
Яндекс

Яндекс

State of BEM 2019

Эволюция Большого фронтенда

Хохулин Алексей
@xoxulin

Яндекс

фантастические фильмы 2019

Найти

Алексей Хохулин

Поиск Картинки Видео Карты Маркет Новости Эфир Коллекции Знатоки Услуги Ещё

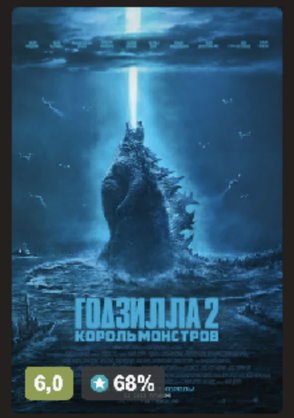
Каталог › Фильмы

фантастика | по популярности | 2019

163 оценки Развернуть

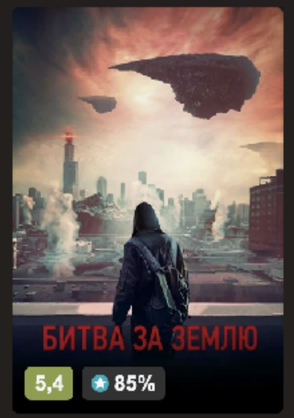
Дитя робота
2019, ужасы, фантастика

Люди в чёрном: Интернэшнл
2019, фантастика, боевик



Годзилла 2: Король монстров
2019, фантастика, фэнтези

Шазам!
2019, фэнтези, боевик



Битва за Землю
2019, фантастика, триллер

Стекло
2019, фантастика, триллер

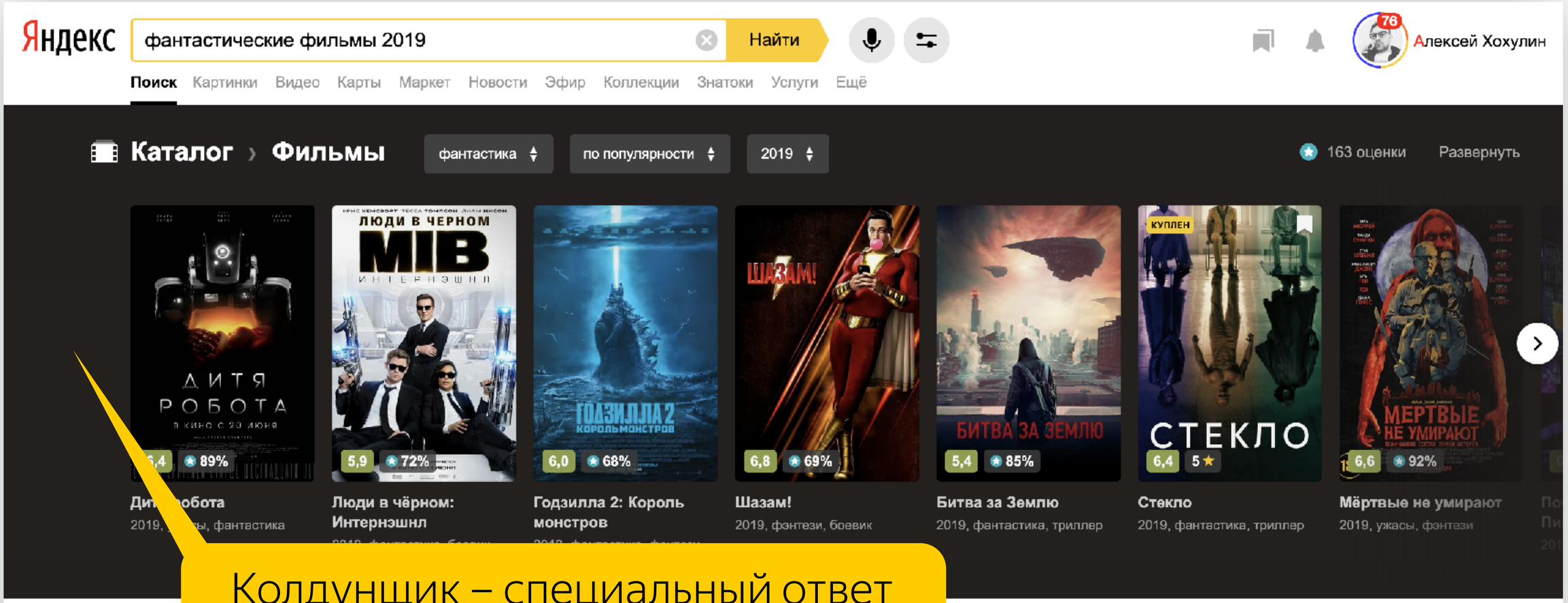
Мёртвые не умирают
2019, ужасы, фэнтези

Больше похожих фильмов

Фантастика 2019 года – списки лучших **фильмов**...
kinopoisk.ru › lists/navigator/sci-fi/2019/
Все страны. **2019**. Навигатор по **фильмам**. **фантастика**. ... **Фантастика 2019** года. Смотреть онлайн. **Фильмы** Сериалы С высоким рейтингом Российские Зарубежные.
Читать ещё ›

Похожие подборки

Большой поиск



Колдунщик – специальный ответ
микро приложение

БЭМ 2010

Методология

Методология

Блок __Элемент _Модификатор

Методология

- Блок __Элемент _Модификатор
- компонентный подход

Методология

- Блок __Элемент _Модификатор
- компонентный подход
- декларативность

Методология

- Блок __Элемент _Модификатор
- компонентный подход
- декларативность
  - композиция

Методология

- Блок __Элемент _Модификатор
- компонентный подход
- декларативность
  - композиция
  - уровни переопределения

Методология

- Блок __Элемент _Модификатор
- компонентный подход
- декларативность
  - композиция
  - уровни переопределения
  - эксперименты

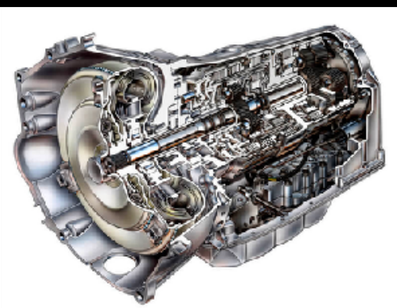
Методология

```
bem in css // styles
i-bem.js   // client logic
bemjson    // blocks / pages
bemhtml    // templates
bemdeps    // dependencies
```

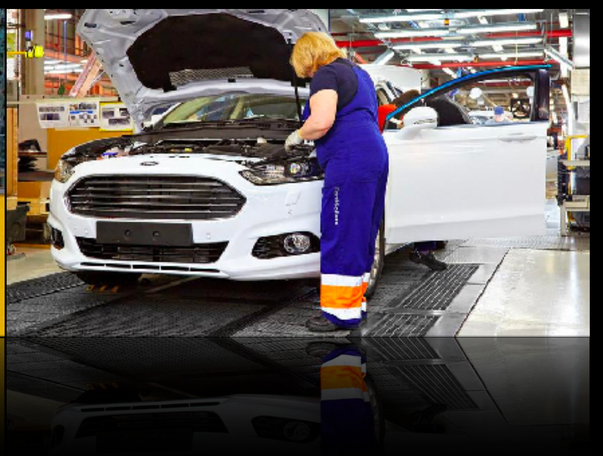
i-bem.js + jquery.js = ❤️

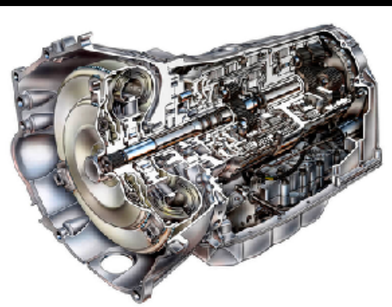
Блоки

Блоки

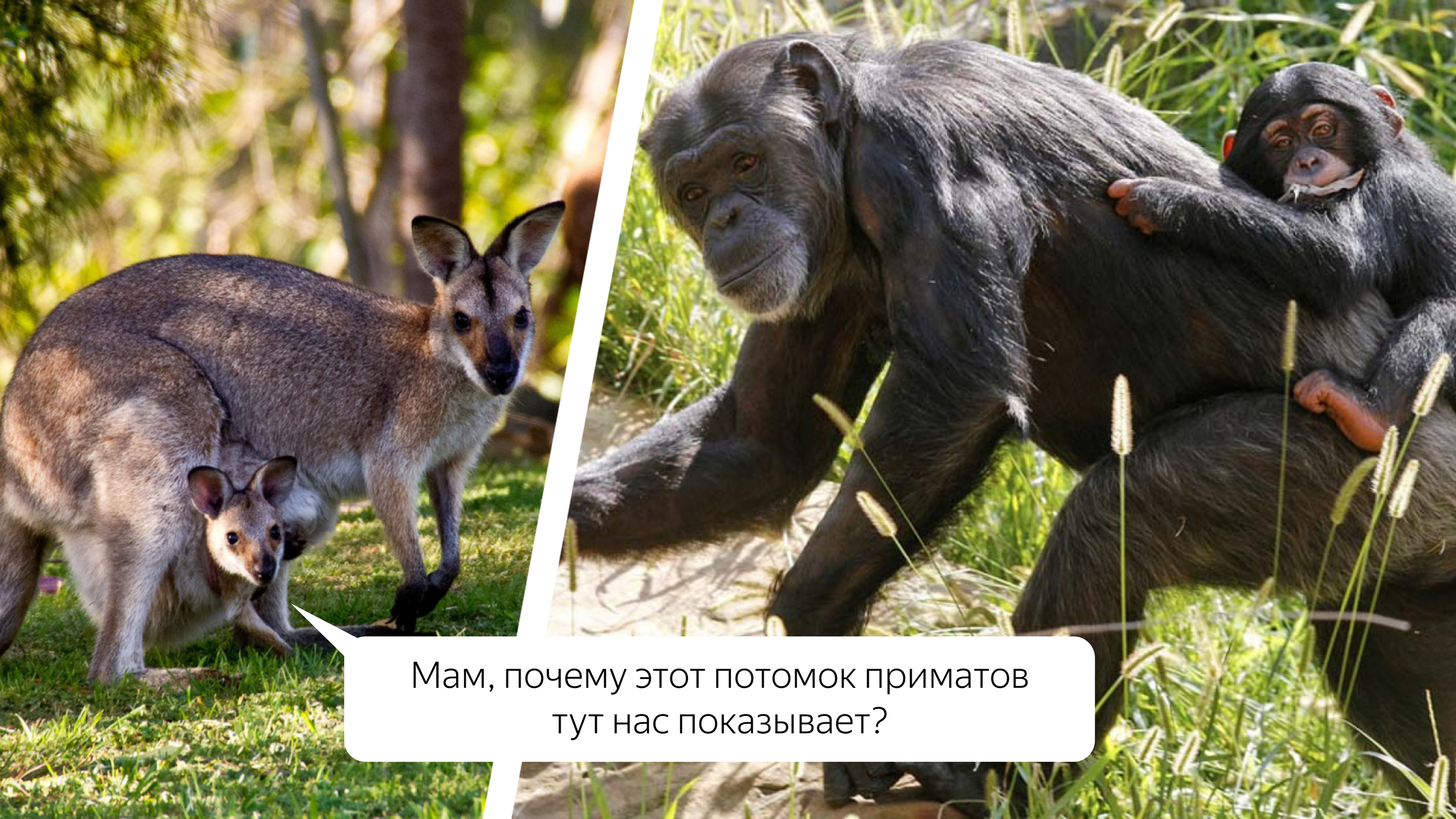


Уровни



Блоки				
Уровни				
Таргеты	Econom	Comfort	Business	Luxury

Эксперименты


Мам, почему этот потомок приматов
тут нас показывает?

Мы с тобой эксперименты, прикинь!


Эксперименты и искусственный отбор!
Мутации и естественный отбор!

$$feature(x) = \sum_{i=1}^{x} experiments_i$$

$$KPI(feature(i+1)) > KPI(feature(i))$$

Experiments

Mutations

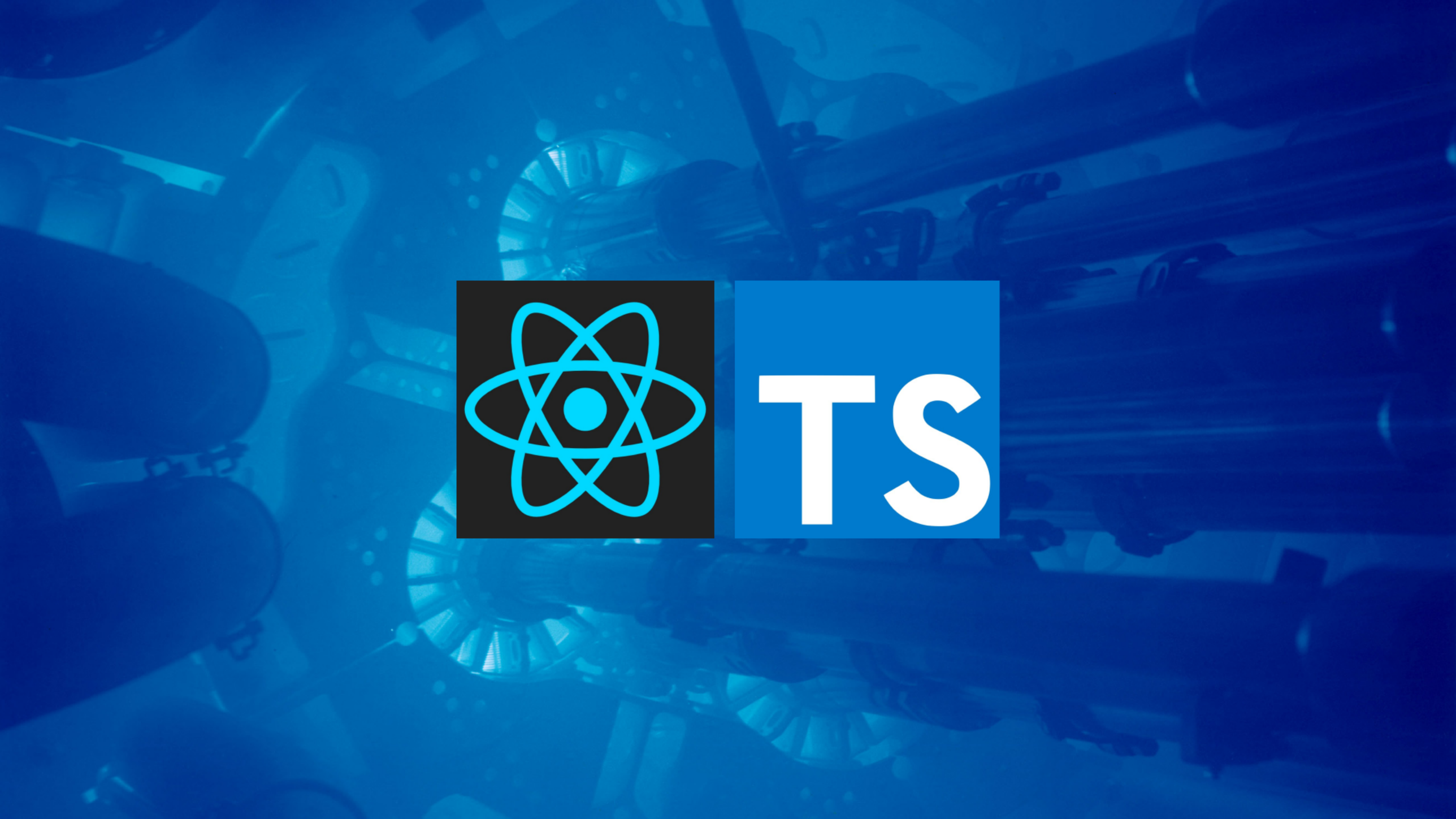
$$feature(x) = \sum_{i=1}^{x} experiments_i$$

$$KPI(feature(i+1)) > KPI(feature(i))$$

Selection

Experiments

БЭМ 2019


TS

react

- всё явно
- всё декларативно

react

typescript

- интерфейсы
- обобщенное программирование
- функциональность
- поддержка IDE

typescript

Methodology

```
import { Methodology } from 'BEM';
```

Methodology

```
import { Methodology } from 'BEM';
```

Methodology

```
import { Methodology } from 'BEM';

class IBemJS extends Methodology<JQuery>;
```

Methodology

```
import { Methodology } from 'BEM';

class IBemJS extends Methodology<JQuery>;
class BemReact extends Methodology<React>;
```

Methodology

```
bem in css // styles
react.js   // client logic
react.js   // blocks / pages
jsx        // templates
imports    // dependencies
```

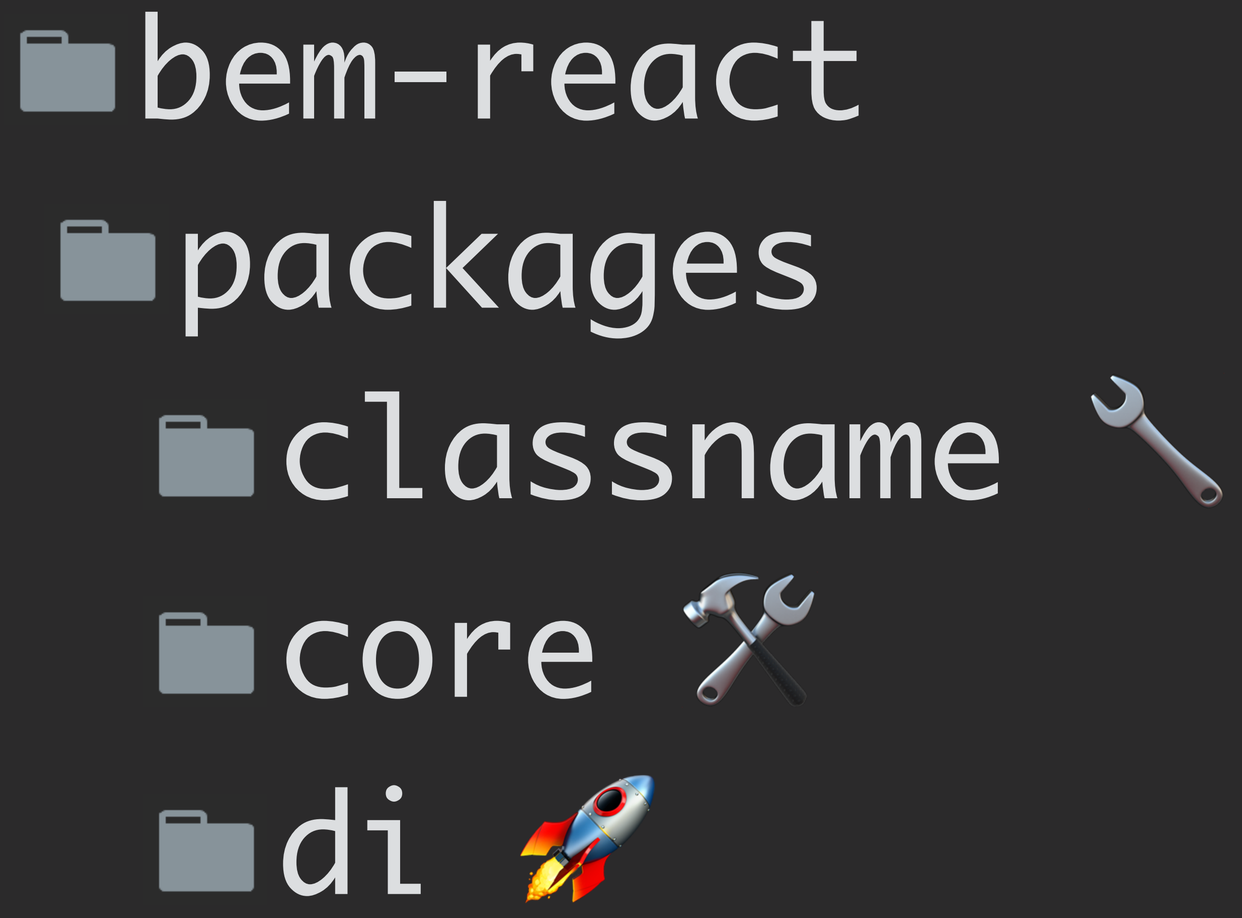
bem-react + react = ❤️

bem-react

- компонентный подход
- изоморфный код
- уровни переопределения
- явные зависимости

bem-react

bem-react



bem-react

classname

```
import { IClassNameProps } from '@bem-react/core'
import { cn } from '@bem-react/classname';

export interface IButtonProps extends IClassNameProps {
  tag?: string;
}

export const cnButton = cn('Button');
cnButton('Text');          // Button Button-Text
cnButton({ size: 'l' }); // Button Button_size_l
```

bem-react

core baseBlock

```tsx
import React, { FC } from 'react';

import { IButtonProps, cnButton } from './index';

export const Button: FC<IButtonProps> = ({
  children,
  className,
  tag: TagName = 'button',
}) => (
  <TagName className={cnButton({}, [className])}>
    {children}
  </TagName>
);
```

bem-react

core modifier

```
import React from 'react';
import { withBemMod } from '@bem-react/core';

import { IButtonProps, cnButton } from '../index';

interface IButtonTypeLinkProps {
  type?: 'link';
}

const withButtonTypeLink = withBemMod<IButtonTypeLinkProps,
IButtonProps>(cnButton(), { type: 'link' }, (Button) => (
  (props) => (
    <Button {...props} tag="a" />
  )
));
```

bem-react

core compose

```
import { Button as ButtonPresenter } from 'Button';
import { withButtonThemeAction, … } from '…';
import { compose, composeU } from '@bem-react/core';

const Button = compose(
  composeU(withButtonThemeAction, withButtonThemeDefault),
  withButtonTypeLink,
)(ButtonPresenter);

export const App: FC = () => (
  <div className="App">
    <Button>Basic</Button>
    <Button theme="action" type="link">All</Button>
  </div>
);
```

bem-react

di primitive App.tsx

```tsx
import { cn } from '@bem-react/classname';
import { Footer } from './Components/Footer';
import { Header } from './Components/Header';

const cnApp = cn('App');

export const App = () => {
    return (
        <div className={cnApp()}>
            <Header />
            <Footer />
        <div/>
    );
};
```

bem-react

di by hooks App.tsx

```tsx
import { cn } from '@bem-react/classname';
import { useComponentRegistry } from '@bem-react/di';

const cnApp = cn('App');

export const App = () => {
    const { Header, Footer } = useComponentRegistry(cnApp());

    return (
        <>
            <Header />
            <Footer />
        </>
    );
};
```

bem-react

di injection App@desktop.tsx

```
import { cn } from '@bem-react/classname';
import { Registry, withRegistry } from '@bem-react/di';
import { App as AppCommon, registryId } from './App';
import { Header } from ‘./Components/Header/Header@desktop';
import { Footer } from './Components/Footer/Footer@desktop';

export const cnApp = cn('App');
export const registryId = cnApp();
export const registry = new Registry({ id: registryId });

registry.set('Header', Header); registry.set('Footer', Footer);

export const AppDesktop = withRegistry(registry)(AppCommon);
```

bem-react

di experiment

```
import { AppDesktop, registryId } from './App@desktop';
import { HeaderExp } from './exp/Components/Header/Header';

const expRegistry = new Registry({ id: registryId });

expRegistry.set('Header', HeaderExp);

export const AppDesktopExp = withRegistry(
    expRegistry
)(AppDesktop);
```

bem-react

https://github.com/bem/bem-react/

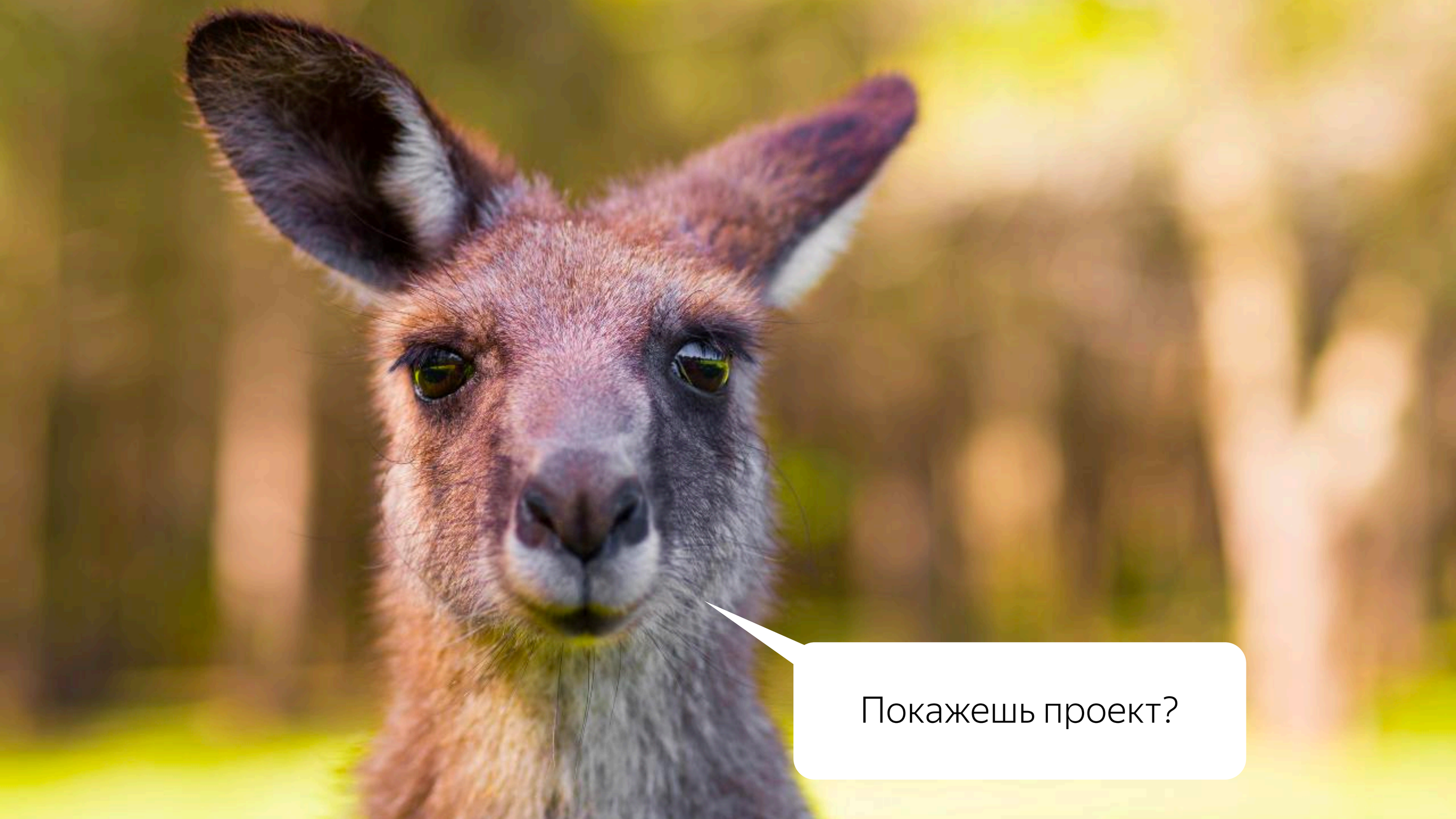
Package	Version	Size
@bem-react/classname	npm v1.5.3	minzipped size 532 B
@bem-react/classnames	npm v1.3.4	minzipped size 316 B
@bem-react/core	npm v2.0.2	minzipped size 638 B
@bem-react/di	npm v2.0.2	minzipped size 743 B

bem-react

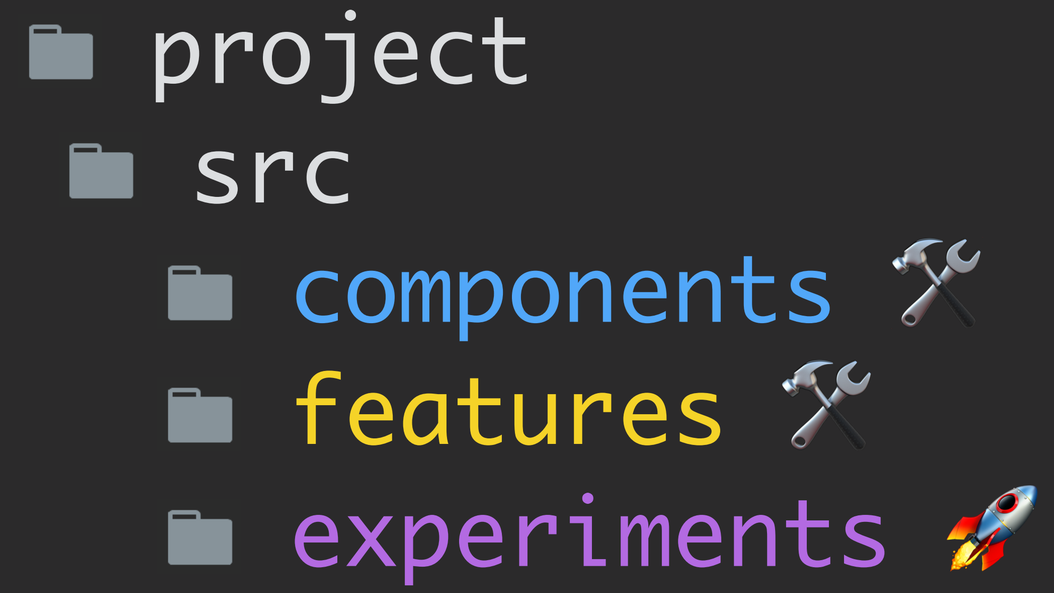
Возможности

- малая инвазивность
- автономность модулей
- удобная сборка

Возможности

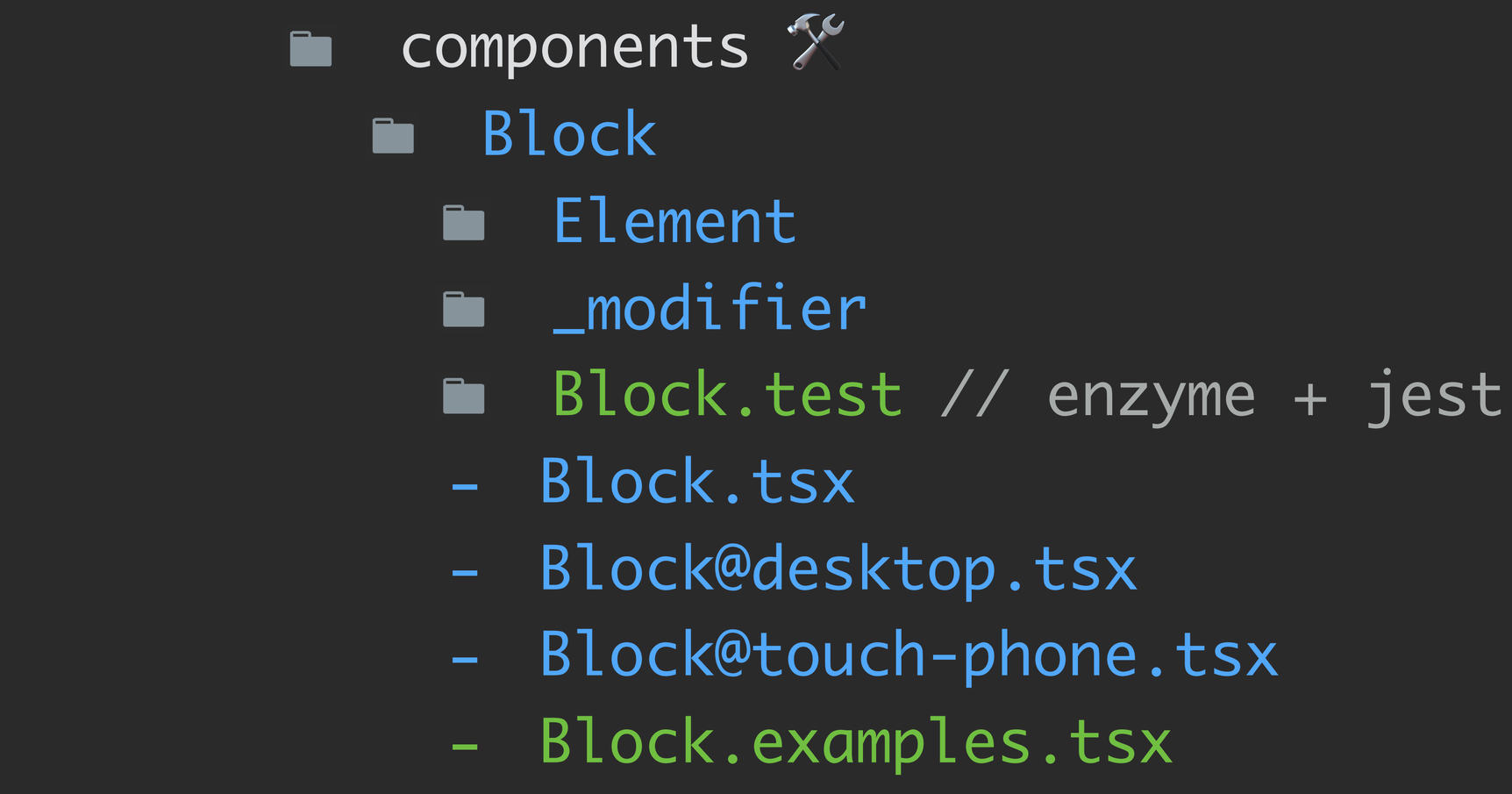

Покажешь проект?

Проект



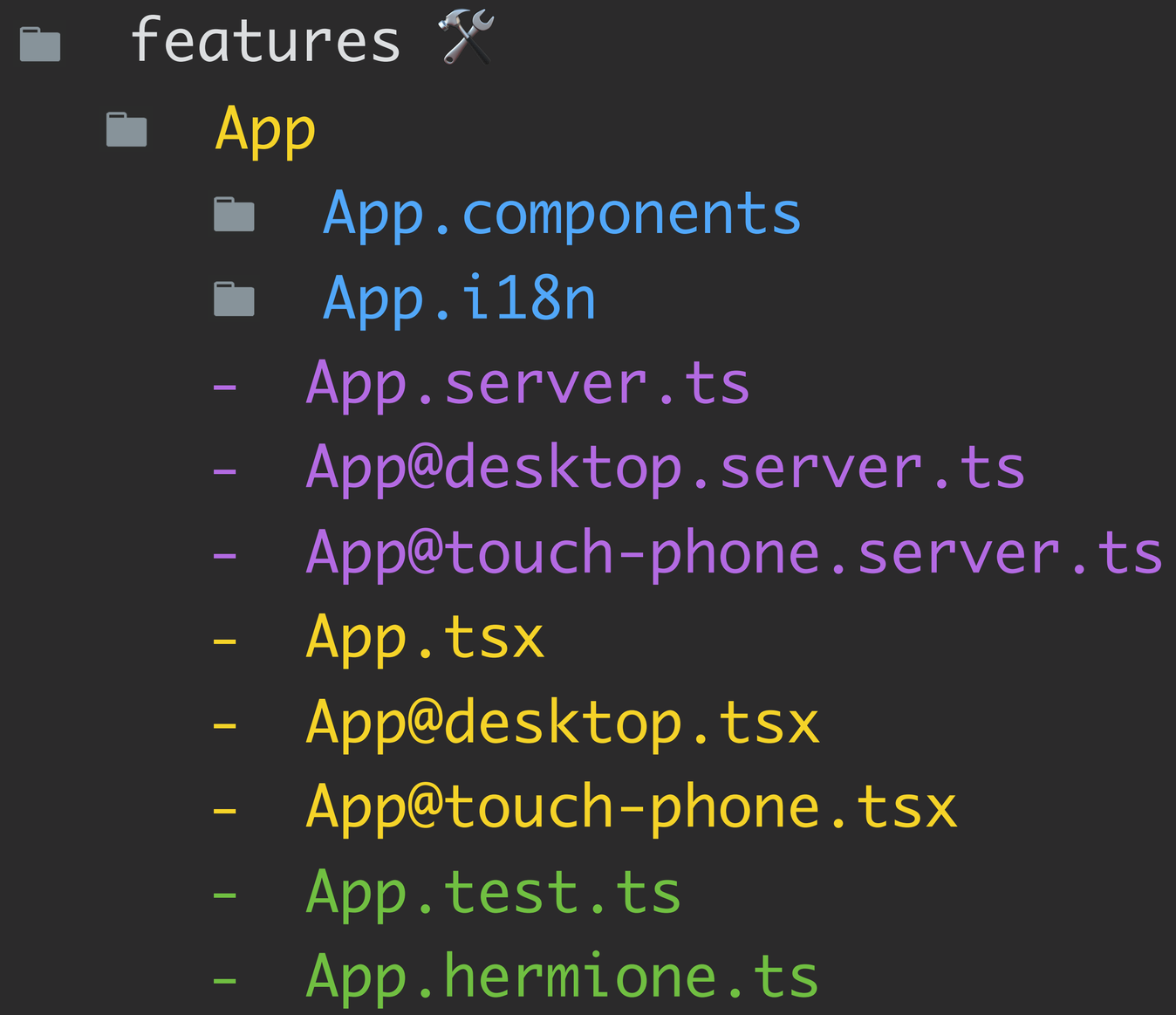
Проект

Общие компоненты



Общие компоненты

Адаптеры и приложения



Адаптеры и приложения

- features 🛠️
  - App
    - App.components
    - App.i18n
    - App.server.ts
    - App@desktop.server.ts
    - App@touch-phone.server.ts
    - App.tsx
    - App@desktop.tsx
    - App@touch-phone.tsx
    - App.test.ts
    - App.hermione.ts

```
Adapters<RawData>: AppProps
```

Адаптеры и приложения

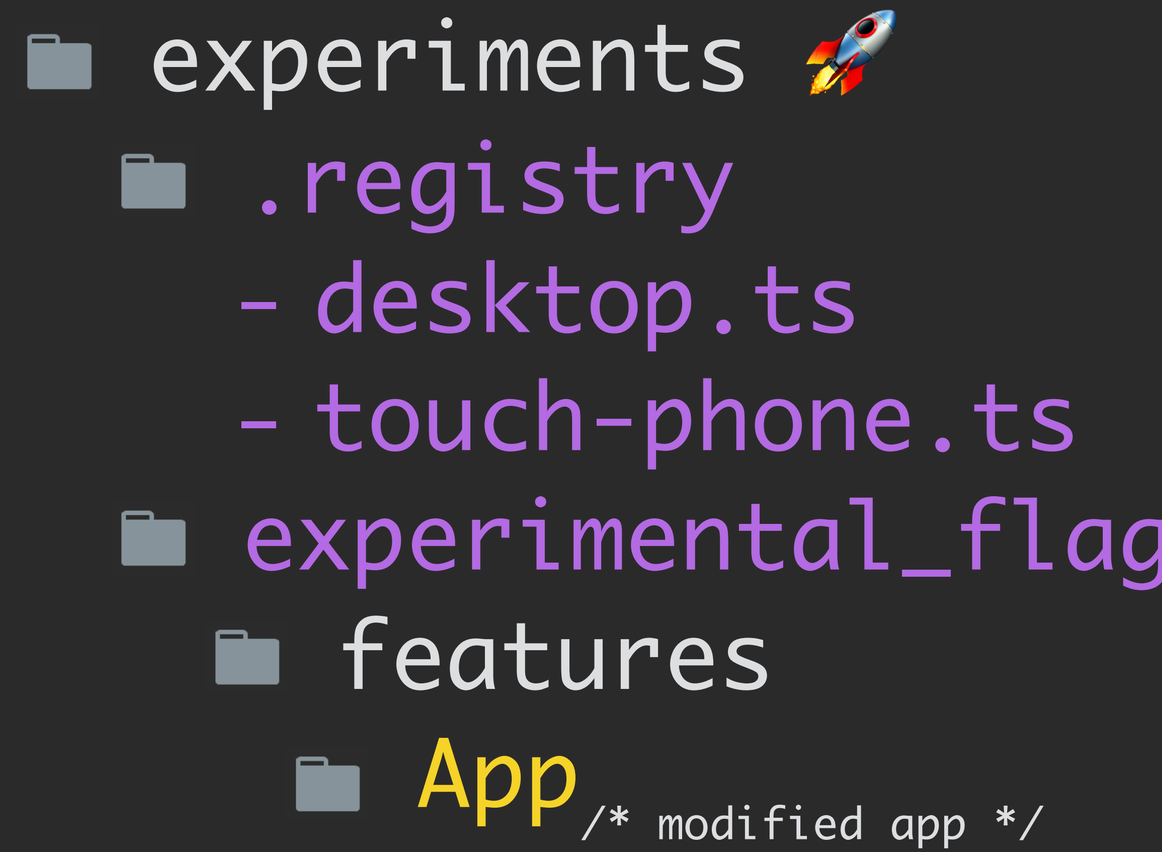
hermione.js

hermione.js

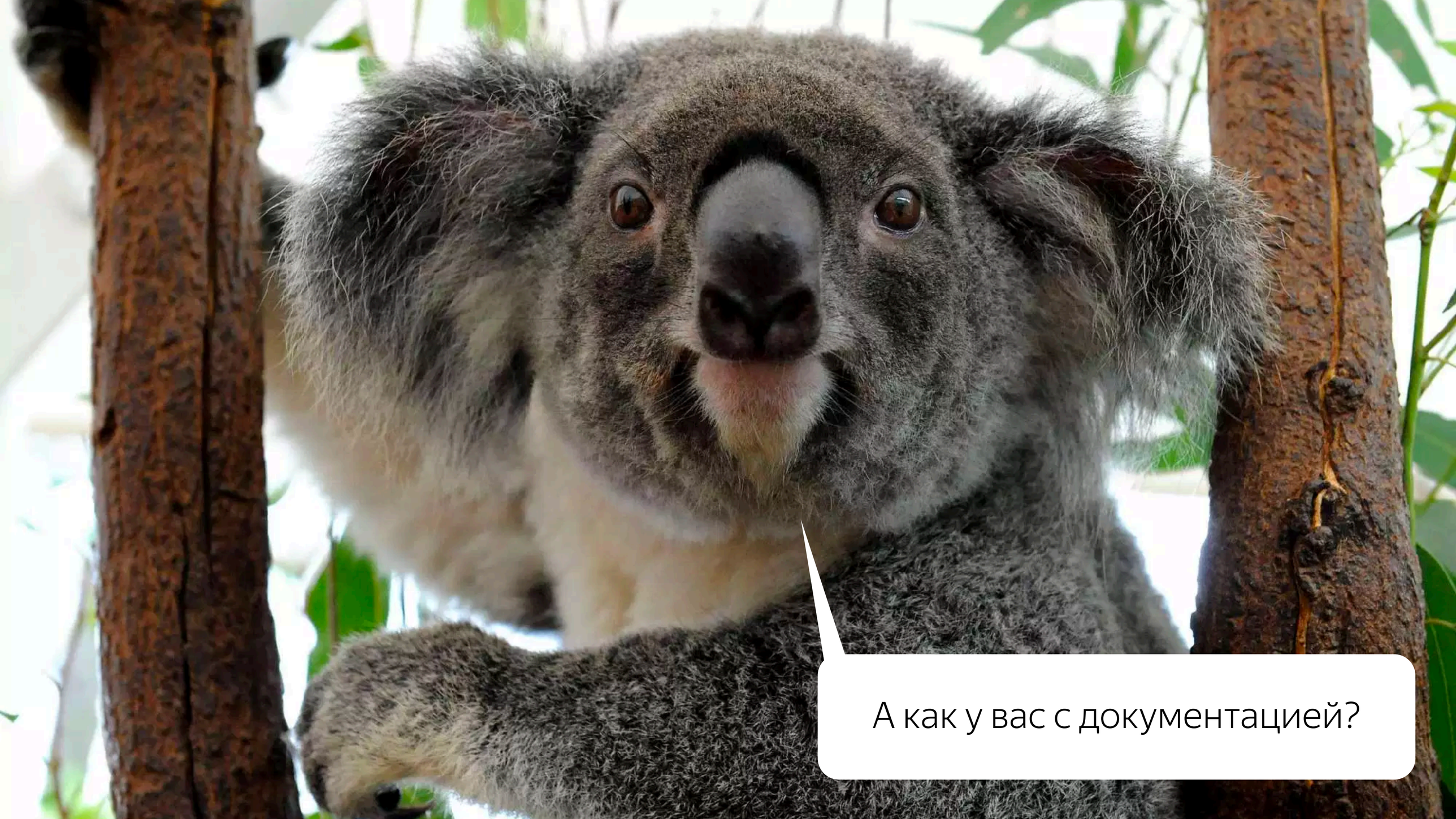
Skip Ad

Protego! Интеграционные тесты!

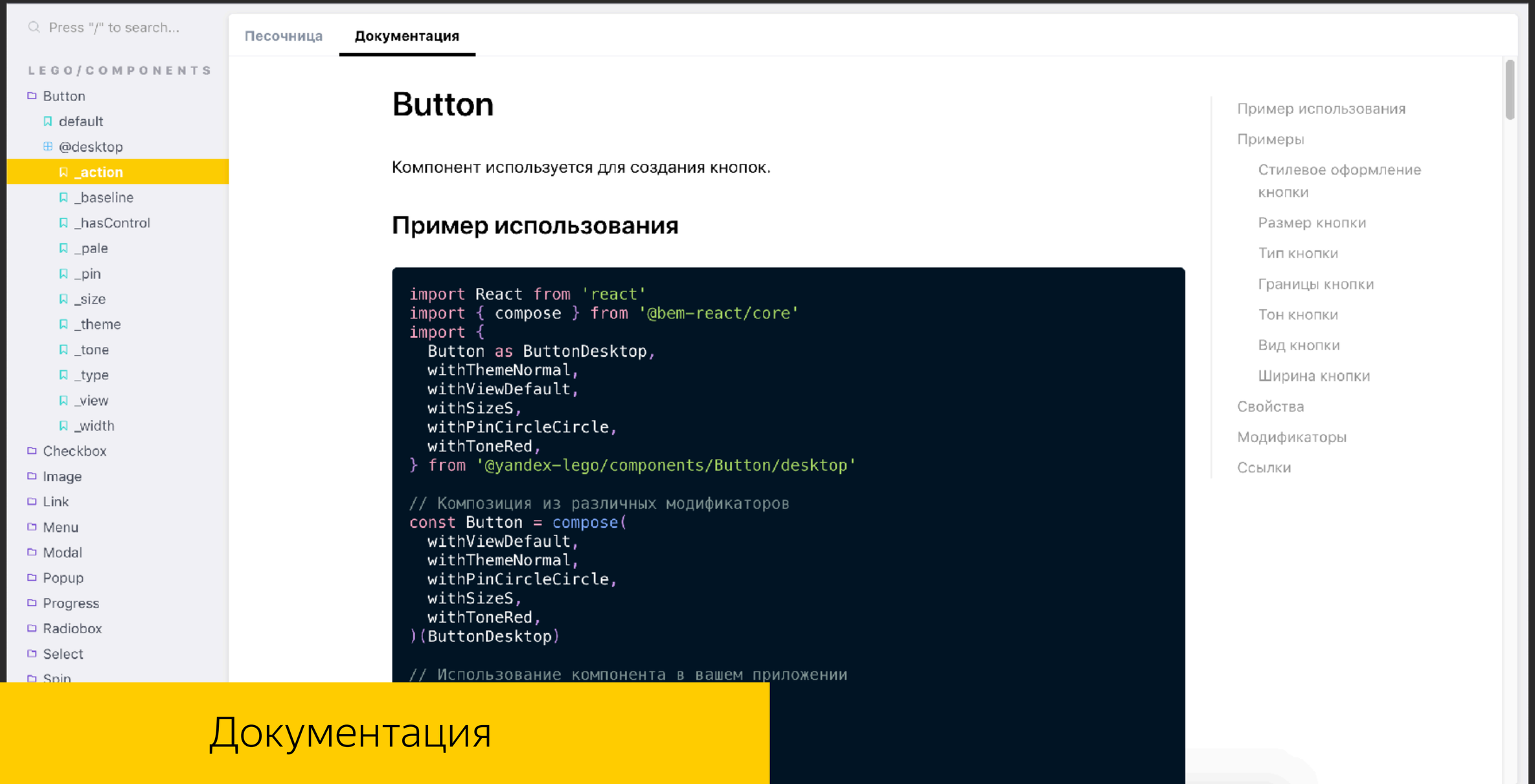
Эксперименты



Эксперименты


А как у вас с документацией?

storybook 🚀



Документация

Summary

React + TS

Методология

Эксперименты

Скорость,
качество

Промышленный
инструмент

Open source

Хохулин Алексей

руководитель группы
разработки интерфейсов

 @xoxulin

Спасибо! Вопросы?

Хохулин Алексей

руководитель группы
разработки интерфейсов

@xoxulin