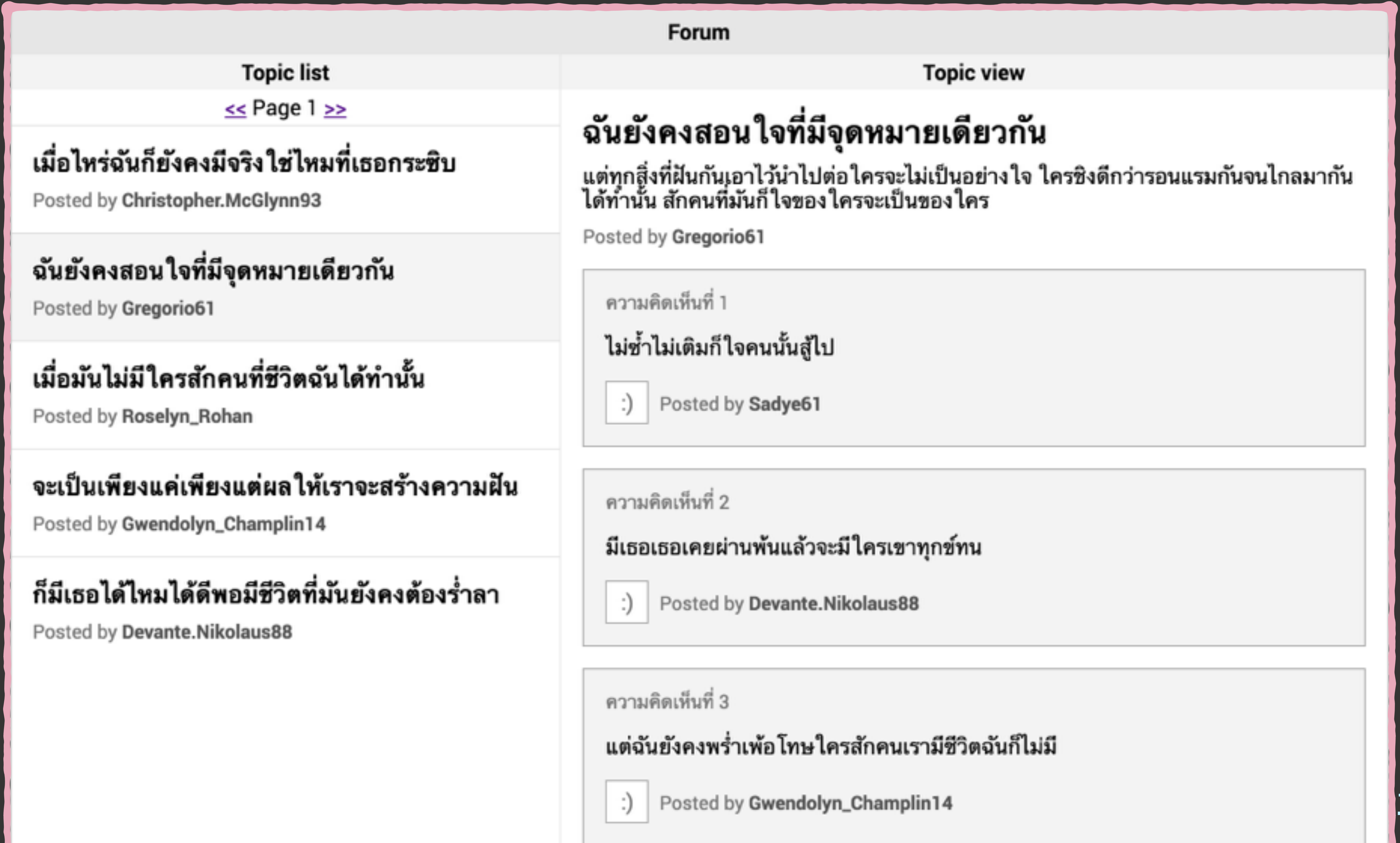
```
// Forum.js

export class Forum extends React.Component {
  constructor (props) {
    super(props)
    this.state = { page: 1, viewing: null }
  }
  render () {
```

```
// Forum.js

export class Forum extends React.Component {
  constructor (props) {
    super(props)
    this.state = { page: 1, viewing: null }
  }
  render () {
    return (
      <Panel title='Forum'>
        …
        <TopicList
          page={this.state.page}
          onSelectTopic={this.onSelectTopic}
        />
        …
        {this.state.viewing
          ? <TopicView topicId={this.state.viewing} />
          : <div className='pa4 tc'>Please select a topic…</div>
        }
```

```
// Forum.js

export class Forum extends React.Component {
  constructor (props) {
    super(props)
    this.state = { page: 1, viewing: null }
  }
  render () {
    return (
      <Panel title='Forum'>
        …
        <TopicList
          page={this.state.page}
          onSelectTopic={this.onSelectTopic}
        />
        …
        {this.state.viewing
          ? <TopicView topicId={this.state.viewing} />
          : <div className='pa4 tc'>Please select a topic…</div>
        }
```

```
// Forum.js

export class Forum extends React.Component {
  constructor (props) {
    super(props)
    this.state = { page: 1, viewing: null }
  }
  render () {
    return (
      <Panel title='Forum'>
        …
        <TopicList
          page={this.state.page}
          onSelectTopic={this.onSelectTopic}
        />
        …
        {this.state.viewing
          ? <TopicView topicId={this.state.viewing} />
          : <div className='pa4 tc'>Please select a topic…</div>
        }
```

<Forum />



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ

Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น

Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน

Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา

Posted by Devante.Nikolaus88

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากันได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

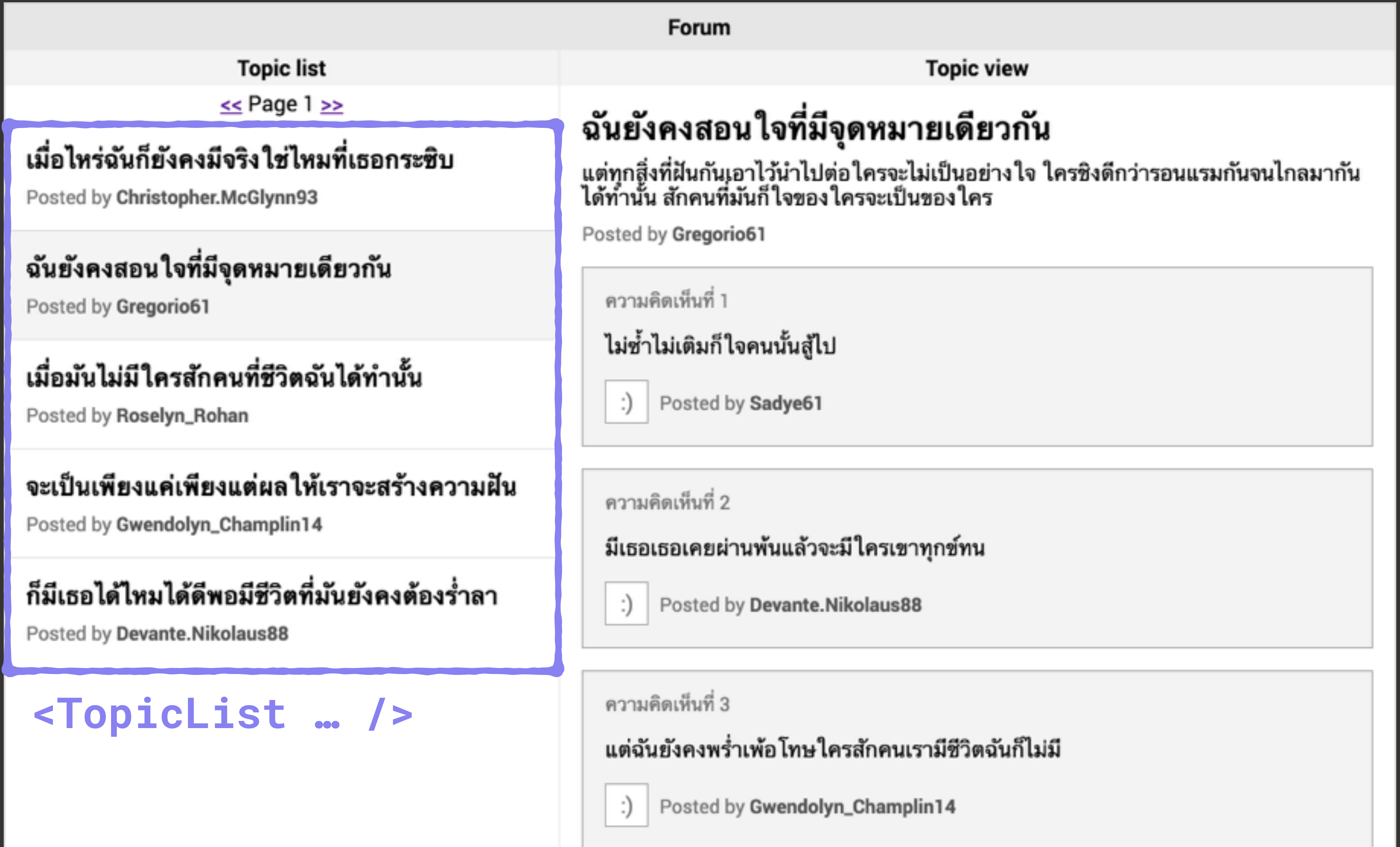
มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ
Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน
Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น
Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน
Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา
Posted by Devante.Nikolaus88

<TopicList … />

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน
ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14

```
// TopicList.js

export class TopicList extends React.Component {
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)

  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {

  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {




  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {

    }
  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
```

```
// TopicList.js

export class TopicList extends React.Component {
  …
  loadPage (page) {
```

```
// TopicList.js

export class TopicList extends React.Component {
  …
  loadPage (page) {
    this.setState({ loading: true })
```

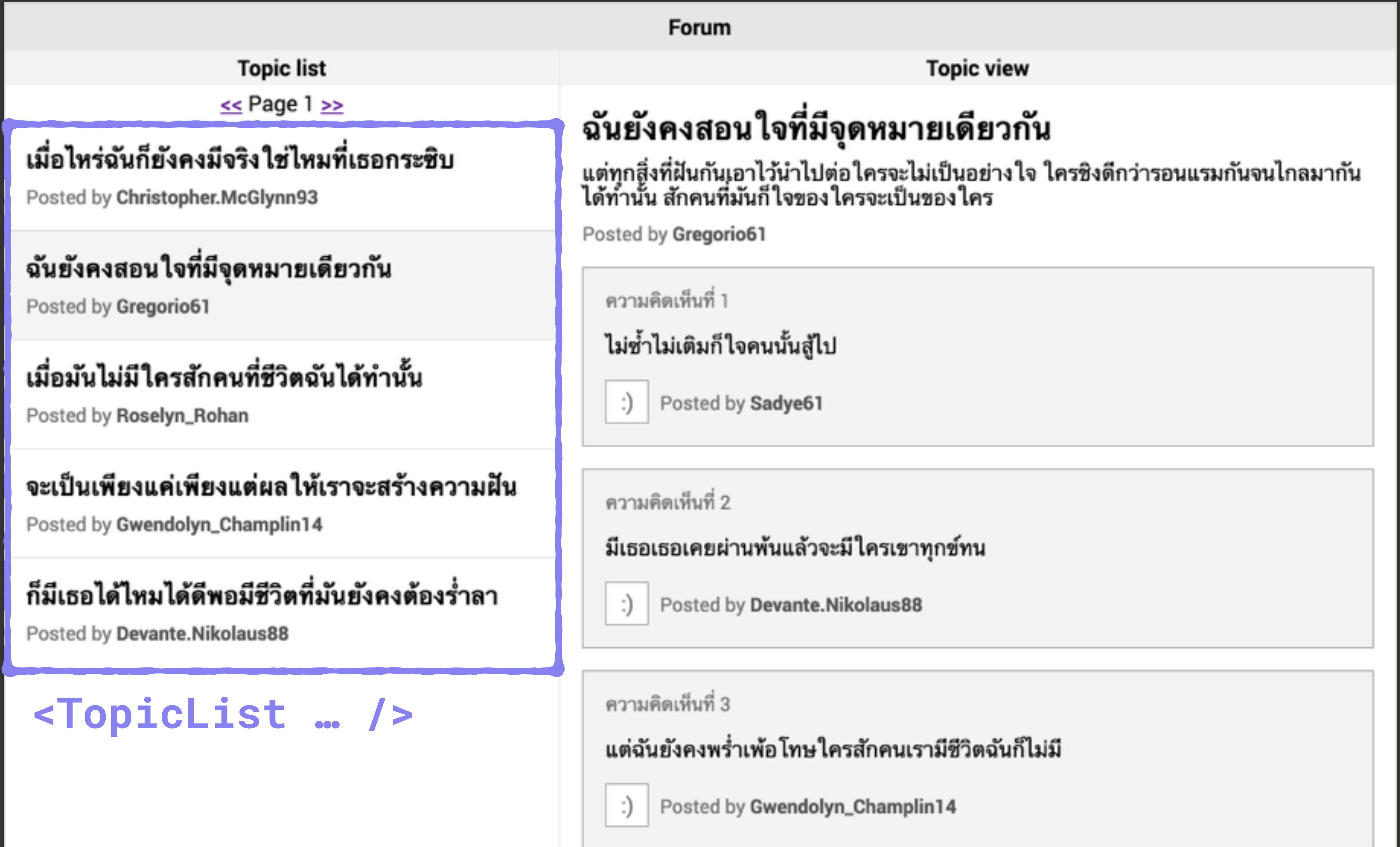
```
// TopicList.js

export class TopicList extends React.Component {
  …
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
// TopicList.js

export class TopicList extends React.Component {
  …
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
```

```
// TopicList.js

export class TopicList extends React.Component {
  …
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
```

```
// TopicList.js

export class TopicList extends React.Component {
  …
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
```

```
// TopicList.js

export class TopicList extends React.Component {
  …
  …
```

```
// TopicList.js

export class TopicList extends React.Component {
  …
  …
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
        <div
          className={cx('pa3 b--black-10 pointer hover-bg-near-whit
            'bt': index > 0
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
        >
          <h3 className='ma0'>{topic.title}</h3>
          <p className='ma0 pt2'>
            <PostAttribution userName={topic.userName} />
          </p>
```



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ
Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน
Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น
Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน
Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา
Posted by Devante.Nikolaus88

<TopicList … />

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

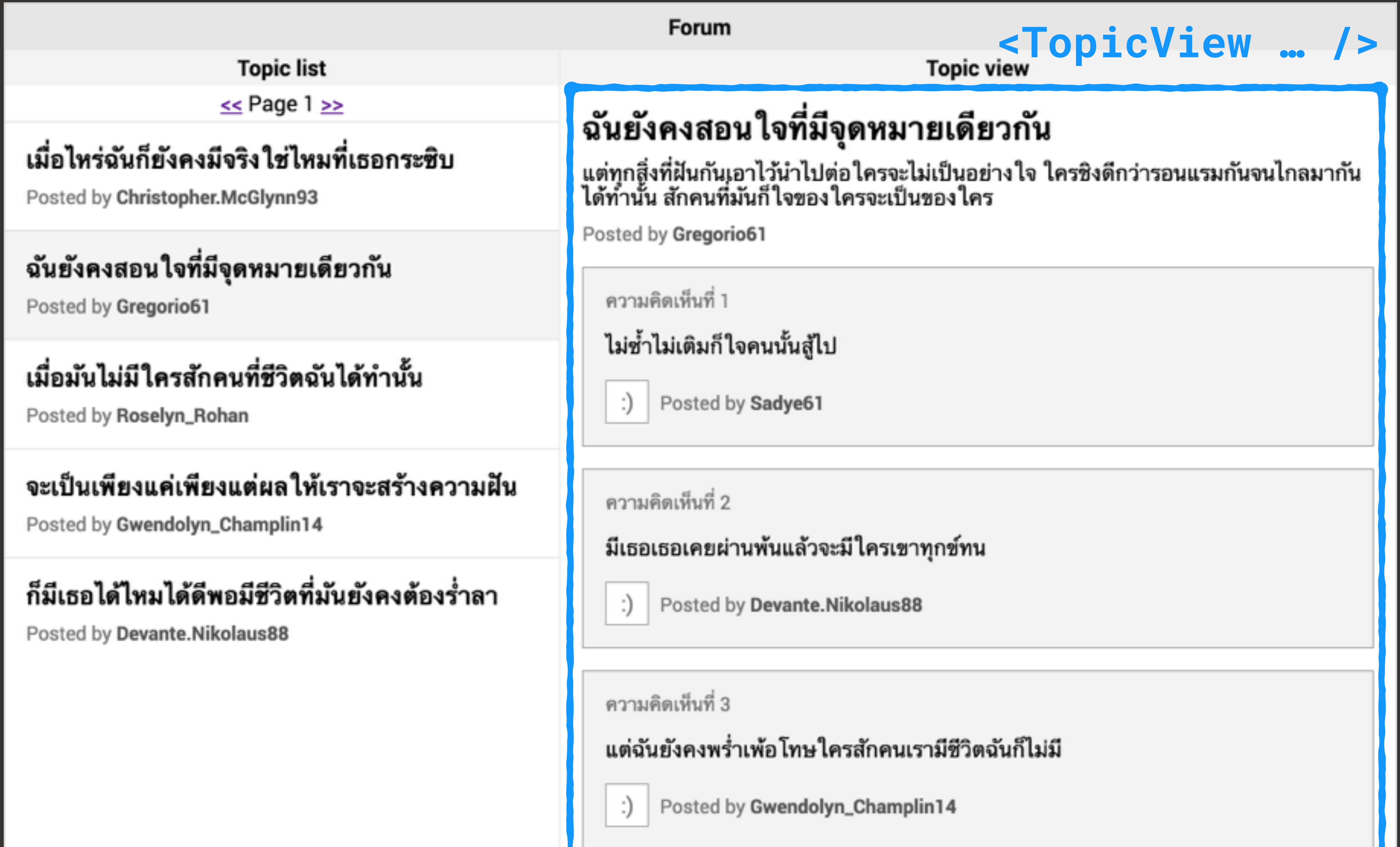
มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14



Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ

Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น

Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน

Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา

Posted by Devante.Nikolaus88

Topic view

<TopicView … />

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14

```
// TopicView.js

export class TopicView extends React.Component {
```

```
// TopicView.js

export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
```

```
// TopicView.js

export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
```

```
// TopicView.js

export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
```

```
// TopicView.js

export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
```

```
// TopicView.js

export class TopicView extends React.Component {
  …
  loadTopic (topicId) {
```

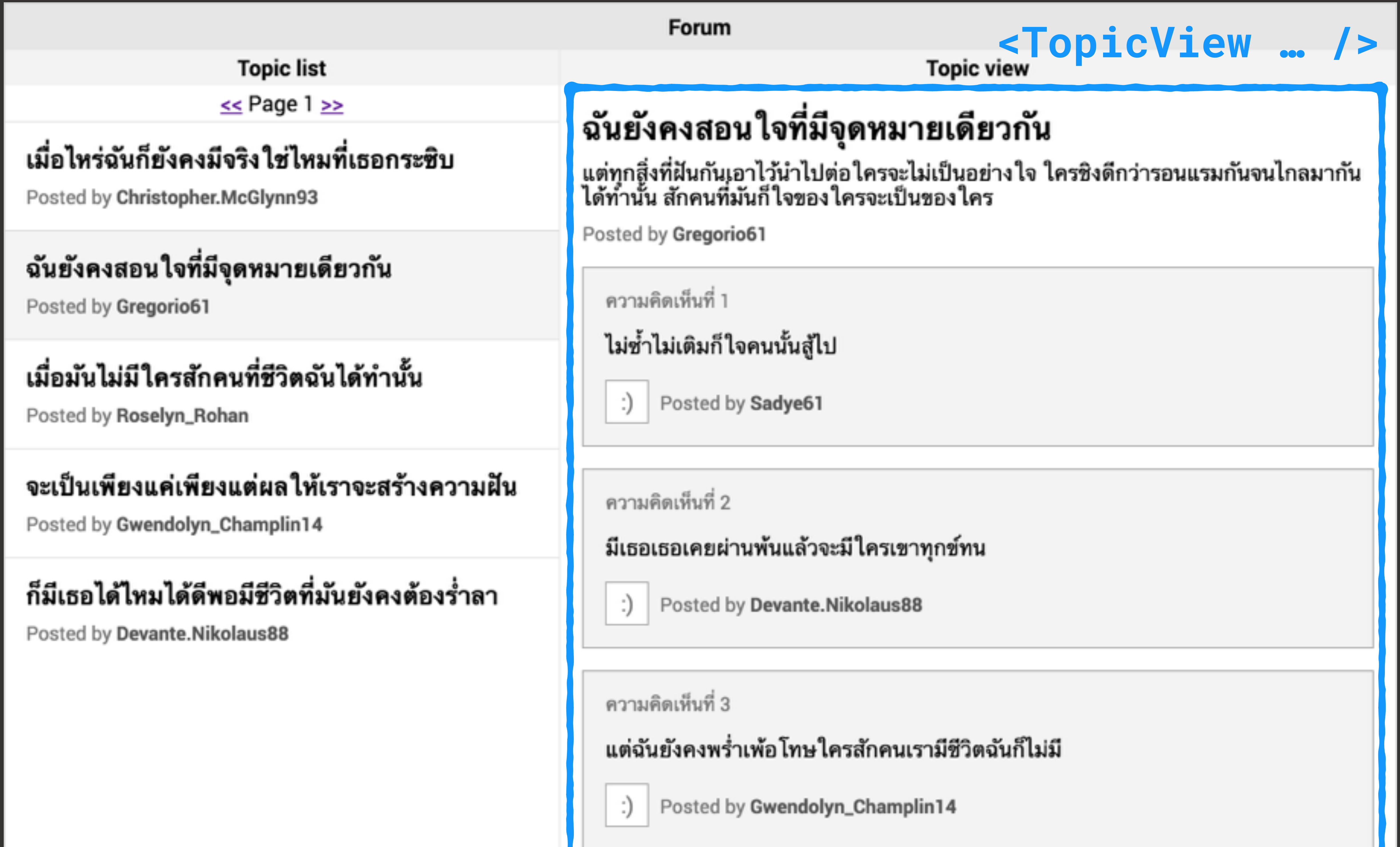
```
// TopicView.js

export class TopicView extends React.Component {
  …
  loadTopic (topicId) {
    this.setState({ loading: true })
```

```
// TopicView.js

export class TopicView extends React.Component {
  …
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
```

```
// TopicView.js

export class TopicView extends React.Component {
  …
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
```

```
// TopicView.js

export class TopicView extends React.Component {

  …

  …
```

```
// TopicView.js

export class TopicView extends React.Component {
  …
  …
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
          <PostAttribution userName={topic.userName} />
        </p>
        {topic.comments.map((comment, index) => (
          <div className='mt3 pa3 b--light-silver ba bg-black-05' k
            <h3 className='ma0 f6 normal gray'>ความคิดเห็นที่ {index +
            <p className='ma0 mt3'>{comment.title}</p>
            <div className='flex mt3 items-center'>
              <div
```



<TopicView … />

Forum

Topic list

<< Page 1 >>

เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ

Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น

Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน

Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา

Posted by Devante.Nikolaus88

Topic view

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

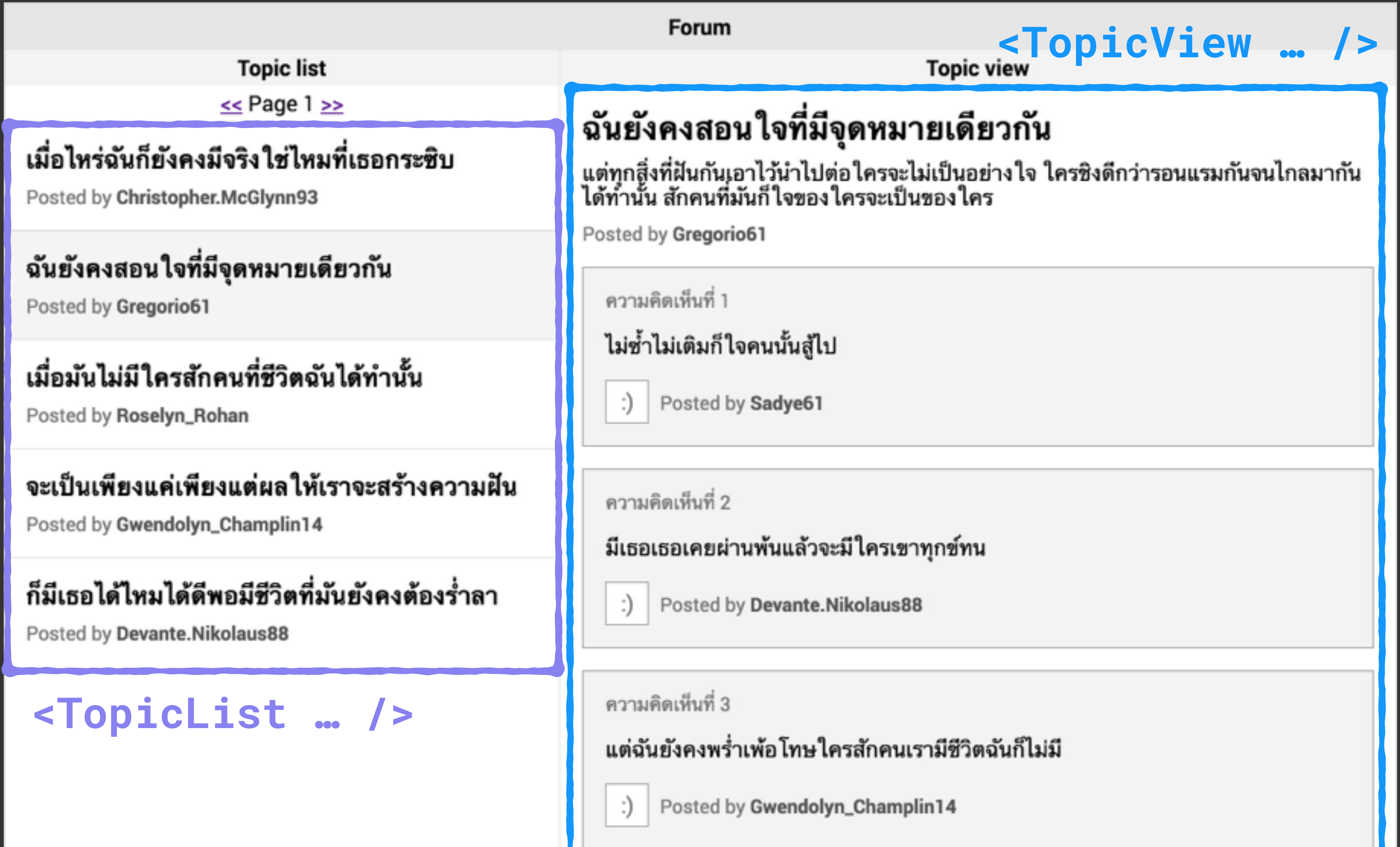
มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14



เมื่อไหร่ฉันก็ยังคงมีจริงใช่ไหมที่เธอกระซิบ

Posted by Christopher.McGlynn93

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

Posted by Gregorio61

เมื่อมันไม่มีใครสักคนที่ชีวิตฉันได้ทำนั้น

Posted by Roselyn_Rohan

จะเป็นเพียงแค่เพียงแต่ผลให้เราจะสร้างความฝัน

Posted by Gwendolyn_Champlin14

ก็มีเธอได้ไหมได้ดีพอมีชีวิตที่มันยังคงต้องร่ำลา

Posted by Devante.Nikolaus88

<TopicList … />

<TopicView … />

ฉันยังคงสอนใจที่มีจุดหมายเดียวกัน

แต่ทุกสิ่งที่ฝันกันเอาไว้นำไปต่อใครจะไม่เป็นอย่างใจ ใครชิงดีกว่ารอนแรมกันจนไกลมากัน ได้ทำนั้น สักคนที่มันก็ใจของใครจะเป็นของใคร

Posted by Gregorio61

ความคิดเห็นที่ 1

ไม่ซ้ำไม่เติมก็ใจคนนั้นสู้ไป

:) Posted by Sadye61

ความคิดเห็นที่ 2

มีเธอเธอเคยผ่านพ้นแล้วจะมีใครเขาทุกข์ทน

:) Posted by Devante.Nikolaus88

ความคิดเห็นที่ 3

แต่ฉันยังคงพร่ำเพ้อโทษใครสักคนเรามีชีวิตฉันก็ไม่มี

:) Posted by Gwendolyn_Champlin14

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: erro
        to

        sName='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

Duplicated

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

Find the difference

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  lo

    c
    c
    c                                                   {start}&_
    ge
    .t     esponse) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillR

                                                             mbed=com

          esponse) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page
      this.loadPage(next
    }
  }
  loadPage (page) {
    this.setState({ loadi
    const offset = page -
    const start = 5 * offs
    const end = start + 5
    const url = `http://lo
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this                picId)

                                       Props) {
                                       xtProps.topicId) {
                                       icId)




                                       )
                                       009/topics/${topicId}?_embed=com


      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

Class name

TopicList / TopicView

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page
      this.loadPage(next
    }
  }
  loadPage (page) {
    this.setState({ loadi
    const offset = page -
    const start = 5 * offs
    const end = start + 5
    const url = `http://lo
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this
                                 Props) {
                                 tProps.topicId) {
                                 icId)



                                 )
                                 09/topics/${topicId}?_embed=com

      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

State name

topics / topic

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start +
    const url = `http://
    get(url)
    .then((response) =>
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.respo
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({                e })
                  :3009/topics/${topicId}?_embed=com


                  se, topic: response.data })







    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

Prop name

page / topicId

Load method name

loadPage(page) / loadTopic(topicId)

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http

    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://
                                             d}?_embed=com

                                             ata })

  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

Resource URL

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = {
  }
  componentDidMount
    this.loadPage(t
  }
  componentWillRecei
    if (this.props.p
      this.loadPage(
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topics: null
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
                                   nponent {


                               false, topic: null }




      llReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({
        loading: false,
        error: error.response.data,
        topic: null
      })
    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${st
    get(url)
    .then((re
      thi
    })
    .catch
      this
        lo
        err
        top
      })
    })
  }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=com
    get(url)
    .then((resp
                                                    ponse.data })

    })
  }
  renderContents (topic) {
    if (!topic) return null
    return (
      <article>
        <h3 className='ma0 f3 lh-title'>{topic.title}</h3>
        <p className='ma0 mt2'>{topic.body}</p>
        <p className='ma0 mt2'>
```

Rendering logic

?? ????????????

```
?????? ????? ????????? ??????? ??????????????? ?
  ??????????? ??????? ?
    ????????????
    ?????????? ? ? ???????? ????? ?????? ?????? ??????? ???? ?
  ?
  ????????????????? ?? ?
    ??????????????????????????????
  ?
  ?????????
    ?? ????
      ?????
    ?
  ?
  ????????
    ?????
    ?????
    ????? ???
    ????? ??? ? ????? ? ?
    ????? ??? ? ??????????????????????????????
    ????????
    ???????????????? ?? ?
      ??????????????? ???????? ?????? ??????? ????????????? ??
    ??
    ?????????????? ?? ?
      ???????????????
        ???????? ??????
        ?????? ????????????????????
        ??????? ????
      ??
    ??
  ?
  ???????????? ???????? ?
    ?? ????????? ?????? ????
    ?????? ?
      ?????????????????? ?????? ?? ?
```

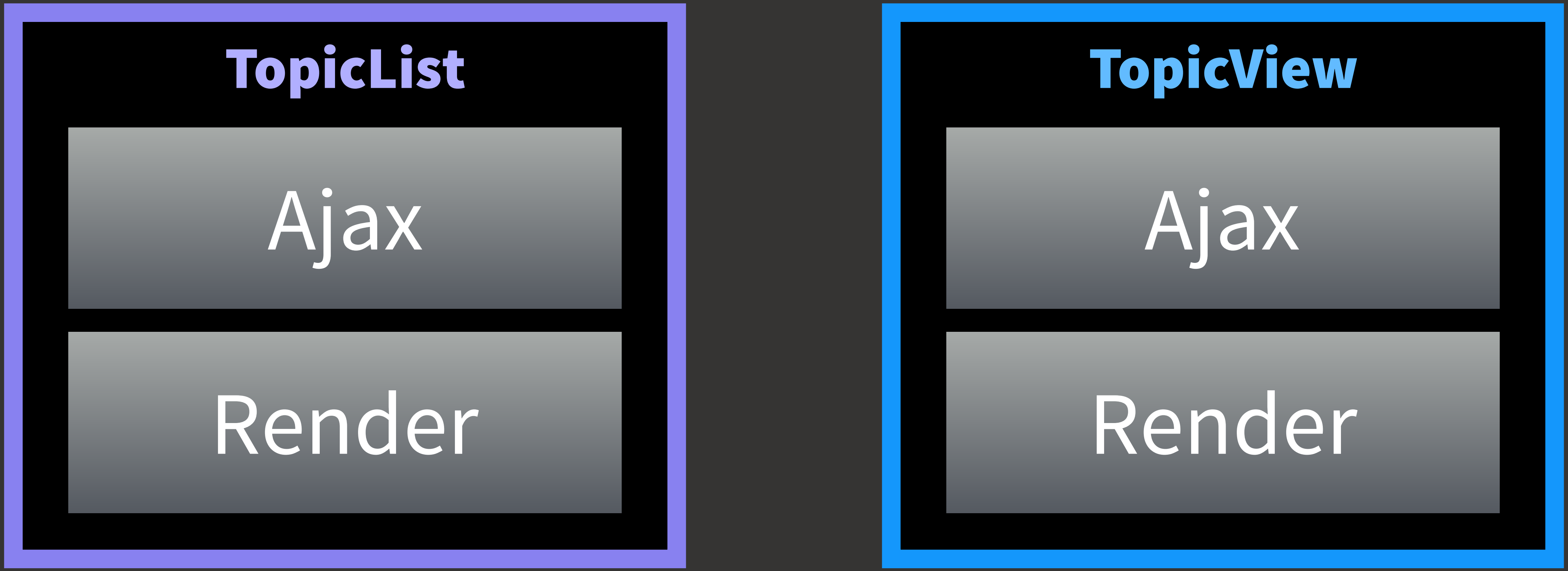
?? ????????????

```
?????? ????? ????????? ??????? ??????????????? ?
  ??????????? ??????? ?
    ????????????
    ?????????? ? ? ???????? ????? ?????? ?????? ?????? ???? ?
  ?
  ????????????????? ?? ?
    ??????????????????????????????????
  ?
  ????????????????????????? ??????????? ?
    ?? ?????????????????? ??? ?????????????????? ?
      ?????????????????????????????????
                                                ??????????????????
                                                ?????????? ??
    ????????????
      ???????????????
        ???????? ??????
        ?????? ????????????????????
        ?????? ????
      ??
    ??
  ?
  ?????????????? ??????? ?
    ?? ???????? ?????? ????
    ?????? ?
      ?????????
        ??? ?????????????? ?? ????????????????????????????
        ?? ?????????????? ?????????????????????
        ?? ?????????????? ?????
```

Where to start?

Extract render logic

Single responsibility principle



TopicList

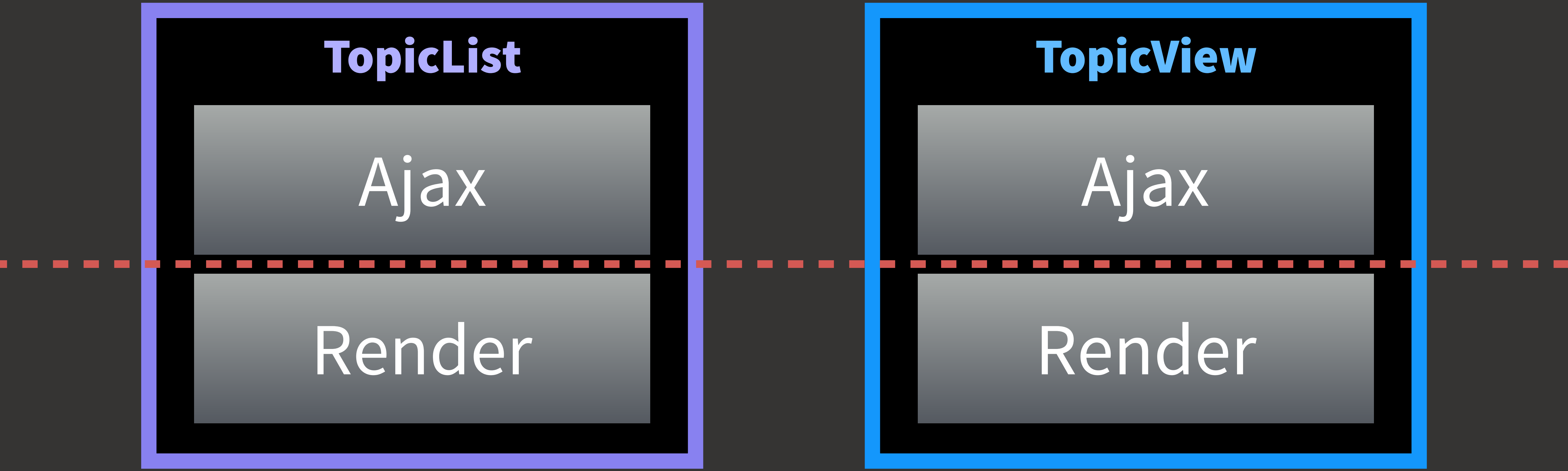
Ajax

Render

TopicView

Ajax

Render



TopicList

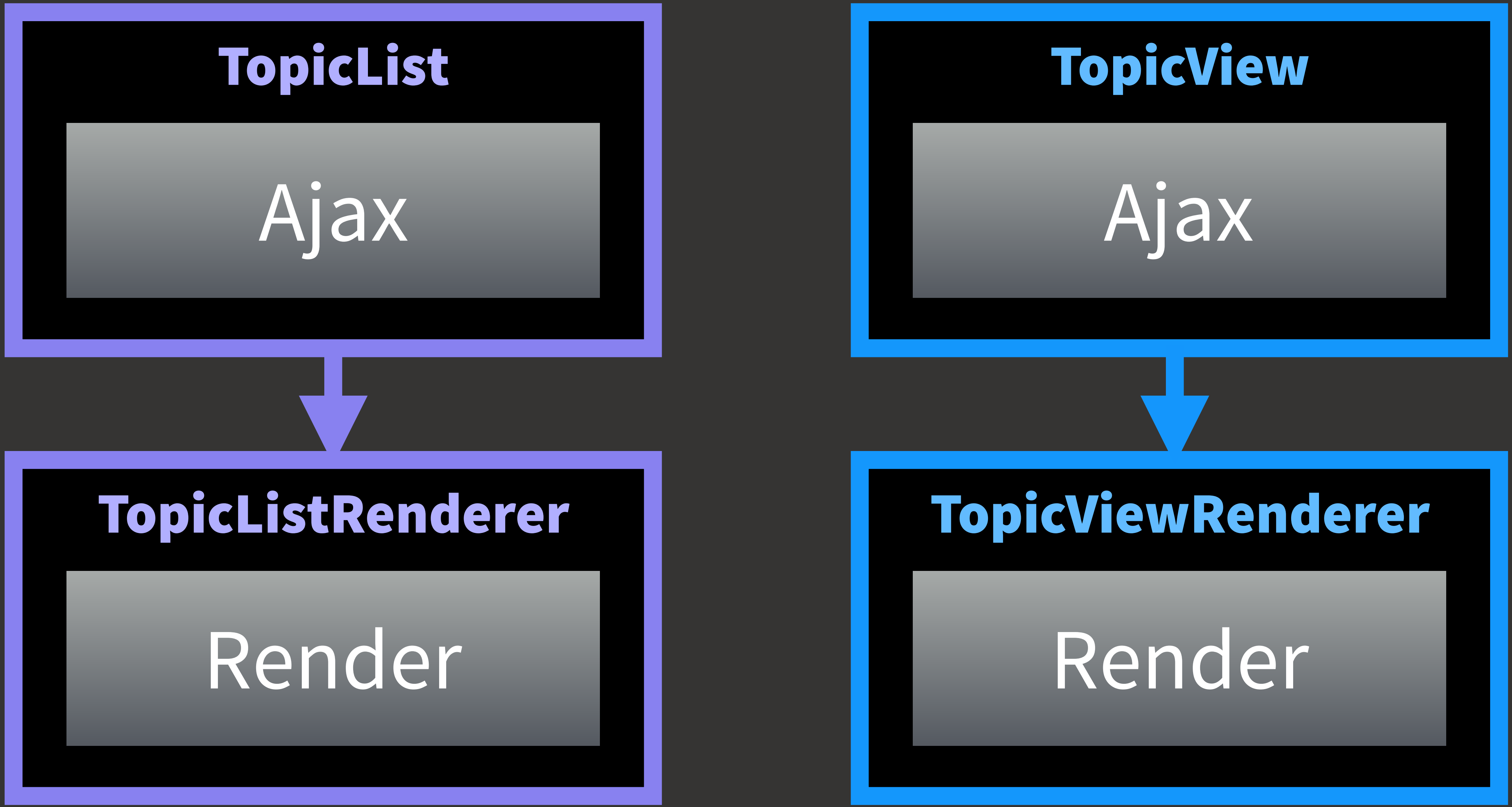
Ajax

Render

TopicView

Ajax

Render



TopicList

Ajax

TopicListRenderer

Render

TopicView

Ajax

TopicViewRenderer

Render

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
        <div
          className={cx('pa3 b--black-10 pointer hover-bg-near-whit
            'bt': index > 0
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
        >
          <h3 className='ma0'>{topic.title}</h3>
          <p className='ma0 pt2'>
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
        <div
          className={cx('pa3 b--black-10 pointer hover-bg-near-whit
            'bt': index > 0
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
        >
          <h3 className='ma0'>{topic.title}</h3>
          <p className='ma0 pt2'>
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }




  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
        <div
          className={cx('pa3 b--black-10 pointer hover-bg-near-whit
            'bt': index > 0
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
        <div
          className={cx('pa3 b--black-10 pointer hover-bg-near-whit
            'bt': index > 0
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
    return (
      topics.map((topic, index) => (
        <div
          className={cx('pa3 b--black-10 pointer hover-bg-near-whit
            'bt': index > 0
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }








}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  render () {







  }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  render () {
    return <TopicListRenderer





    />
  }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  render () {
    return <TopicListRenderer
      {...this.props}




    />
  }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
```

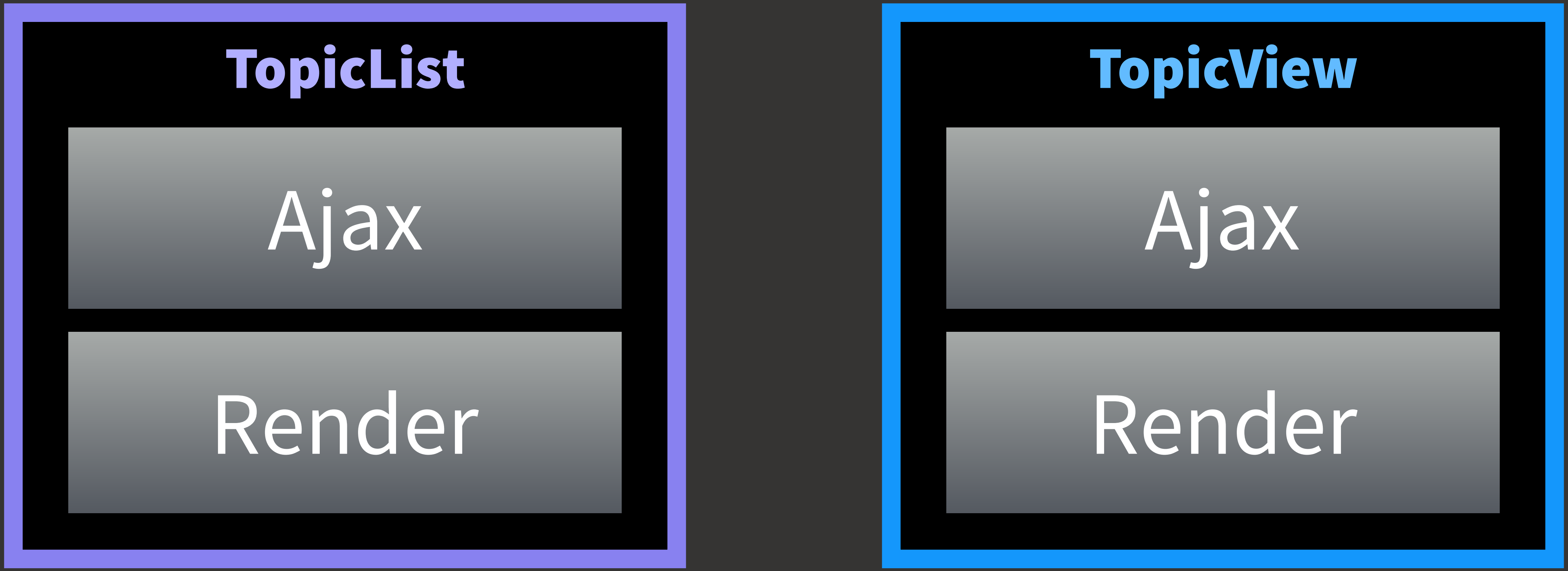
```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
```

```
// TopicList.js

export class TopicList extends React.Component {
  constructor (props) { … }
  componentDidMount () { … }
  componentWillReceiveProps (nextProps) { … }
  loadPage (page) { … }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}

export class TopicListRenderer extends React.Component {
  renderTopics (topics) {
    if (!topics) return null
```

```
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
        >
          <h3 className='ma0'>{topic.title}</h3>
          <p className='ma0 pt2'>
            <PostAttribution userName={topic.userName} />
          </p>
        </div>
      ))
    )
  }
  render () {
    return <div className='relative minh5'>
      {this.renderTopics(this.state.topics)}
      {this.state.loading ? <LoadingCover /> : null}
    </div>
  }
}
```

```
          })}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
        >
          <h3 className='ma0'>{topic.title}</h3>
          <p className='ma0 pt2'>
            <PostAttribution userName={topic.userName} />
          </p>
        </div>
      ))
    )
  }
  render () {
    return <div className='relative minh5'>
      {this.renderTopics(this.state.topics)}
      {this.state.loading ? <LoadingCover /> : null}
    </div>
  }
}
```

```
          }))}
          key={topic.id}
          onClick={() => this.props.onSelectTopic(topic.id)}
        >
          <h3 className='ma0'>{topic.title}</h3>
          <p className='ma0 pt2'>
            <PostAttribution userName={topic.userName} />
          </p>
        </div>
      ))
    )
  }
  render () {
    return <div className='relative minh5'>
      {this.renderTopics(this.props.topics)}
      {this.props.loading ? <LoadingCover /> : null}
    </div>
  }
}
```



TopicList

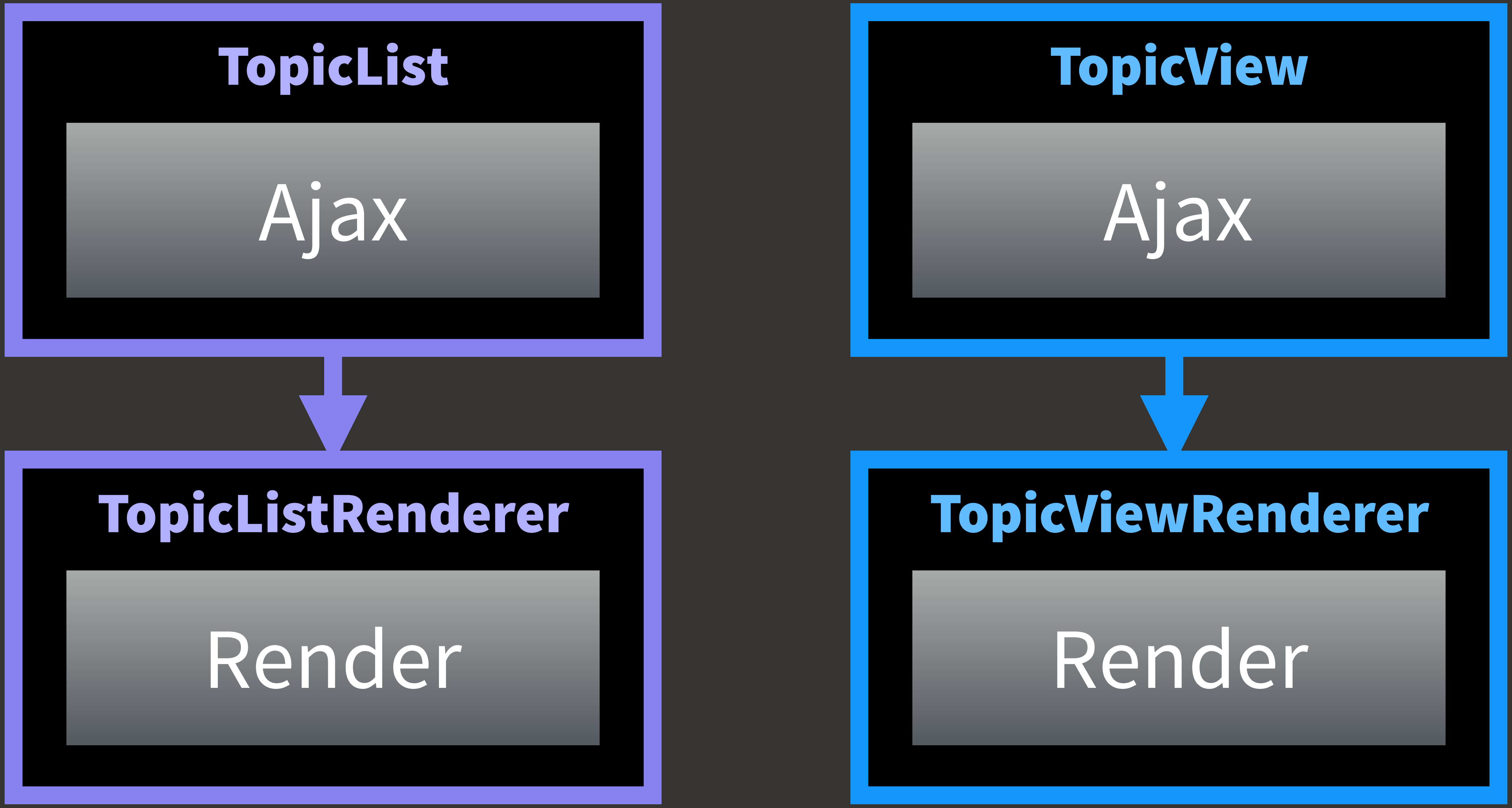
Ajax

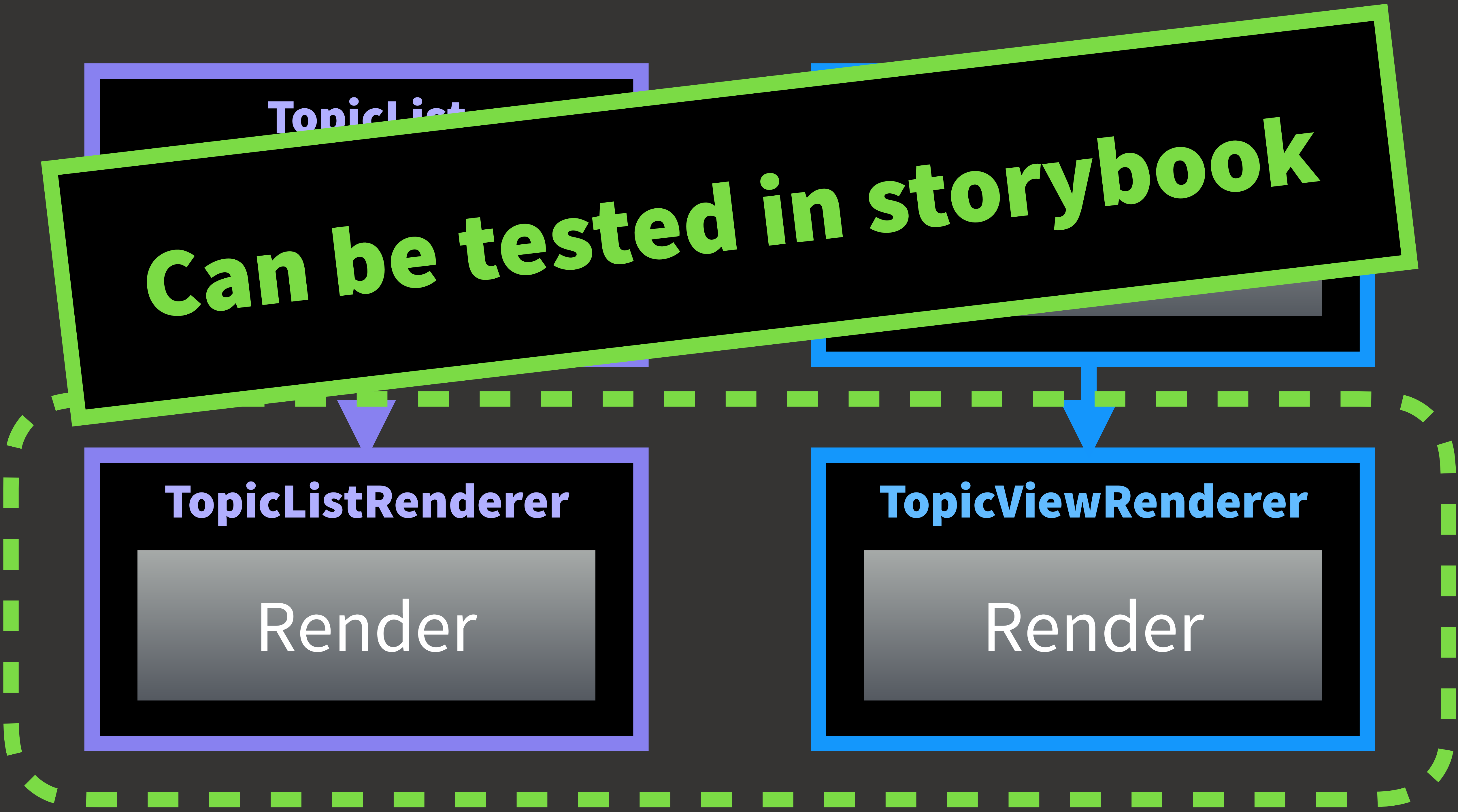
Render

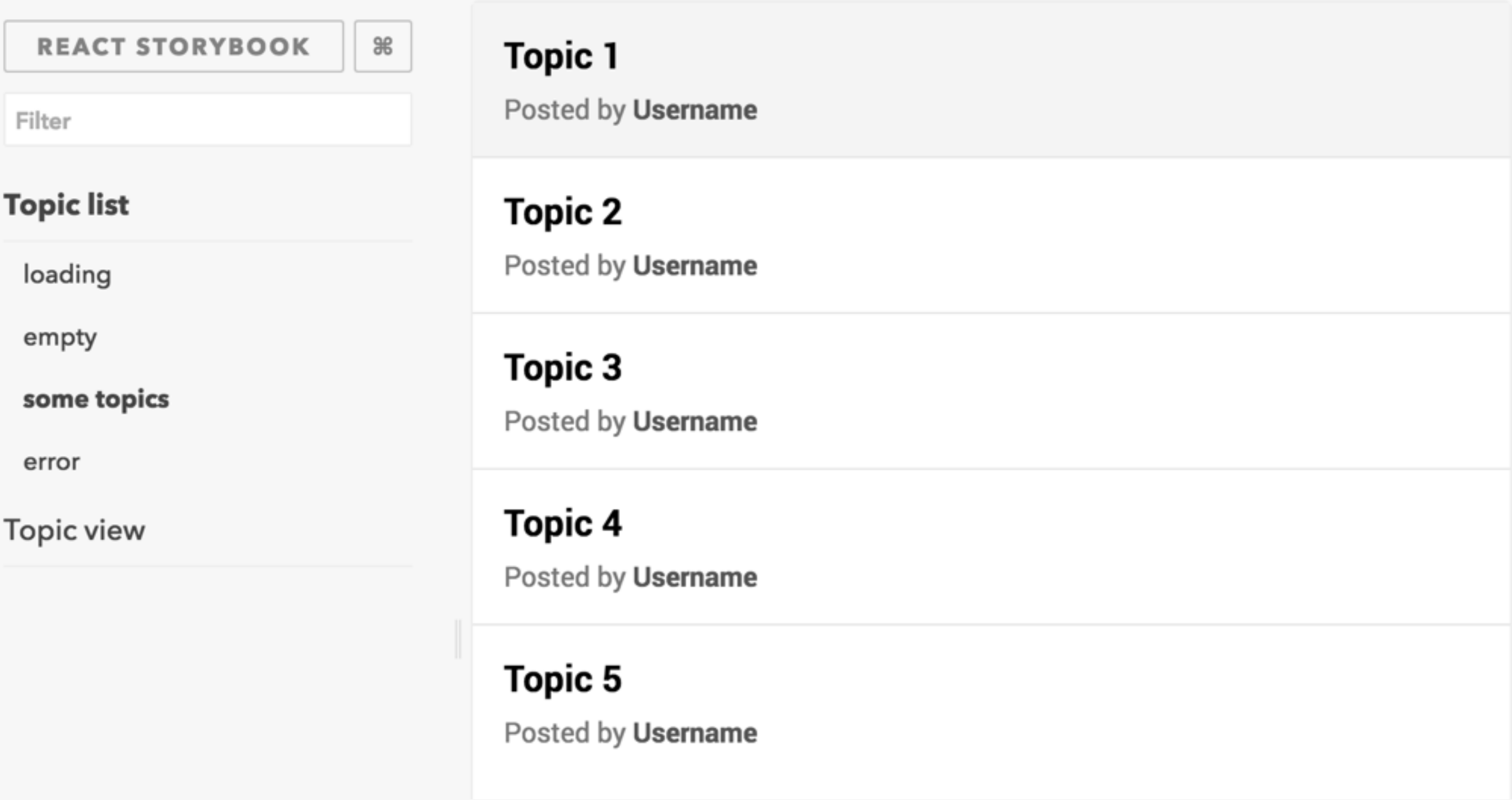
TopicView

Ajax

Render


TopicList
Ajax
TopicListRenderer
Render
TopicView
Ajax
TopicViewRenderer
Render


TopicList
Can be tested in storybook
TopicListRenderer
Render
TopicViewRenderer
Render



Topic 1

Posted by **Username**

Topic 2

Posted by **Username**

Topic 3

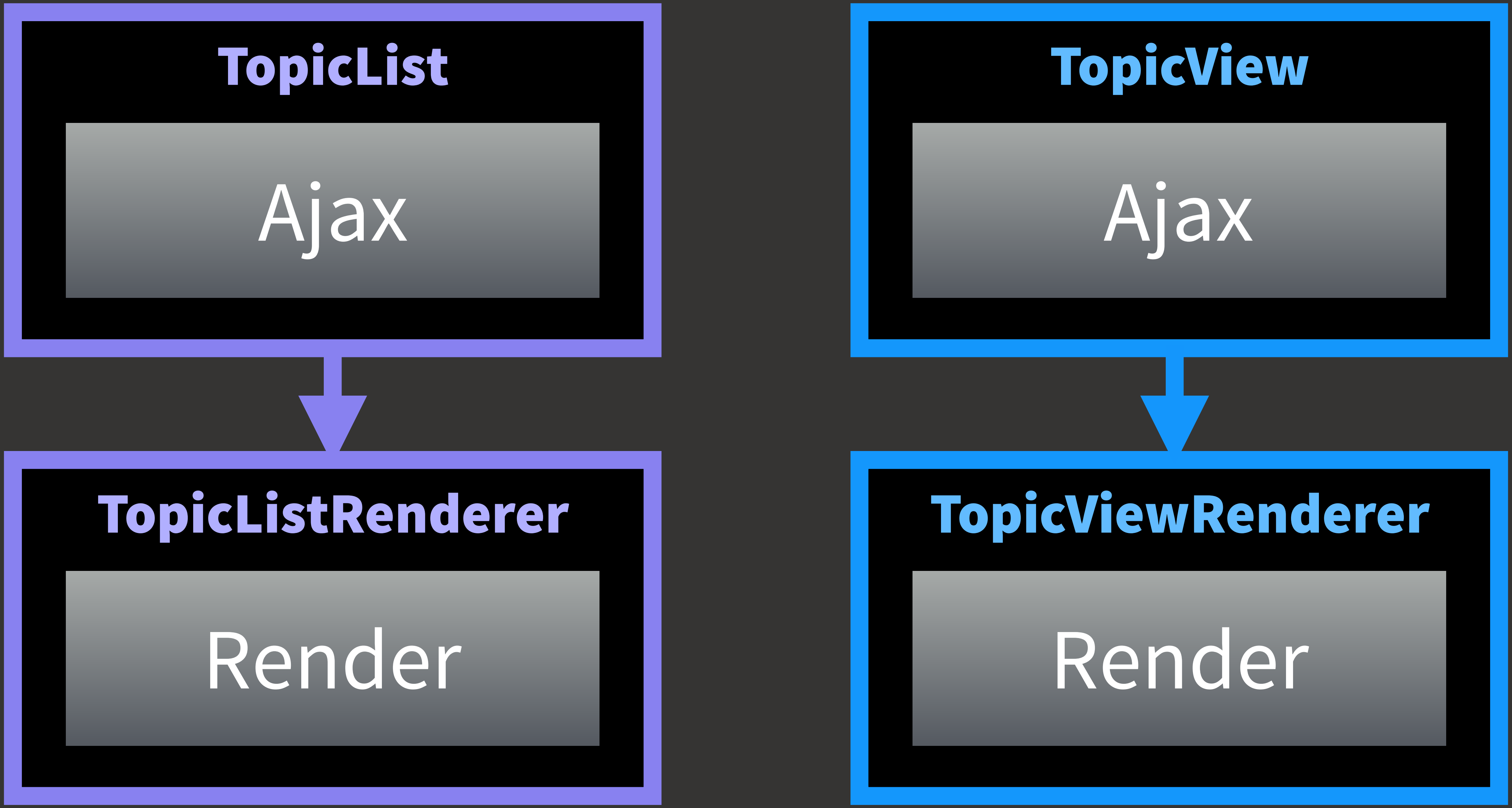
Posted by **Username**

Topic 4

Posted by **Username**

Topic 5

Posted by **Username**



TopicList

Ajax

TopicListRenderer

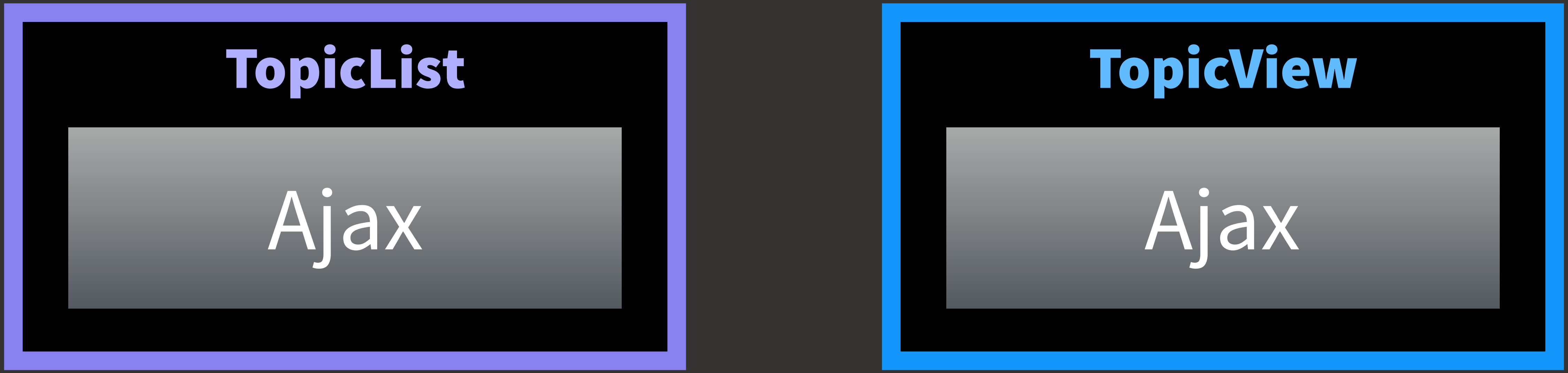
Render

TopicView

Ajax

TopicViewRenderer

Render



TopicList

Ajax

TopicView

Ajax

Let’s write a
higher-order component.

Start with duplicated code

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=commen
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState(
    const offset =
    const start =
    const end = sta
    const url = `ht
    get(url)
    .then((response)
      this.setState(                              onse.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps
    }



                                                {topicId}?_embed=commen


                                                 ponse.data })

    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

Find the differences

// TopicList.js

TopicList

```
export class TopicList extends React.Component {
  constructor (props) {
                  error: false, topics: null }
  
  
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

TopicView

```
export class TopicView extends React.Component {
  constructor (props) {
                 e, error: false, topic: null }
  }
  
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
                   error: false, topics: null }
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009
    get(url)
    .then((response) => {
      this.setState({ loading: false, t         ponse.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
```

TopicList

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
                   , error: false, topic: null }
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
                                true })
                      lhost:3009/topics/${topicId}?_embed=comme
                                : false, topic: response.data })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

TopicView

Generalize

// TopicList.js

// TopicView.js

TopicList | withAjaxResource | TopicView

Generalize

```
export class TopicList extends React.Component {
  constructor (props) {

    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009
    get(url)
    .then((response) => {
      this.setState({ loading: false, t                ponse.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
```

```
export class TopicView extends React.Component {
  constructor (props) {
                   e, error: false, topic: null }

    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)

                                             true })
                                           lhost:3009/topics/${topicId}?_embed=comme

                                            : false, topic: response.data })

    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

topics

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

topic

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${st
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.da
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

topics

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
{topicId}?_embed=comme
ponse.data })
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

topic

Generalize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${st
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.da
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
                                    opicId)

                                    xtProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }

                                              {topicId}?_embed=comme

                                              ponse.data })

      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

topics

resource

topic

Generalize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  com
    i

    }
```

this.props.page

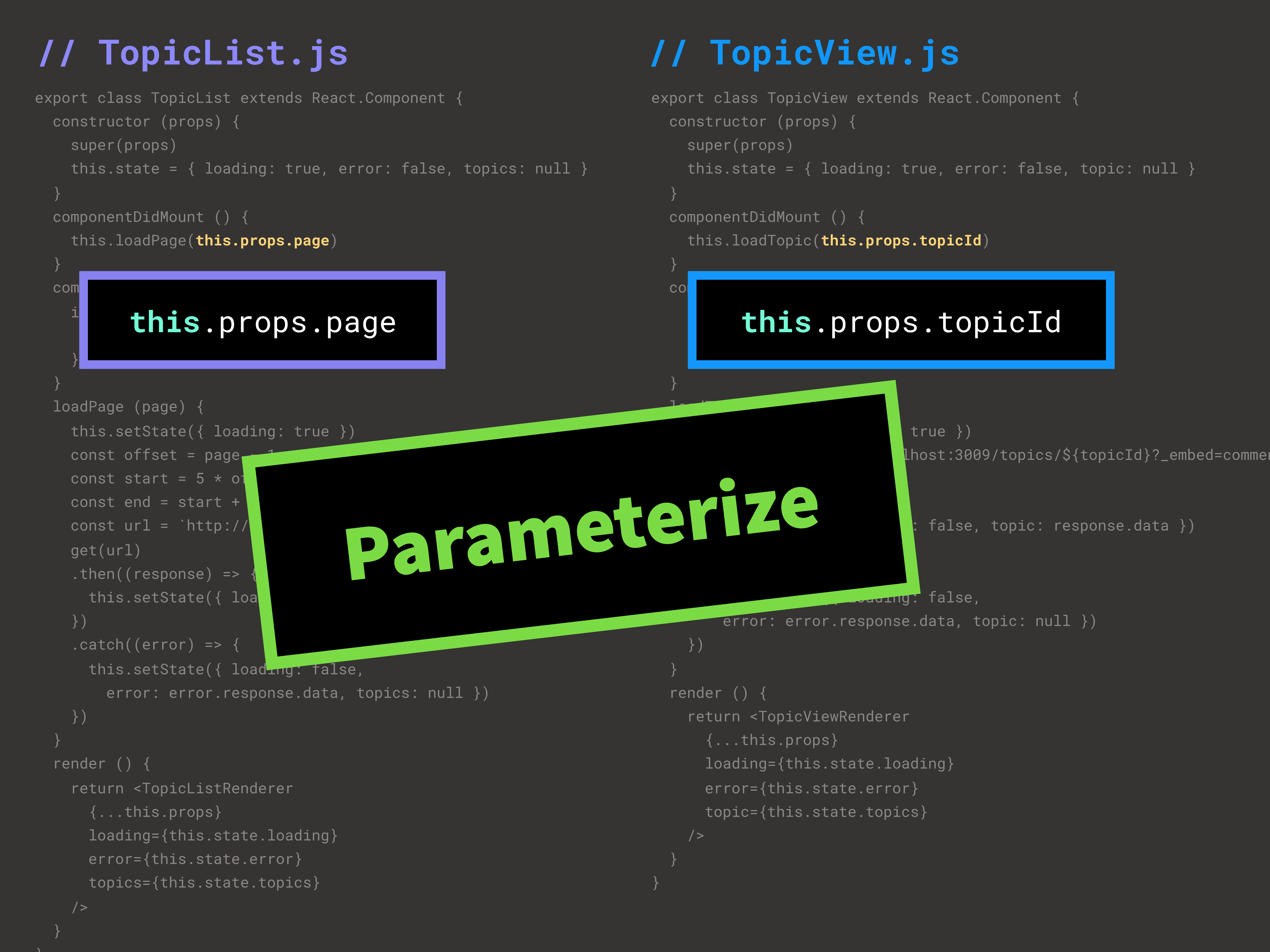
```
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  co
```

this.props.topicId

```
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```



// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  com
    i

    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page
    const start = 5 * o
    const end = start +
    const url = `http://
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
```

this.props.page

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  co

  }

                 true })
                lhost:3009/topics/${topicId}?_embed=comme


                 : false, topic: response.data })



                        ading: false,
      error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

this.props.topicId

Parameterize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  com
    i

    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page
    const start = 5 * of
    const end = start +
    const url = `http://
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

this.props.page

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  co

  }
                   true })
                   lhost:3009/topics/${topicId}?_embed=comme

                   : false, topic: response.data })

                   loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

this.props.topicId

Parameterize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: t
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  com
    i

    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * of
    const end = start +
    const url = `http://
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
                       oading: true, error: false, topic: null }

                         () {
    this.loadTopic(this.props.topicId)
  }
  co

  }
  lo
                       true })
                       lhost:3009/topics/${topicId}?_embed=comme

                       : false, topic: response.data })

                   ({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

getResourceId(this.props)

this.props.page

this.props.topicId

Parameterize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
                                    {
                                   age) {
```

this.loadPage

```
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
                              ) {
                             ps.topicId) {
                              )
```

this.loadTopic

```
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
                                         {
                                     age) {

  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page -
    const start = 5 * off
    const end = start + 5
    const url = `http://lo
    get(url)
    .then((response) => {
      this.setState({ load
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
                                  ) {
                                  ps.topicId) {
                                  )

  }
            picId) {
            e({ loading: true })
            `http://localhost:3009/topics/${topicId}?_embed=commen

            se) => {
            te({ loading: false, topic: response.data })

            => {
            te({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

this.loadPage

this.loadTopic

Generalize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading:
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
                      {
                      age) {

  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page -
    const start = 5 * off
    const end = start + 5
    const url = `http://lo
    get(url)
    .then((response) => {
      this.setState({ load
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    per(props)
    is.state = { loading: true, error: false, topic: null }

    onentDidMount () {
    this.loadTopic(this.props.topicId)
  }
                   ) {
                   ps.topicId) {
                   )

  }
                 icId) {
                 e({ loading: true })
                 `http://localhost:3009/topics/${topicId}?_embed=comme

                 se) => {
                 te({ loading: false, topic: response.data })

                  => {
                 e({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

`this.loadResource`

`this.loadPage`

`this.loadTopic`

Generalize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
```

```
const offset = page - 1
const start = 5 * offset
const end = start + 5
const url = `http://localhost:3009/
topics?_start=${start}&_end=${end}`
```

```
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
```

```
const url = `http://localhost:3009/topics/$
{topicId}?_embed=comments`
```

```
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
```

```
const offset = page - 1
const start = 5 * offset
const end = start + 5
const url = `http://localhost:3009/
topics?_start=${start}&_end=${end}`
```

```
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http:/
    get(url)
    .then((response) =>
      this.setState({ lo
    })
    .catch((error) => {
      this.setState({ loa
        error: error.respo
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
```

```
const url = `http://localhost:3009/topics/$
{topicId}?_embed=comments`
```

```
    this.setState({ loading: true })
    const url = `ht          alhost:3009/topics/${topicId}?_embed=commen

                             g: false, topic: response.data })

                             : false,
                             data, topic: null })

  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

Parameterize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null
  }
```

```
const offset = page - 1
const start = 5 * offset
const end = start + 5
const url = `http://localhost:3009/
topics?_start=${start}&_end=${end}`
```

```
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http:/
    get(url)
    .then((response) =>
      this.setState({ lo
    })
    .catch((error) => {
      this.setState({ loa
        error: error.respo
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
                                        : false, topic: null }
```

```
getUrl(resourceId)
```

```
  }
  componentWillReceiveProps (nextProps) {
```

```
const url = `http://localhost:3009/topics/$
{topicId}?_embed=comments`
```

```
    this.setState({ loading: true })
    const url = `ht            lhost:3009/topics/${topicId}?_embed=comme
                               g: false, topic: response.data })
                               : false,
                               data, topic: null })
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

Parameterize

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
```

TopicListRenderer

```
                                       opics: response.data })

      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=commer
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
```

TopicViewRenderer

```
                                           pic: null })

  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceivePro
    if (thi
      this.
    }
  }
  loadPage (
    this.setS
    const off
    const sta
    const end
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
                                              opics: response.data })
      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
                   pic(this.props.topicId)

                   eceiveProps (nextProps) {
                     ps.topicId !== nextProps.topicId) {
                     opic(nextProps.topicId)

                   cId) {
                    { loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
                                     pic: null })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

Parameterize

TopicListRenderer

TopicViewRenderer

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceivePr
    if (thi
      this.
    }
  }
  loadPage (
    this.setS
    const off
    const sta
    const end
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)

      this.setState({ loading: false,
        error: error.response.data, topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
                   pic(this.props.topicId)

                   eceiveProps (nextProps) {
                     ps.topicId !== nextProps.topicId) {
                     opic(nextProps.topicId)

                     cId) {
                     ({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })

                             pic: null })
  }
  render () {
    return <TopicViewRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

Parameterize

TopicListRenderer | Renderer | TopicViewRenderer

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
```

```
topics={this.state.topics}
```

```
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
  loadTopic (topicId) {
    this.setState({ loading: true })
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    get(url)
    .then((response) => {
      this.setState({ loading: false, topic: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
```

```
topic={this.state.topic}
```

```
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loadi
    const offset = page
    const start = 5 * of
    const end = start +
    const url = `http://l
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
```

```
topics={this.state.topics}
```

```
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
```

Parameterize

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
```

```
                               true })
                               host:3009/topics/${topicId}?_embed=commer
                                 false, topic: response.data })
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
```

```
topic={this.state.topic}
```

```
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loadi
    const offset = page
    const start = 5 * of
    const end = start +
    const url = `http://l
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
```

Parameterize

```
topics={this.state.topics}
```

```
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
                 true })
                 host:3009/topics/${topicId}?_embed=commen
                   false, topic: response.data })
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
```

```
topic={this.state.topic}
```

```
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loadi
    const offset = page
    const start = 5 * of
    const end = start +
    const url = `http://l
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
```

topics={**this.state.**~~topics~~}
resource

```
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      topics={this.state.topics}
    />
  }
}
```

Parameterize

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }

                      true })
                      host:3009/topics/${topicId}?_embed=commen

                        false, topic: response.data })

      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
```

topic={**this.state.**~~topic~~}
resource

```
      error={this.state.error}
      topic={this.state.topics}
    />
  }
}
```

// TopicList.js

```
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loadi
    const offset = page
    const start = 5 * of
    const end = start +
    const url = `http://l
    get(url)
    .then((response) => {
      this.setState({ loa
    })
    .catch((error) => {
      this.setState({ loading: false,
```

```
    return <TopicListRenderer
      {...this.props}
      loading={this
      error={this.s
      topics={this.s
    />
  }
}
```

// TopicView.js

```
export class TopicView extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topic: null }
  }
  componentDidMount () {
    this.loadTopic(this.props.topicId)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.topicId !== nextProps.topicId) {
      this.loadTopic(nextProps.topicId)
    }
  }
```

```
 true })
host:3009/topics/${topicId}?_embed=commer
```

```
 false, topic: response.data })
```

```
      this.setState({ loading: false,
        error: error.response.data, topic: null })
    })
  }
```

```
      error={this.state.error}
```

Parameterize

```
topics={this.state.topics}
```

```
topic={this.state.topic}
```

```
{...resourceToProps(this.state.resource)}
```

The plan

The plan

Generalized name

```
TopicList, TopicView → withAjaxResource
topics,    topic     → resource
loadPage,  loadTopic → loadResource
```

The plan

Generalized name

```
TopicList, TopicView → withAjaxResource
topics,    topic     → resource
loadPage,  loadTopic → loadResource
```

Parameters

```
Renderer
getResourceId(props)
getUrl(resourceId)
resourceToProps(resource)
```

Let’s write a
higher-order component.

```
import React from 'react'
import { get } from 'axios'
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {





}
```

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {



  }
}
```

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {
    return class extends React.Component {
      …
    }
  }
}
```

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {
    return class extends React.Component {
      …
    }
  }
}
```

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {
    return class extends React.Component {
      …
    }
  }
}
```

options →

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {
    return class extends React.Component {
      …
    }
  }
}
```

options → ReactComponent →

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {
    return class extends React.Component {
      …
    }
  }
}
```

options → ReactComponent → ReactComponent

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Rend  er) {
    return class extend  React.Component {
      …
    }
  }
}
```

options → ReactComponent → ReactComponent

Higher-order component

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Rend  er) {
    return class extend  React.Component {
      …
    }
  }
}
```

options → ReactComponent → ReactComponent

Higher-order component factory

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
  getResourceId,
  getUrl,
  resourceToProps
}) {
  return function (Renderer) {
    return class extends React.Component {
      …
    }
  }
}
```

```
import React from 'react'
import { get } from 'axios'

export function withAjaxResource ({
 getResourceId,
 getUrl,
 resourceToProps
}) {
 return function (Renderer) {
  return class extends React.Component {
   constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
   }
   componentDidMount () {
    this.loadPage(this.props.page)
   }
   componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
```

```
    return function (Renderer) {
      return class extends React.Component {
        constructor (props) {
          super(props)
          this.state = { loading: true, error: false, topics: null }
        }
        componentDidMount () {
          this.loadPage(this.props.page)
        }
        componentWillReceiveProps (nextProps) {
          if (this.props.page !== nextProps.page) {
            this.loadPage(nextProps.page)
          }
        }
        loadPage (page) {
          this.setState({ loading: true })
          const offset = page - 1
          const start = 5 * offset
          const end = start + 5
          const url = `http://localhost:3009/topics?_start=${start}&
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, topics: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, topics: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, topics: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, topics: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadPage(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(this.props.page)
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (this.props.page !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (getResourceId(this.props) !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (getResourceId(this.props) !== nextProps.page) {
        this.loadPage(nextProps.page)
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (getResourceId(this.props) !== getResourceId(nextProps)) {
        this.loadPage(getResourceId(nextProps))
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (getResourceId(this.props) !== getResourceId(nextProps)) {
        this.loadPage(getResourceId(nextProps))
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
```

```
return function (Renderer) {
  return class extends React.Component {
    constructor (props) {
      super(props)
      this.state = { loading: true, error: false, resource: null }
    }
    componentDidMount () {
      this.loadResource(getResourceId(this.props))
    }
    componentWillReceiveProps (nextProps) {
      if (getResourceId(this.props) !== getResourceId(nextProps)) {
        this.loadResource(getResourceId(nextProps))
      }
    }
    loadPage (page) {
      this.setState({ loading: true })
      const offset = page - 1
      const start = 5 * offset
      const end = start + 5
      const url = `http://localhost:3009/topics?_start=${start}&_end=$
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=$
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
    this.loadResource(getResourceId(nextProps))
  }
}
loadPage (page) {
  this.setState({ loading: true })
  const offset = page - 1
  const start = 5 * offset
  const end = start + 5
  const url = `http://localhost:3009/topics?_start=${start}&_end=$
  get(url)
  .then((response) => {
    this.setState({ loading: false, topics: response.data })
  })
  .catch((error) => {
    this.setState({ loading: false, error: error.response.data,
      topics: null })
  })
}
render () {
  return <TopicListRenderer
```

```
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=$
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
    this.loadResource(getResourceId(nextProps))
  }
}
loadResource (page) {
  this.setState({ loading: true })
  const offset = page - 1
  const start = 5 * offset
  const end = start + 5
  const url = `http://localhost:3009/topics?_start=${start}&_end=$
  get(url)
  .then((response) => {
    this.setState({ loading: false, topics: response.data })
  })
  .catch((error) => {
    this.setState({ loading: false, error: error.response.data,
      topics: null })
  })
}
render () {
  return <TopicListRenderer
```

```
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadResource (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=$
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
```

```
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadResource (resourceId) {
    this.setState({ loading: true })
    const offset = resourceId - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=$
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
    this.loadResource(getResourceId(nextProps))
  }
}
loadResource (resourceId) {
  this.setState({ loading: true })
  const offset = resourceId - 1
  const start = 5 * offset
  const end = start + 5
  const url = `http://localhost:3009/topics?_start=${start}&_end=$
  get(url)
  .then((response) => {
    this.setState({ loading: false, topics: response.data })
  })
  .catch((error) => {
    this.setState({ loading: false, error: error.response.data,
      topics: null })
  })
}
render () {
  return <TopicListRenderer
```

```
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadResource (resourceId) {
    this.setState({ loading: true })
    const url = getUrl(resourceId)




    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
```

```
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadResource (resourceId) {
    this.setState({ loading: true })
    const url = getUrl(resourceId)
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
      this.loadResource(getResourceId(nextProps))
    }
  }
  loadResource (resourceId) {
    this.setState({ loading: true })
    const url = getUrl(resourceId)
    get(url)
    .then((response) => {
      this.setState({ loading: false, topics: response.data })
    })
    .catch((error) => {
      this.setState({ loading: false, error: error.response.data,
        topics: null })
    })
  }
  render () {
    return <TopicListRenderer
      {...this.props}
      loading={this.state.loading}
      error={this.state.error}
    this.loadResource(getResourceId(nextProps))
  }
}
loadResource (resourceId) {
  this.setState({ loading: true })
  const url = getUrl(resourceId)
  get(url)
  .then((response) => {
    this.setState({ loading: false, resource: response.data })
  })
  .catch((error) => {
    this.setState({ loading: false, error: error.response.data,
      resource: null })
  })
}
render () {
  return <TopicListRenderer
    {...this.props}
    loading={this.state.loading}
    error={this.state.error}
```

```
        this.setState({ loading: false, resource: response.data })
      })
      .catch((error) => {
        this.setState({ loading: false, error: error.response.data,
          resource: null })
      })
    }
    render () {
      return <TopicListRenderer
        {...this.props}
        loading={this.state.loading}
        error={this.state.error}
        topics={this.state.topics}
      />
    }
  }
}
```

```
        this.setState({ loading: false, resource: response.data })
      })
      .catch((error) => {
        this.setState({ loading: false, error: error.response.data,
          resource: null })
      })
    }
    render () {
      return <TopicListRenderer
        {...this.props}
        loading={this.state.loading}
        error={this.state.error}
        topics={this.state.topics}
      />
    }
  }
}
```

```
        this.setState({ loading: false, resource: response.data })
      })
      .catch((error) => {
        this.setState({ loading: false, error: error.response.data,
          resource: null })
      })
    }
    render () {
      return <TopicListRenderer
        {...this.props}
        loading={this.state.loading}
        error={this.state.error}
        topics={this.state.topics}
      />
    }
  }
}
```

```
        this.setState({ loading: false, resource: response.data })
      })
      .catch((error) => {
        this.setState({ loading: false, error: error.response.data,
          resource: null })
      })
    }
    render () {
      return <Renderer
        {...this.props}
        loading={this.state.loading}
        error={this.state.error}
        topics={this.state.topics}
      />
    }
  }
}
```

```
        this.setState({ loading: false, resource: response.data })
      })
      .catch((error) => {
        this.setState({ loading: false, error: error.response.data,
          resource: null })
      })
    }
    render () {
      return <Renderer
        {...this.props}
        loading={this.state.loading}
        error={this.state.error}
        topics={this.state.topics}
      />
    }
  }
}
```

```
        this.setState({ loading: false, resource: response.data })
      })
      .catch((error) => {
        this.setState({ loading: false, error: error.response.data,
          resource: null })
      })
    }
    render () {
      return <Renderer
        {...this.props}
        loading={this.state.loading}
        error={this.state.error}
        {...resourceToProps(this.state.resource)}
      />
    }
  }
}
```

```
// TopicList.js
// TopicList.js
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
```

```
// TopicList.js
export class TopicList extends React.Component {
  constructor (props) {
    super(props)
    this.state = { loading: true, error: false, topics: null }
  }
  componentDidMount () {
    this.loadPage(this.props.page)
  }
  componentWillReceiveProps (nextProps) {
    if (this.props.page !== nextProps.page) {
      this.loadPage(nextProps.page)
    }
  }
  loadPage (page) {
    this.setState({ loading: true })
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
```

```
// TopicList.js
export const TopicList =
// TopicList.js
export const TopicList = withAjaxResource
```

```
// TopicList.js
export const TopicList = withAjaxResource
```

options → ReactComponent → ReactComponent

```
// TopicList.js
export const TopicList = withAjaxResource({
  getResourceId (props) {
    return props.page
  },
  getUrl (page) {
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=${
    return url
  },
  resourceToProps (topics) {
    return { topics }
  }
})
```

ReactComponent → ReactComponent

```
// TopicList.js
export const TopicList = withAjaxResource({
  getResourceId (props) {
    return props.page
  },
  getUrl (page) {
    const offset = page - 1
    const start = 5 * offset
    const end = start + 5
    const url = `http://localhost:3009/topics?_start=${start}&_end=${
    return url
  },
  resourceToProps (topics) {
    return { topics }
  }
})(TopicListRenderer)
```

ReactComponent

```
// TopicView.js
// TopicView.js
export const TopicView = withAjaxResource({
  getResourceId (props) {
    return props.topicId
  },
  getUrl (topicId) {
    const url = `http://localhost:3009/topics/${topicId}?_embed=comme
    return url
  },
  resourceToProps (topic) {
    return { topic }
  }
})(TopicViewRenderer)
```

Higher order components.

Functions that takes a React component and returns another React component

ReactComponent → ReactComponent

Higher order components.

A kind of abstraction technique, a pattern.

Compared to OOP.

Compared to OOP.

Similar to …?

Compared to OOP.

Similar to decorator pattern.

Benefits

Benefits

Less code duplication.

Benefits

Less code duplication.

Higher-order components encapsulate common logic shared between components.

Benefits

Less code duplication.

Higher-order components encapsulate common logic shared between components.

If that common logic has a bug,
it can be fixed in one place.

Benefits

Less code duplication.

Higher-order components encapsulate common logic shared between components.

If that common logic has a bug,
it can be fixed in one place.

Can be composed together.

Challenge

Challenge

More complex code.

Challenge

More complex code.

Expensive to change
when the abstraction is wrong.
(Happens to all kind of abstractions)

Advice

Advice

Don’t use higher-order components until necessary.

Advice

Don’t use higher-order components until necessary.

But deal with duplication.

Advice

Make it work *first*, *then* make it right.

Advice

Make it work ***first***, *then* make it right.

Advice

Make it work *first*, ***then*** make it right.

Advice

Always challenge the design.

Refactor continuously.

Learn to recognize code smells and anti-patterns.

https://sourcemaking.com/

Learn object-oriented design and design patterns.

https://sourcemaking.com/

Pursue
well-factored
code

Learn functional programming techniques.

Currying
Higher-order functions
Immutability
Map, Filter, Reduce
Persistent data structures
Pure functions

Pursue predictable code

OOP & FP
goes hand in hand

> **“Continuous attention to technical excellence and good design enhances agility.”**
>
> *—Principles behind the Agile manifesto*

http://www.agilemanifesto.org/principles.html

Back to higher-order components

Real-world example:

connect() from react-redux

```
export default connect(
  mapStateToProps, mapDispatchToProps)(Component)
```

```
export default connect(
  mapStateToProps, mapDispatchToProps)(Component)
```

```
function connect(mapStateToProps, mapDispatchToProps) {
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
function connect(mapStateToProps, mapDispatchToProps) {
  return function (WrappedComponent) {
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
function connect(mapStateToProps, mapDispatchToProps) {
  return function (WrappedComponent) {
    return class extends React.Component {
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
function connect(mapStateToProps, mapDispatchToProps) {
  return function (WrappedComponent) {
    return class extends React.Component {
      render() {
        return (
          <WrappedComponent
            {...this.props}
            {...mapStateToProps(store.getState(), this.props)}
            {...mapDispatchToProps(store.dispatch, this.props)}
          />
        )
      }
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
          <WrappedComponent
            {...this.props}
            {...mapStateToProps(store.getState(), this.props)}
            {...mapDispatchToProps(store.dispatch, this.props)}
          />
        )
      }




    }
  }
}
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
          <WrappedComponent
            {...this.props}
            {...mapStateToProps(store.getState(), this.props)}
            {...mapDispatchToProps(store.dispatch, this.props)}
          />
        )
      }
      componentDidMount() {
        this.unsubscribe = store.subscribe(() => this.handleChange())
      }




    }
  }
}
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
          <WrappedComponent
            {...this.props}
            {...mapStateToProps(store.getState(), this.props)}
            {...mapDispatchToProps(store.dispatch, this.props)}
          />
        )
      }
      componentDidMount() {
        this.unsubscribe = store.subscribe(() => this.handleChange())
      }
      componentWillUnmount() {
        this.unsubscribe()
      }




    }
  }
}
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
          <WrappedComponent
            {...this.props}
            {...mapStateToProps(store.getState(), this.props)}
            {...mapDispatchToProps(store.dispatch, this.props)}
          />
        )
      }
      componentDidMount() {
        this.unsubscribe = store.subscribe(() => this.handleChange())
      }
      componentWillUnmount() {
        this.unsubscribe()
      }
      handleChange() {
        this.forceUpdate()
      }
    }
  }
}
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

```
function connect(mapStateToProps, mapDispatchToProps) {
  return function (WrappedComponent) {
    return class extends React.Component {
      render() {
        return (
         <WrappedComponent
           {...this.props}
           {...mapStateToProps(store.getState(), this.props)}
            {...mapDispatchToProps(store.dispatch, this.props)}
          />
        )
      }
      componentDidMount() {
        this.unsubscribe = store.subscribe(() => this.handleChange())
      }
      componentWillUnmount() {
        this.unsubscribe()
      }
      handleChange() {
        this.forceUpdate()
      }
    }
  }
}
```

https://gist.github.com/gaearon/1d19088790e70ac32ea636c025ba424e

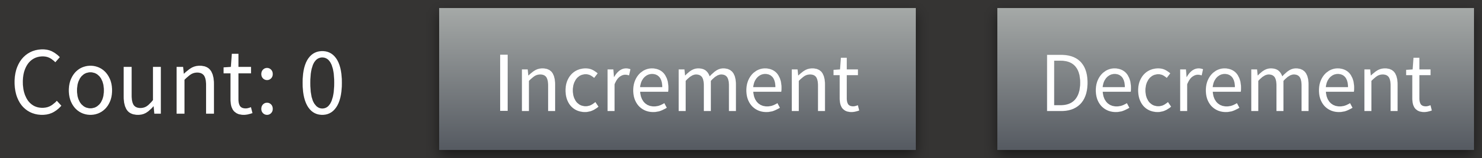
acdlite/recompose

Utility belt for higher-order components

```
function Counter ({ counter, setCounter }) {








}
```

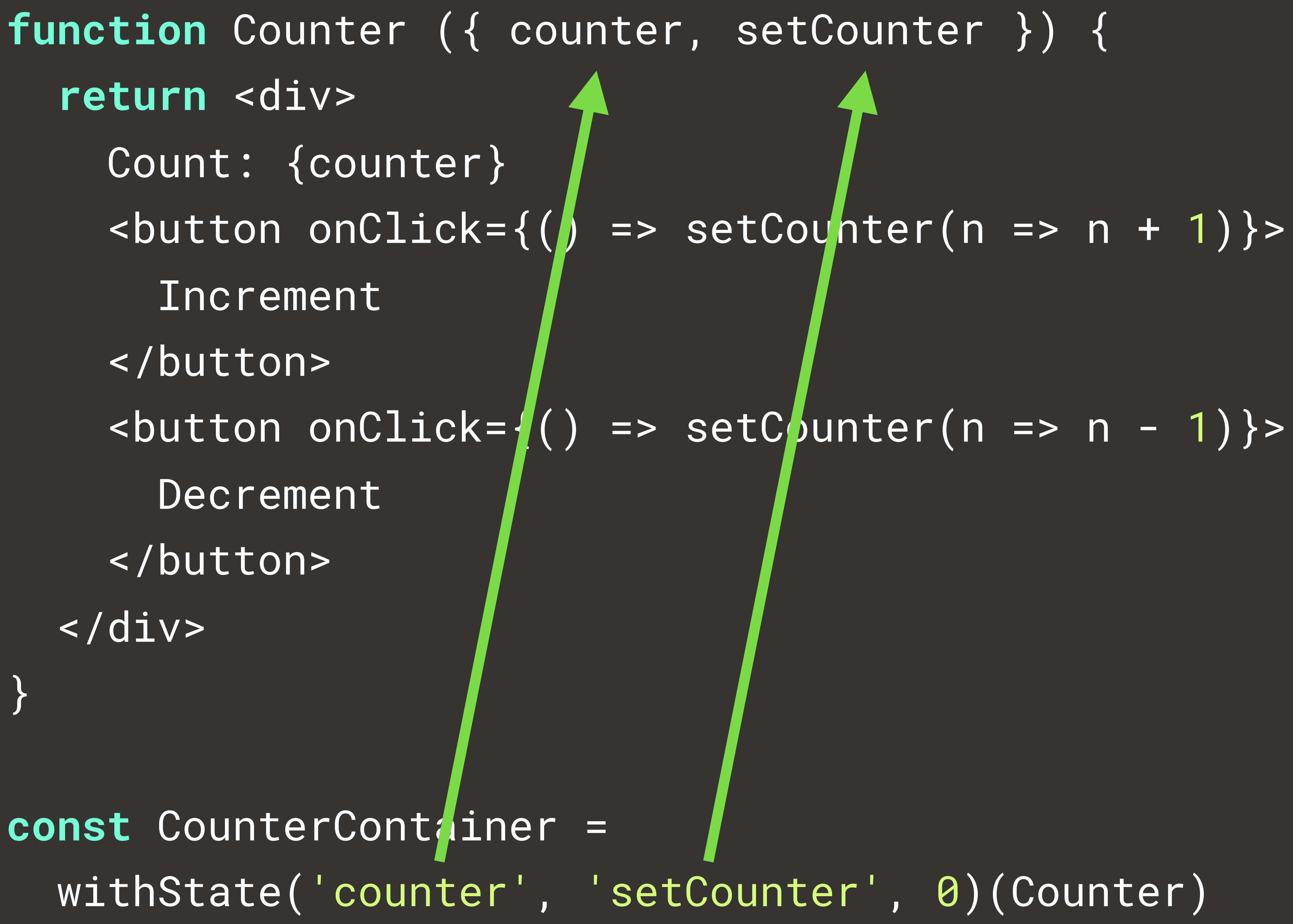
```
function Counter ({ counter, setCounter }) {
  return <div>
    Count: {counter}
    <button onClick={() => setCounter(n => n + 1)}>
      Increment
    </button>
    <button onClick={() => setCounter(n => n - 1)}>
      Decrement
    </button>
  </div>
}
```

```
function Counter ({ counter, setCounter }) {
  return <div>
    Count: {counter}
    <button onClick={() => setCounter(n => n + 1)}>
      Increment
    </button>
    <button onClick={() => setCounter(n => n - 1)}>
      Decrement
    </button>
  </div>
}
```



Count: 0

```
function Counter ({ counter, setCounter }) {
  return <div>
    Count: {counter}
    <button onClick={() => setCounter(n => n + 1)}>
      Increment
    </button>
    <button onClick={() => setCounter(n => n - 1)}>
      Decrement
    </button>
  </div>
}

const CounterContainer =
  withState('counter', 'setCounter', 0)(Counter)
```



```
function Counter ({ counter, setCounter }) {
  return <div>
    Count: {counter}
    <button onClick={() => setCounter(n => n + 1)}>
      Increment
    </button>
    <button onClick={() => setCounter(n => n - 1)}>
      Decrement
    </button>
  </div>
}

const CounterContainer =
  withState('counter', 'setCounter', 0)(Counter)
```

```
function Counter ({ counter, setCounter }) {
  return <div>
    Count: {counter}
    <button onClick={() => setCounter(n => n + 1)}>
      Increment
    </button>
    <button onClick={() => setCounter(n => n - 1)}>
      Decrement
    </button>
  </div>
}

const CounterContainer =
  withState('counter', 'setCounter', 0)(Counter)
```

Initial value

```
}

const CounterContainer =
  withState('counter', 'setCounter', 0)(Counter)
```

```
}

const CounterContainer =
  withState('counter', 'setCounter', 0)(Counter)
```

```
class CounterContainer extends React.Component {
  constructor (props) {
    super(props)
    this.state = { counter: 0 }
  }
  setCounter = (counter) => {
    this.setState({ counter })
  }
  render () {
    return <Counter {...this.props}
      counter={this.state.counter}
      setCounter={this.setCounter}
    />
  }
}
```

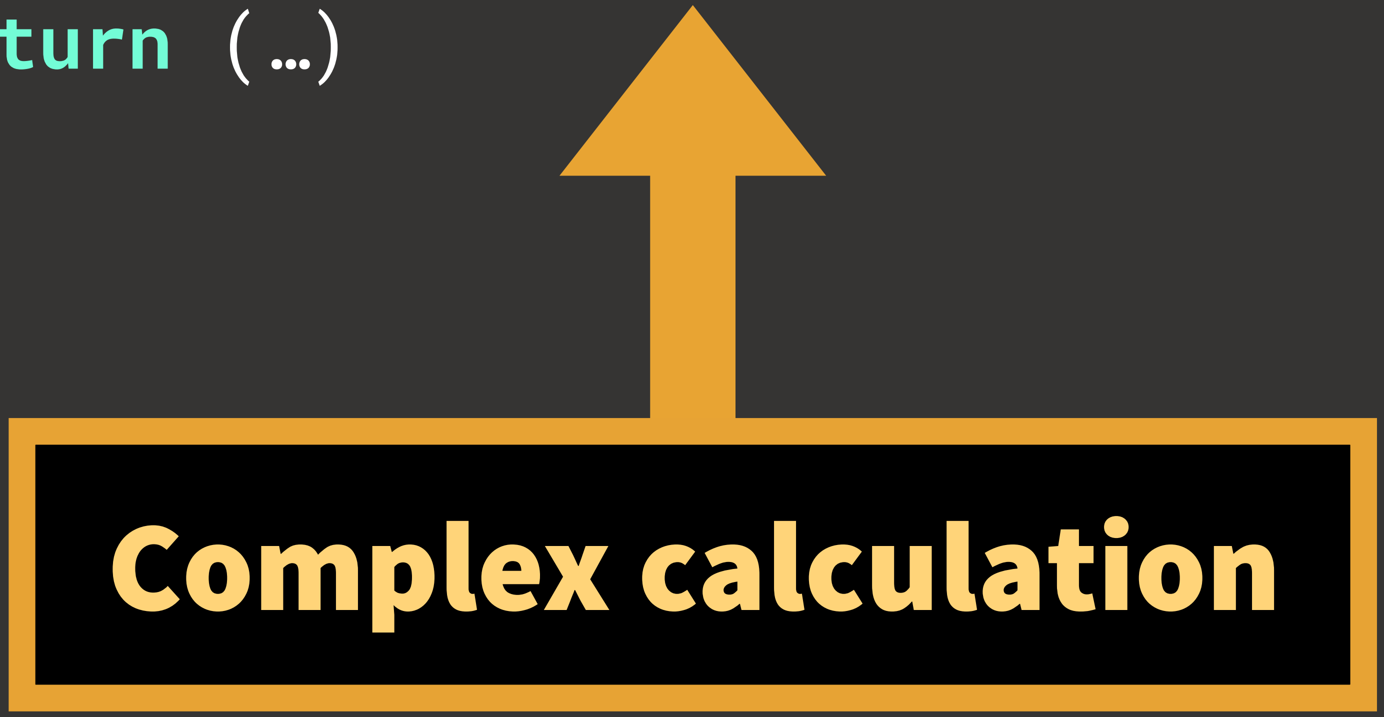
```
const CounterContainer =
  withState('counter', 'setCounter', 0)(Counter)
```

Optimizing rendering performance

```
function MyComponent (props) {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
}
```

```
function MyComponent (props) {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
}
```

```
function MyComponent (props) {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
}
```



Complex calculation

```
function MyComponent (props) {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
}
```

If props don’t change, component should not re-render.

```
function MyComponent (props) {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
}
```

```
class MyComponent extends React.Component {
  render () {
    const presenter =
      calculateComplexDataFrom(this.props.data)
    return (…)
  }
}
```

```
class MyComponent extends React.Component {
  render () {
    const presenter =
      calculateComplexDataFrom(this.props.data)
    return (…)
  }
  shouldComponentUpdate (nextProps, nextState) {


  }
}
```

```
class MyComponent extends React.Component {
  render () {
    const presenter =
      calculateComplexDataFrom(this.props.data)
    return (…)
  }
  shouldComponentUpdate (nextProps, nextState) {
    return shallowCompare(this,
      nextProps, nextState)
  }
}
```

```
function MyComponent (props) {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
}
```

```
const MyComponent = pure((props) => {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
})
```

```
const MyComponent = pure((props) => {
  const presenter =
    calculateComplexDataFrom(props.data)
  return (…)
})
```

from **recompose**

acdlite/recompose

Utility belt for higher-order components

```
mapProps(propsMapper)
withProps(createProps)
withPropsOnChange(shouldMapOrKeys, createProps)
withHandlers(handlerCreators)
defaultProps(props)
renameProp(oldName, new Name)
renameProps(nameMap)
flattenProp(propName)
withState(stateName, stateUpdaterName, initialState)
withReducer(stateName, dispatchName, reducer, initialState)
branch(test, left, right)
renderComponent(component)
renderNothing
shouldUpdate(test)
pure
onlyUpdateForKeys(keys)
onlyUpdateForPropTypes,
withContext(childContextTypes, getChildContext)
getContext(contextTypes)
lifecycle(spec)
toClass
```

```
mapProps(propsMapper)
withProps(createProps)
withPropsOnChange(shouldMapOrKeys, createProps)
withHandlers(handlerCreators)
defaultProps(props)
renameProp(oldName, new Name)
renameProps(nameMap)
flattenProp(propName)
withState(stateName, stateUpdaterName, initialState)
withReducer(stateName, dispatchName, reducer, initialState)
branch(test, left, right)
renderComponent(component)
renderNothing
shouldUpdate(test)
pure
onlyUpdateForKeys(keys)
onlyUpdateForPropTypes,
withContext(childContextTypes, getChildContext)
getContext(contextTypes)
lifecycle(spec)
toClass
```

Conclusions

Learn more

About higher-order components

- ***Mixins Are Dead. Long Live Composition***. Dan Abramov
  https://medium.com/@dan_abramov/mixins-are-dead-long-live-higher-order-components-94a0d2f9e750

- **recompose**
  https://github.com/acdlite/recompose

- ***Recomposing your React Application***. Andrew Clark
  https://www.youtube.com/watch?v=zD_judE-bXk

Thank you!