



Сборка\_

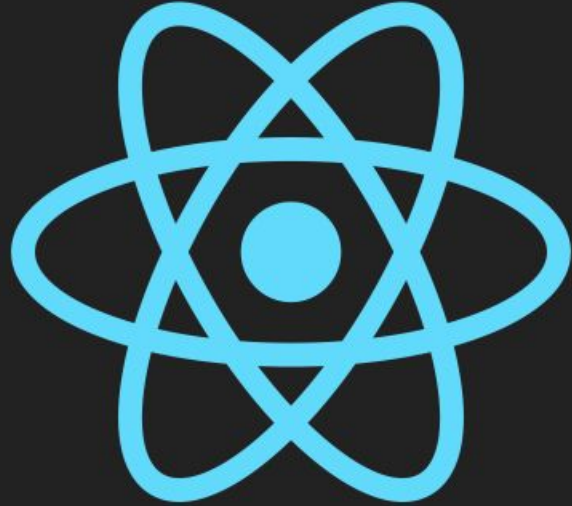


АТВИНТА



GoodLine

Оператор связи Кузбасса



Разработка на React.js

# Компонент - это функция



```
const GreetWorld = props => (  
  <h1>{props.greeting}, World!</h1>  
);  
  
const MOUNT_NODE = document.getElementById('app')  
  
ReactDOM.render(  
  <GreetWorld greeting="Hello" />, MOUNT_NODE  
);
```

# Приложение - это function от state



```
function App(state) {  
  // view of state  
}
```

React  FP

React != Reactivity

# Виды КОМПОНЕНТОВ

# Functional components

Stateless components





```
const UsersList = ({ users }) => (  
  <ul>  
    {users.map(user =>  
      <li key={user.id}>{user.name}</li>  
    )}  
  </ul>  
);
```

# Container components

Stateful components

```
class GithubUsers extends Component {
  state = {
    users: []
  };

  componentDidMount() {
    fetch('api.github.com/users')
      .then(response => response.json())
      .then((users) => {
        this.setState({ users })
      });
  }

  render() {
    <UsersList users={this.state.users} />
  }
}
```

# **React.Context**

Provider & Consumer

```
const ThemeContext = React.createContext('light')

class ThemeProvider extends React.Component {
  state = {theme: 'light'}
  render() {
    return (
      <ThemeContext.Provider value={this.state.theme}>
        {this.props.children}
      </ThemeContext.Provider>
    )
  }
}

const App = () => (
  <ThemeProvider>
    <ThemeContext.Consumer>
      {theme => <Button theme={theme}>Click</Button>}
    </ThemeContext.Consumer>
  </ThemeProvider>
);
```

# Higher-Order Components

(HOC)



```
const enhance = compose(  
  withRouter,  
  withUser  
);  
  
export default enhance(UserPage);
```

```
const enhance = compose(  
  useState('value', 'updateValue', ''),  
  withHandlers({  
    onChange: props => event => {  
      props.updateValue(event.target.value)  
    },  
    onSubmit: props => event => {  
      event.preventDefault()  
      submitForm(props.value)  
    }  
  })  
)
```

```
const Form = enhance(({ value, onChange, onSubmit }) =>  
  <form onSubmit={onSubmit}>  
    <label>Value  
      <input type="text" value={value} onChange={onChange} />  
    </label>  
  </form>  
)
```



~~Side Effect~~ Component



```
const DocumentTitle = (props) => {  
  document.title = props.title  
  return null;  
};
```

# **Хранение данных**

и бизнес-логика



**Redux**



**ACTION**



**REDUCER**



**STORE**






```
const increment = () => ({ type: 'INCREMENT' });  
const decrement = () => ({ type: 'DECREMENT' });  
  
const setValue = value => ({  
  type: 'SET_VALUE',  
  payload: value  
});
```



```
function reducer(state = 0, action) {  
  switch (action.type) {  
    case 'INCREMENT':  
      return state + 1  
    case 'DECREMENT':  
      return state - 1  
    default:  
      return state  
  }  
}
```



```
export const rootReducer = combineReducers({
  auth: auth.reducer,
  entities: combineReducers({
    users: users.reducers.entities,
    comments: comments.reducers.entities
  }),
  ui: combineReducers({
    users: users.reducers.ui,
    comments: comments.reducers.ui
  }),
  alerts
});
```



# Отладка

инструменты разработчика

## Things to do



- Inspect all the things
- Profit!!
- Profit!!

All

Completed

Remaining

Elements Network Sources Timeline Profiles Resources Audits Console | NReact

```
<Wrap more=["a",2,"c",...] str="thing" awesome=1>
  <div>
    <div style={position: "absolute", top: 20, left: 20, ...}>this is an iframe</div>
    <Todos>
      <div style={fontSize: 20, fontFamily: "sans-serif", padding: 30, ...}>
        <h1 style={margin: 0, fontSize: 25, marginBottom: 10}>Things to do</h1>
        <NewTodo onAdd=fn()>_</NewTodo>
        <TodoItems todos=[{}], filter="All" onToggleComplete=fn()>
          <ul style={listStyle: "none", textAlign: "left", margin: 0, ...}>
            <TodoItem item={title: "Inspect all the things", completed: true, id: 10} onTogg...
            <TodoItem item={title: "Profit!!", completed: false, id: 11} onToggle=fn()>_</Tc...
            <TodoItem item={title: "Profit!!", completed: false, id: 12} onToggle=fn()>_</Tc...
          </ul>
        </TodoItems>
        <Filter onSort=fn() onFilter=fn() filter="All">_</Filter>
      </div>
    </Todos>
    <OldStyle awesome=2>_</OldStyle>
  </div>
</Wrap>
```

**<TodoItem>** (\$r in the console)  
Props read-only  
▶ item: {}  
▶ onToggle: fn()

Search by Component Name

Reset

Revert

Sweep

Commit

todo

@@INIT

▼ state: {} 1 key

▶ todos: [] 1 item

▼ *What needs to be done?*

Use Redux

1 item left

All

Active

Completed

# Альтернативы Redux

GraphQL & Relay

<https://developer.github.com/v4/explorer/>

Рендеры

# Рендеры

- `react-dom`
- `react-native`
- `react-pdf`
- `redux`
- `react-canvas`
- `react-konsole`
- `react-vr`
- `react-tv`
- ...

**Спасибо!**