Повторное использование кода с помощью компонентов высшего порядка в React

Дмитрий Цепелев, AnjLab

AnjLab Tech Talks #9, 16 марта 2018

Демо: пример приложения

Пример можно взять тут:
https://github.com/DmitryTsepelev/recompose-talk-demo

Что хотелось бы сделать лучше

- не показывать табличку если ничего не было загружено
- не дублировать логику отображения спиннера
- избежать перерисовки элементов строк таблицы товаров и чата
    - использовать PureComponent (если в props много всего - то будет медленно)
    - shouldComponentUpdate (если реализовать самому - будет быстрее, но реализация многословная)

Демо: решаем проблемы

Компонент высшего порядка (HOC)

```
const Name = () ⇒ (<p>Guest</p>)

const withGreeting = WrappedComponent ⇒
  return class extends Component {
    render() {
      return (
        <div>
          Hi, <WrappedComponent {...this.props} />
        </div>
      )
    }
  }

withGreeting(Name)
```

Демо: пробуем переписать все на HOC

Композиция

```
withRouter(
  injectIntl(
    connect(mapStateToProps, mapDispatchToProps)
  )
)(SignUpForm)
```

```
const enhance = compose(
  withRouter,
  injectIntl,
  connect(mapStateToProps, mapDispatchToProps)
)

enhance(SignUpForm)
```

Демо: реализуем функцию *compose*

Recompose

- functional components
- повышение повторного использования кода
- оптимизация производительности
- можно тестировать поведение отдельно

Демо: компоненты высшего порядка в *recompose*

Зачем нужен componentFromProp

```
const enhance = compose(
  withProps({ className: 'w-100' }),

  branch(
    ({ to }) => to,

    withProps(() => ({
      activeClassName: 'b',
      component: Link
    }))
  ),

  branch(
    ({ onClick }) => onClick,

    withProps(({ className }) => ({
      className: `${className} pointer`,
      component: 'a'
    }))
  )
);

export default enhance(componentFromProp('component'));
```

Загрузка данных

```
const mapStateToProps = (state, { match }) => {
  const taskId = match.params && match.params.taskId
  let props = { taskId }

  const task = getTask(state, taskId)
  if (task) {
    props = { ...props, task }
  }

  return props
}

const enhance = compose(
  onlyUpdateForPropTypes,

  setPropTypes({
    task: PropTypes.object
  }),

  connect(mapStateToProps),

  lifecycle({
    componentDidMount() {
      requestTask(this.props.taskId)
    }
  })
)
```

Спасибо за внимание!

dmitry.a.tsepelev@gmail.com
https://twitter.com/dmitrytsepelev
https://github.com/DmitryTsepelev