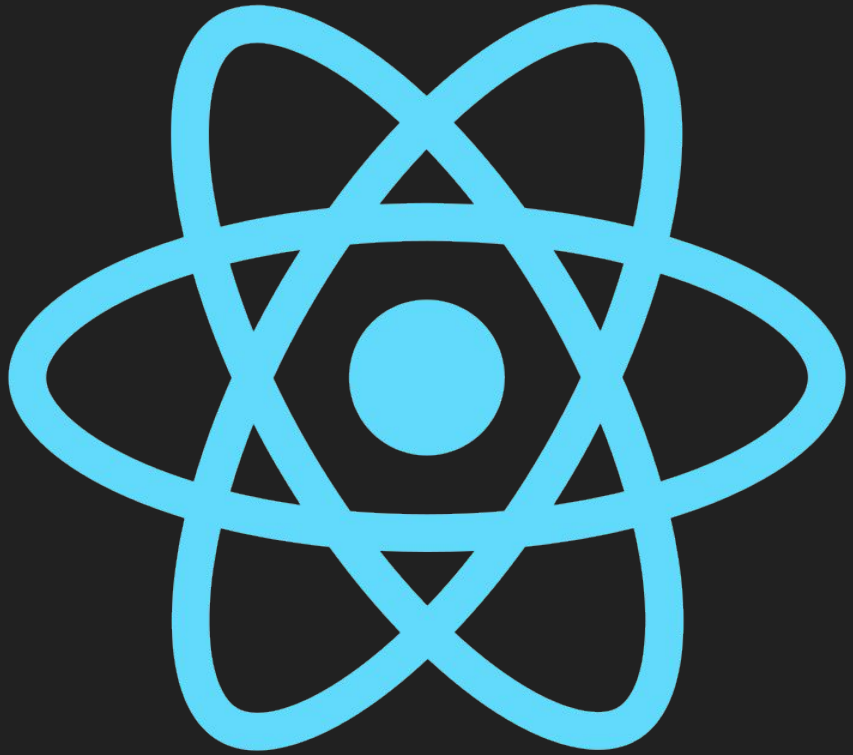
Котерняк Сергей

Front-end dev. at **Heartpace**

Стили в React / CSS-Modules

Мы поговорим о:

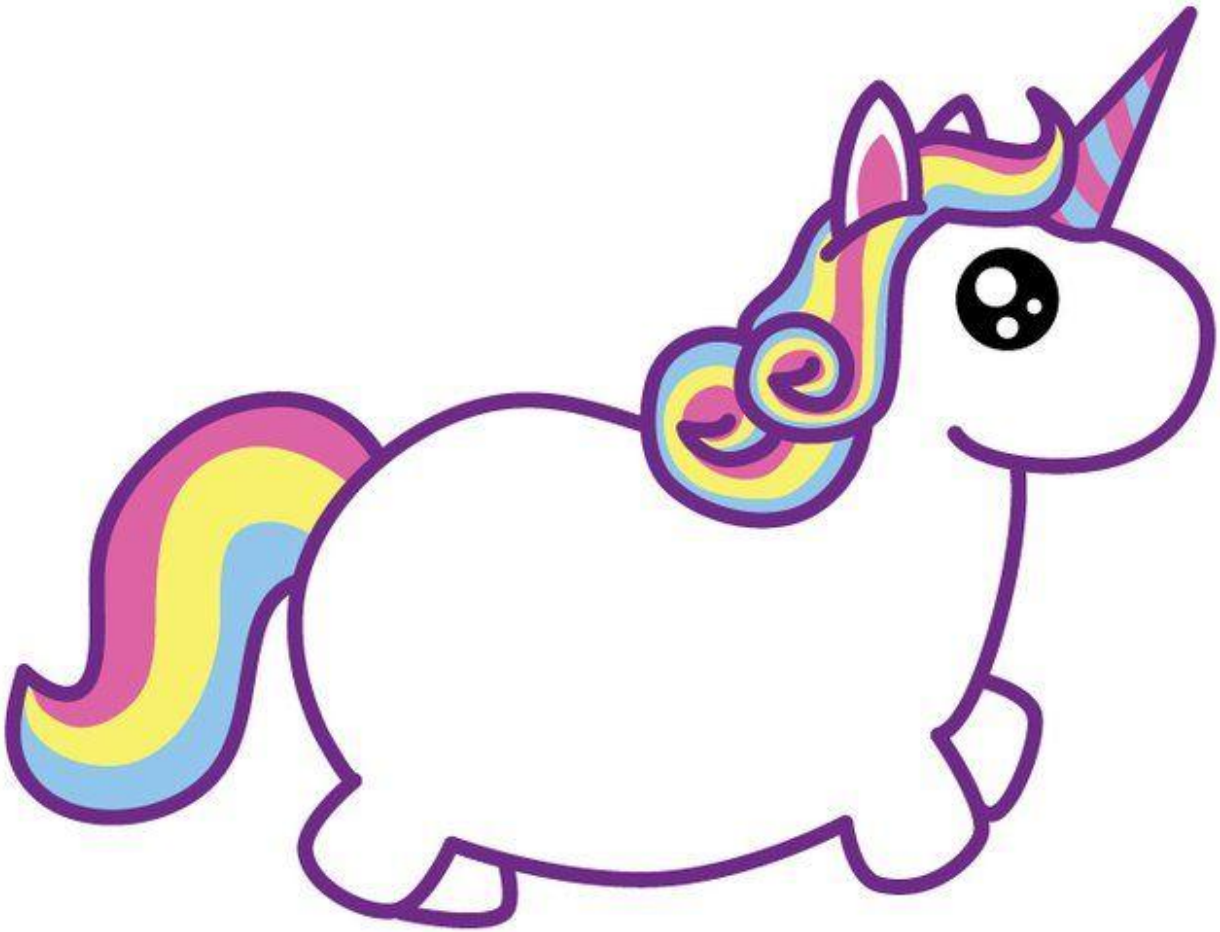
1. Важности выбора подхода по написанию стилей
2. Самых популярных подходах и решениях
3. Css-modules. Почему они и как это работает



Стили - это вообще важно?

1. Архитектура приложения
2. Выбор инструментов для разработки
3. Удобство разработки
4. Скорость решения задач

Самые популярные подходы и решения



CSS

- доступны все нативные свойства
- легко использовать пре/постпроцессоры
- удобно подключать autoprefixer
- легко начать пользоваться

- противоречит компонентному подходу
- глобальная область видимости
- проблемы с названием классов


.title {
font-weight: bold;
font-size: 16px;
}
.email {
padding: .5rem;
}
.submitButton {
padding: .5rem;
margin-top: .5rem;
border: 1px solid #2F79AD;
border-radius: 4px;
background-color: #6DB9EE;
}
.submitButton:hover {
background-color: #2F79AD;
}

BEM

- структурированное наименования элементов
- доступны все плюсы CSS
- низкий порог входа новых или неопытных сотрудников (для стилей)

- глобальная область видимости
- монструозные классы

```
<div class="card">
    <div class="card__header">
        <h2 class="card__title">Title text here</h3>
    </div>

    <div class="card__body">

        <img class="card__img" src="http://some-img.png">
        <p class="card__text">Lorem ipsum dolor sit amet, consectetur</p>
        <p class="card__text">Adipiscing elit. Pellentesque.</p>

        <button class="button button--primary">Click me!</button>

    </div>
</div>
```

```css
.card {
    border: 1px solid red;
}

.card__header {
    padding: 0 15px;
}

.card__title {
    text-transform: uppercase;
    font-size: 20px;
}

.button {
    background: blue;
    color: white;
}

.button--primary {
    background: grey;
}
```

Inline Styles

- независимые компоненты
- локальная область видимости
- поддержка Facebook?

- отсутствие нормальной возможности работать с псевдоклассами, псевдоэлементами и медиазапросами
- проблемы с autoprefixer
- невозможность переопределять стили по более сложным селекторам
- возможность влиять на глобальные стили

```
var divStyle = {
  color: 'white',
  backgroundImage: 'url(' + imgUrl + ')',
  WebkitTransition: 'all',
  msTransition: 'all'
};

ReactDOM.render(<div style={divStyle}>Hello World!</div>, mountNode);
```

CSS in JS

- независимые компоненты
- локальная область видимости
- нет ограничений по использованию стилей

- высокий порог входа
- назад в будущее (все в одном файле)
- очень много решений
- веб-стандарты более надежны

Radium

```
var Radium = require('radium');
var React = require('react');
var color = require('color');

class Button extends React.Component {
  static propTypes = {
    kind: PropTypes.oneOf(['primary', 'warning']).isRequired
  };

  render() {
    return (
      <button
        style={[
          styles.base,
          styles[this.props.kind]
        ]}>
        {this.props.children}
      </button>
    );
  }
}

Button = Radium(Button);
```

```
var styles = {
  base: {
    color: '#fff',
    ':hover': {
      background: color('#0074d9').lighten(0.2).hexString()
    }
  },

  primary: {
    background: '#0074D9'
  },

  warning: {
    background: '#FF4136'
  }
};
```

Aphrodite

```
import React, { Component } from 'react';
import { StyleSheet, css } from 'aphrodite';

class App extends Component {
    render() {
        return <div>
            <span className={css(styles.red)}>
                This is red.
            </span>
            <span className={css(styles.hover)}>
                This turns red on hover.
            </span>
            <span className={css(styles.small)}>
                This turns red when the browser is less than 600px width.
            </span>
            <span className={css(styles.red, styles.blue)}>
                This is blue.
            </span>
            <span className={css(styles.blue, styles.small)}>
                This is blue and turns red when the browser is less than
                600px width.
            </span>
        </div>;
    }
}
```

```
const styles = StyleSheet.create({
    red: {
        backgroundColor: 'red'
    },

    blue: {
        backgroundColor: 'blue'
    },

    hover: {
        ':hover': {
            backgroundColor: 'red'
        }
    },

    small: {
        '@media (max-width: 600px)': {
            backgroundColor: 'red',
        }
    }
});
```

CSS Modules

CSS Modules - что ты такое?(c)

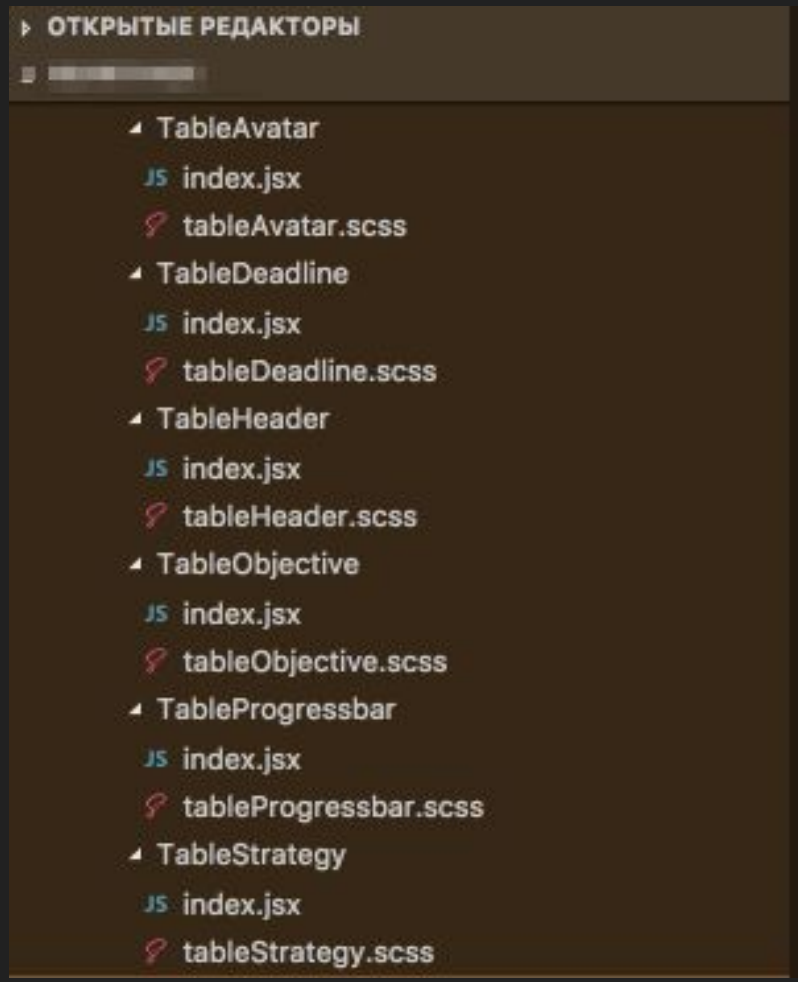
Кратко это звучит так: css-файлы, в которых все классы и анимации по умолчанию находятся в локальной области видимости.

Локальная и глобальная область видимости

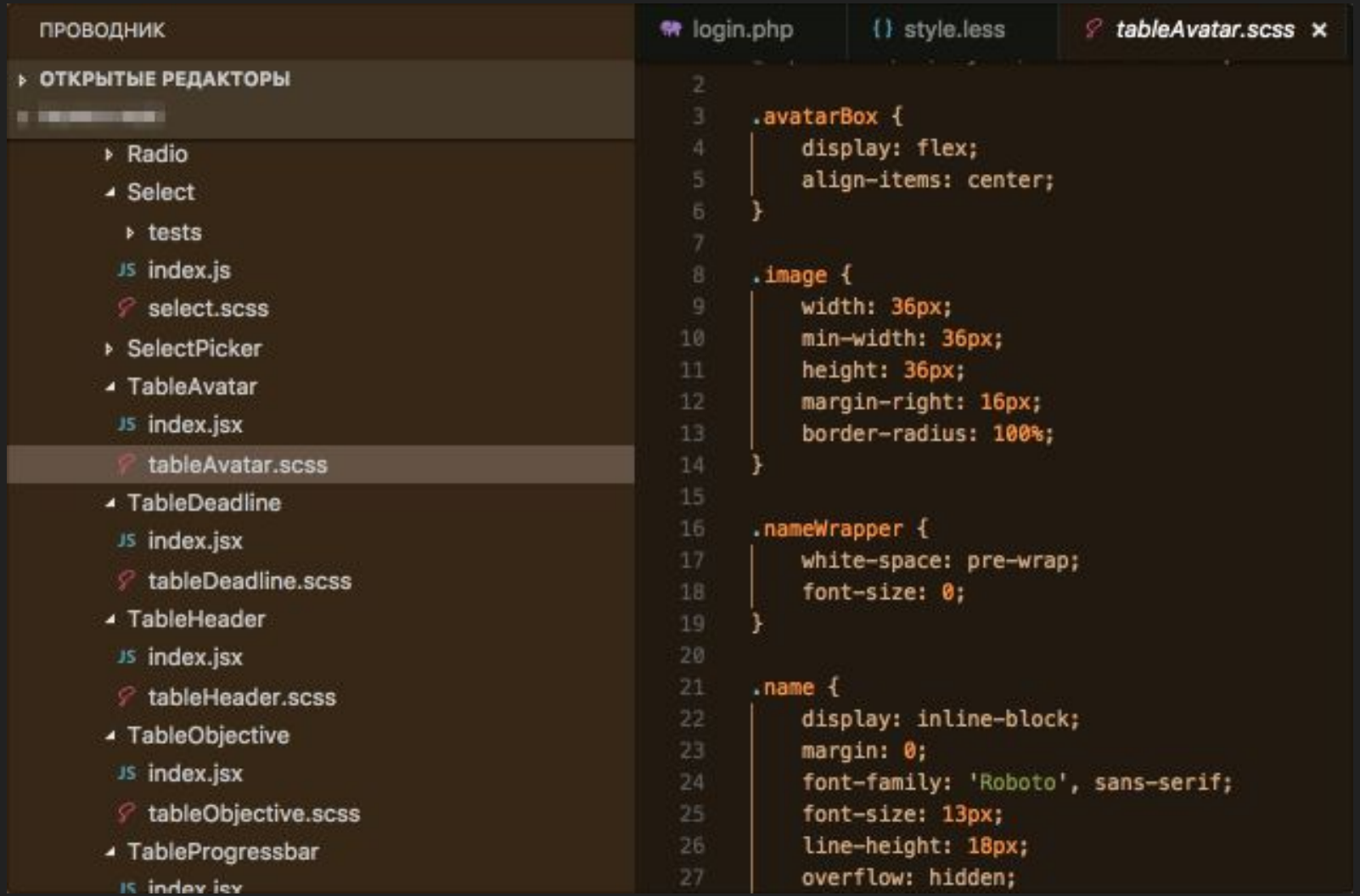

.title {
font-weight: bold;
font-size: 16px;
}
.email {
padding: .5rem;
}
.submitButton {
padding: .5rem;
margin-top: .5rem;
border: 1px solid #2F79AD;
border-radius: 4px;
background-color: #6DB9EE;
}
.submitButton:hover {
background-color: #2F79AD;
}

```scss
@import '../../styles/variables.scss';

:global {
    .react-grid-HeaderCell {
        background: $color-white;
        border: none;
        font-size: 13px;
        padding: 30px 6px 12px;

        &:first-child {
            padding-left: 20px;
        }

        &:last-child {
            padding-right: 20px;
        }
    }
    .react-grid-Row {
        overflow: visible !important;
        position: relative;
    }

    .react-grid-HeaderCell-sortable {
        &:after {
            content: '';
            display: inline-block;
            width: 0;
```

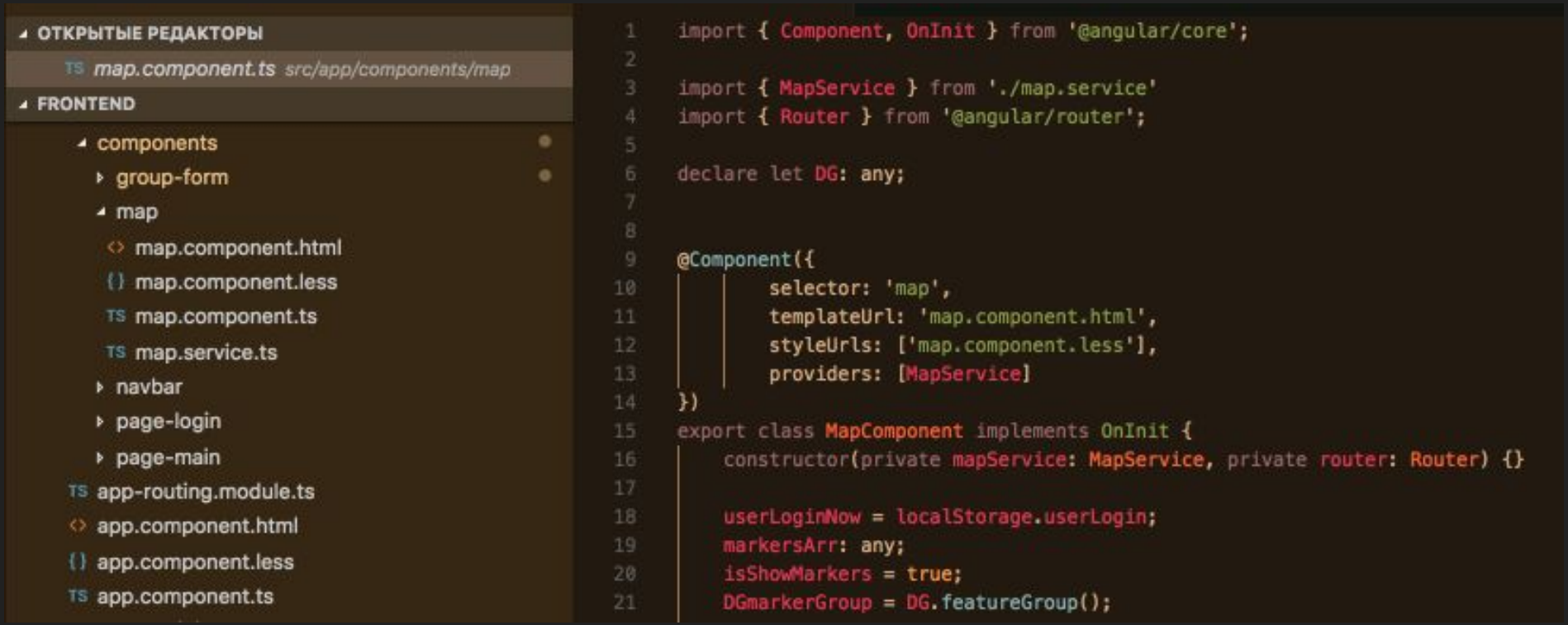
Удобная структура для написания компонентов

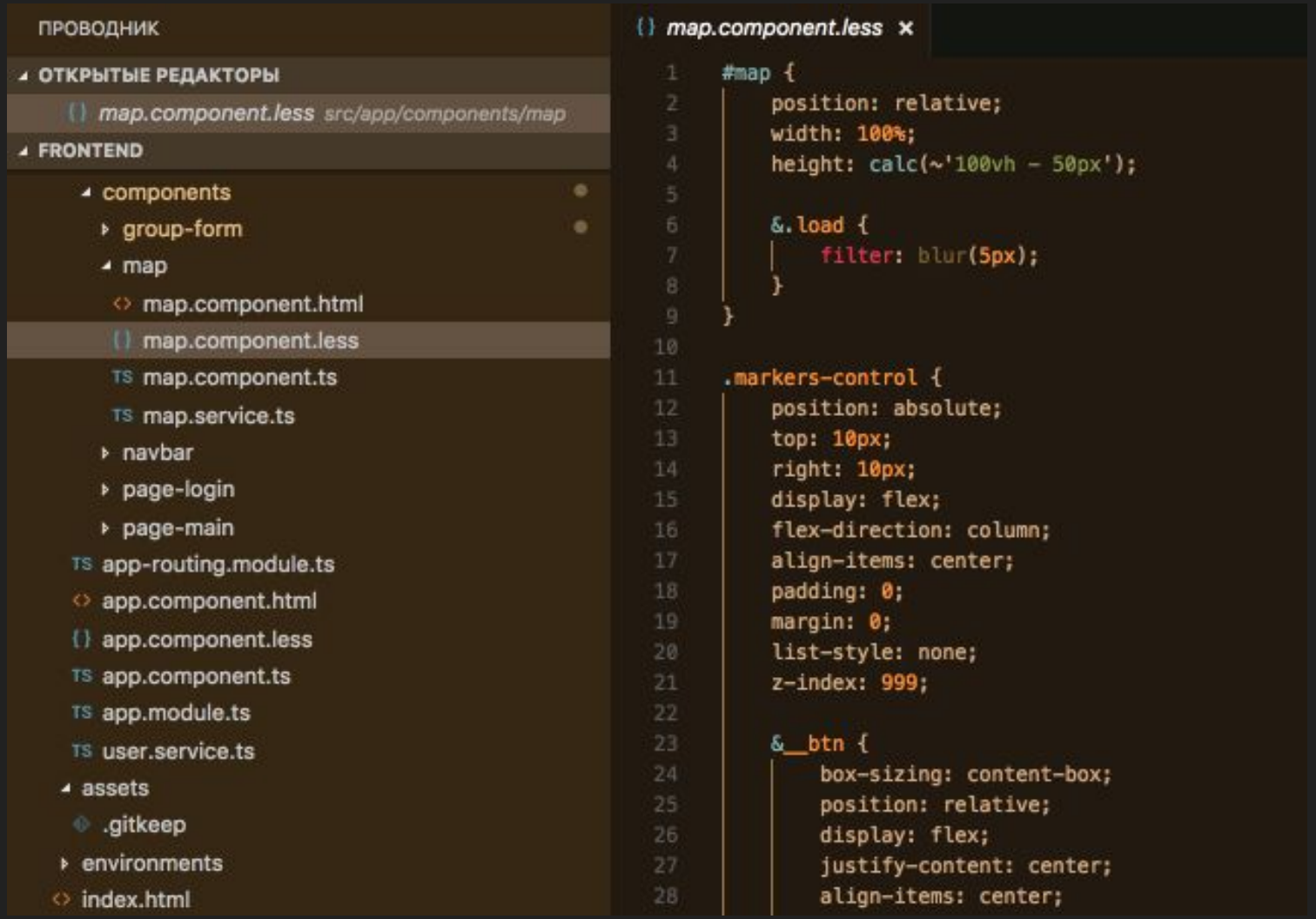



```jsx
import PropTypes from 'prop-types';
import React from 'react';
import styles from './tableAvatar.scss';

class tableAvatarBlock extends React.Component {

    constructor(props) {
        super(props);
    }

    render() {
        const props = this.props.value;
        if (!props.name) {
            return null;
        }

        return (
            <div className={styles.avatarBox}>
                <img src={props.avatar}
                     alt={props.altText}
                     className={styles.image}/>

                <p className={styles.name}>{props.name}</p>
            </div>
        );
```


ПРОВОДНИК
ОТКРЫТЫЕ РЕДАКТОРЫ
Radio
Select
tests
index.js
select.scss
SelectPicker
TableAvatar
index.jsx
tableAvatar.scss
TableDeadline
index.jsx
tableDeadline.scss
TableHeader
index.jsx
tableHeader.scss
TableObjective
index.jsx
tableObjective.scss
TableProgressbar
login.php
style.less
tableAvatar.scss
.avatarBox {
display: flex;
align-items: center;
}
.image {
width: 36px;
min-width: 36px;
height: 36px;
margin-right: 16px;
border-radius: 100%;
}
.nameWrapper {
white-space: pre-wrap;
font-size: 0;
}
.name {
display: inline-block;
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 13px;
line-height: 18px;
overflow: hidden;

Это также просто, как если бы мы писали на Angular

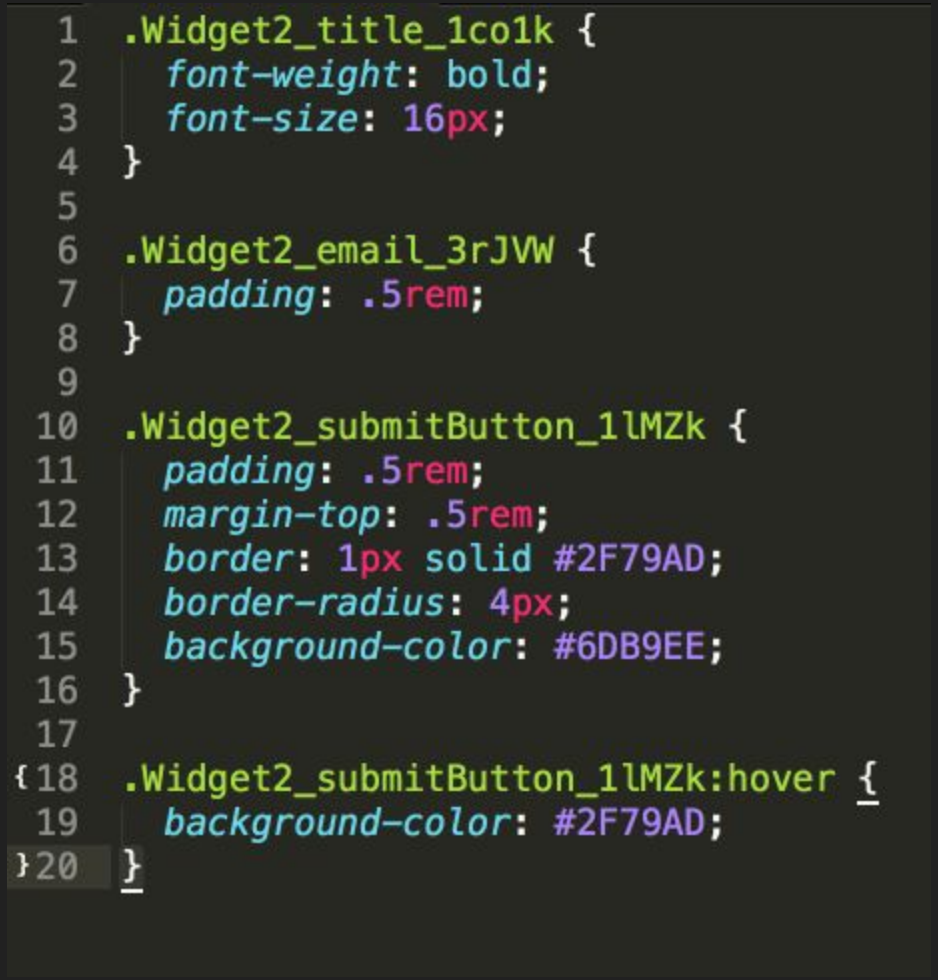

ОТКРЫТЫЕ РЕДАКТОРЫ
map.component.ts src/app/components/map
FRONTEND
components
group-form
map
map.component.html
map.component.less
map.component.ts
map.service.ts
navbar
page-login
page-main
app-routing.module.ts
app.component.html
app.component.less
app.component.ts
import { Component, OnInit } from '@angular/core';
import { MapService } from './map.service'
import { Router } from '@angular/router';
declare let DG: any;
@Component({
selector: 'map',
templateUrl: 'map.component.html',
styleUrls: ['map.component.less'],
providers: [MapService]
})
export class MapComponent implements OnInit {
constructor(private mapService: MapService, private router: Router) {}
userLoginNow = localStorage.userLogin;
markersArr: any;
isShowMarkers = true;
DGmarkerGroup = DG.featureGroup();


ПРОВОДНИК
ОТКРЫТЫЕ РЕДАКТОРЫ
map.component.less src/app/components/map
FRONTEND
components
group-form
map
map.component.html
map.component.less
map.component.ts
map.service.ts
navbar
page-login
page-main
app-routing.module.ts
app.component.html
app.component.less
app.component.ts
app.module.ts
user.service.ts
assets
.gitkeep
environments
index.html
map.component.less
#map {
position: relative;
width: 100%;
height: calc(~'100vh - 50px');
&.load {
filter: blur(5px);
}
}
.markers-control {
position: absolute;
top: 10px;
right: 10px;
display: flex;
flex-direction: column;
align-items: center;
padding: 0;
margin: 0;
list-style: none;
z-index: 999;
&__btn {
box-sizing: content-box;
position: relative;
display: flex;
justify-content: center;
align-items: center;

Возможность использовать пре/постпроцессоров, автопрефиксера, анимаций, псевдо*


THE
END
IS
NEAR


.title {
font-weight: bold;
font-size: 16px;
}
.email {
padding: .5rem;
}
.submitButton {
padding: .5rem;
margin-top: .5rem;
border: 1px solid #2F79AD;
border-radius: 4px;
background-color: #6DB9EE;
}
.submitButton:hover {
background-color: #2F79AD;
}



```
.Widget2_title_1co1k {
  font-weight: bold;
  font-size: 16px;
}

.Widget2_email_3rJVW {
  padding: .5rem;
}

.Widget2_submitButton_1lMZk {
  padding: .5rem;
  margin-top: .5rem;
  border: 1px solid #2F79AD;
  border-radius: 4px;
  background-color: #6DB9EE;
}

.Widget2_submitButton_1lMZk:hover {
  background-color: #2F79AD;
}
```

Спасибо