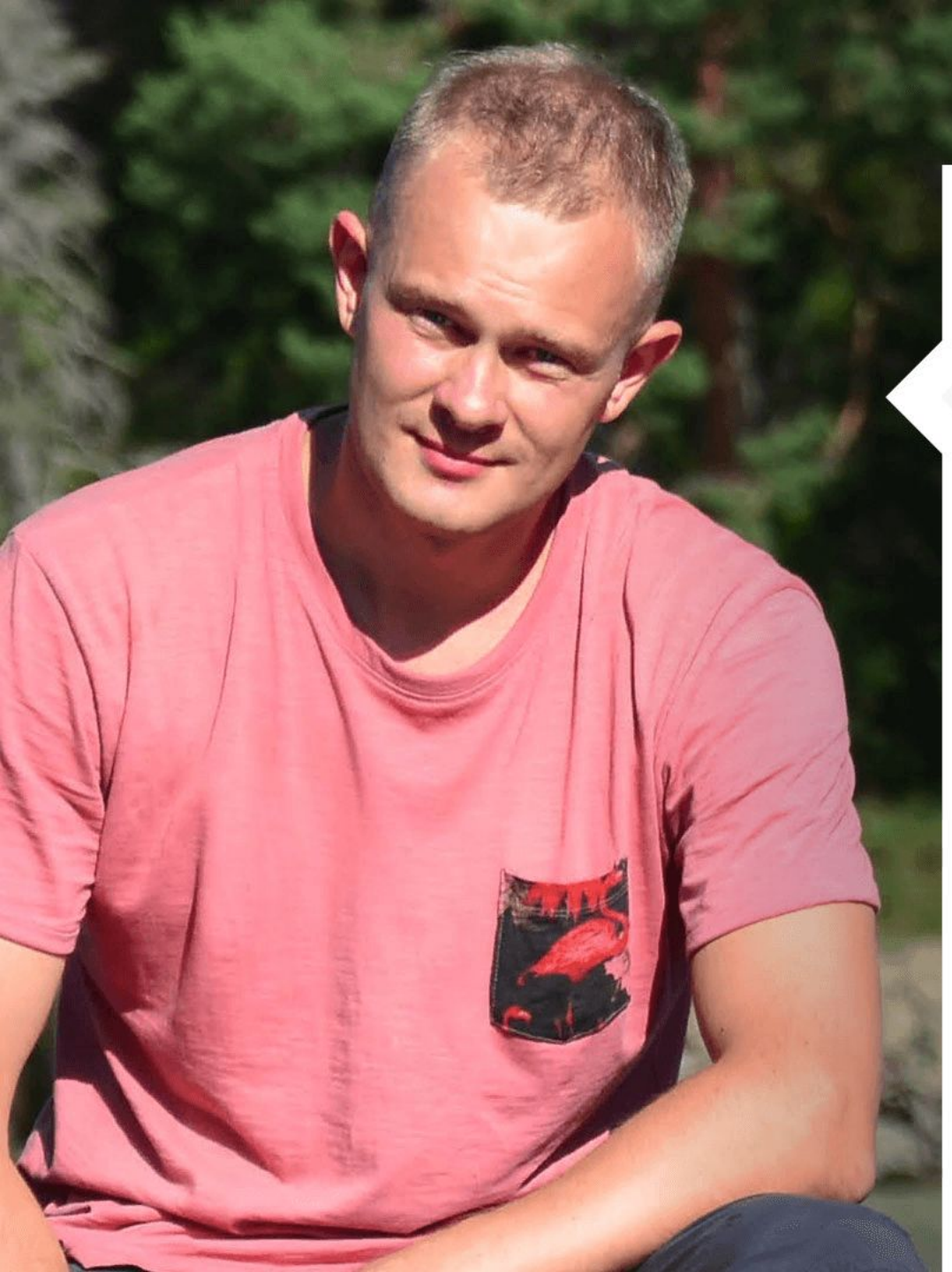


SVELTE. Легкий — не значит быстрый

FRONTEND

Евгений Пономарев
руководитель группы
фронтенд-разработки

ВЕЩАЕТ

Евгений Пономарев

Тимлид команды фронтенда.
Разрабатываю сервисы, продающие мобильный контент.
Компетенции: вёрстка, JS, управление разработкой.

СВЯЗАТЬСЯ СО МНОЙ

@e.ponomarev

evgen_ponomarev

evgenij.ponomarev1

evgeniy1801


Meteor
Spine
Batman
Mithril
Svelte
Knockout
Reactive
CanJS
Aurelia
Backbone
Vue
React
Angular
jQuery
Polymer
Ember

Зачем

1. Ценный опыт и развитие.
2. Видение ситуации в индустрии.
3. Разбавить рутину.
4. Плюс один в список скилов.

Контекст восприятия

- Восторг от первого опыта написания реактивных приложений.
- Простой для разработки простых приложений.
- Модный.

Контекст восприятия

- Сложная отладка.
- Двустороннее связывание данных с представлением по умолчанию.
- Не самый быстрый.
- Не такой уж простой для разработки чего-то серьёзного.
- Module, Component, Directive, Controller, Factory, Service, Provider...😱

Опрос 23 000 разработчиков в 2017 г.

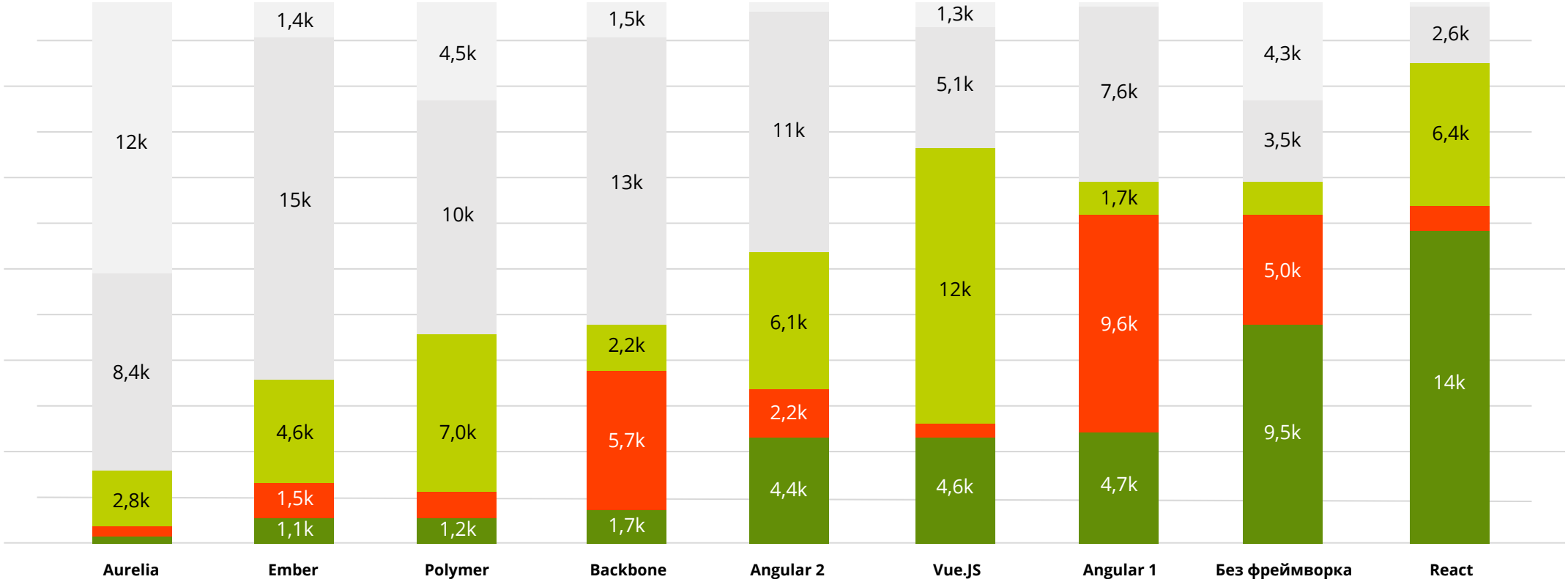
Я **никогда** не слышал об этом

Я слышал об этом, но **не заинтересован**

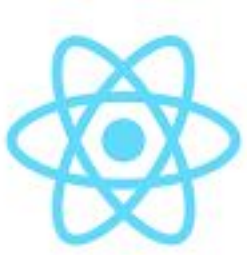
Я слышал и **хотел бы** освоить

Я использовал раньше, но больше **не буду** это использовать

Я использовал раньше и **буду** продолжать использовать



Контекст восприятия



- JSX — расширение JS, а не отдельный язык шаблонов.
- Методы жизненного цикла улучшают контроль и понимание.
- Только отрисовка страницы, ничего лишнего.

Кто и когда

SVELTE

Рич Харрис

Rich_Harris

Первая версия — ноябрь 2016

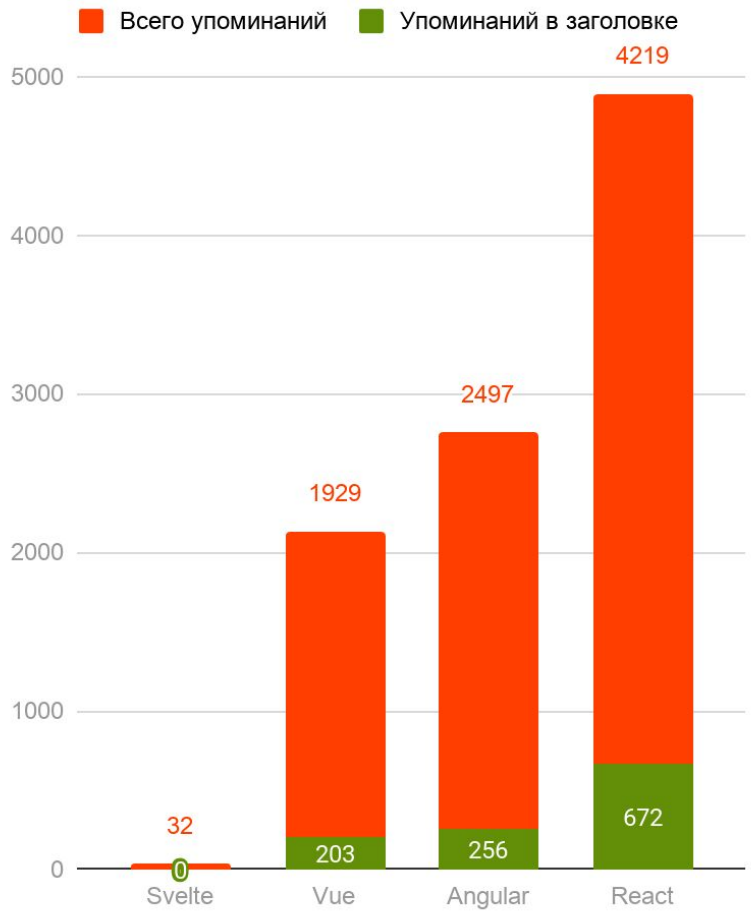
Вторая версия — апрель 2018

Третья версия — апрель 2019

Уже используют Svelte

Скрин https://svelte.dev от 14.08.2020

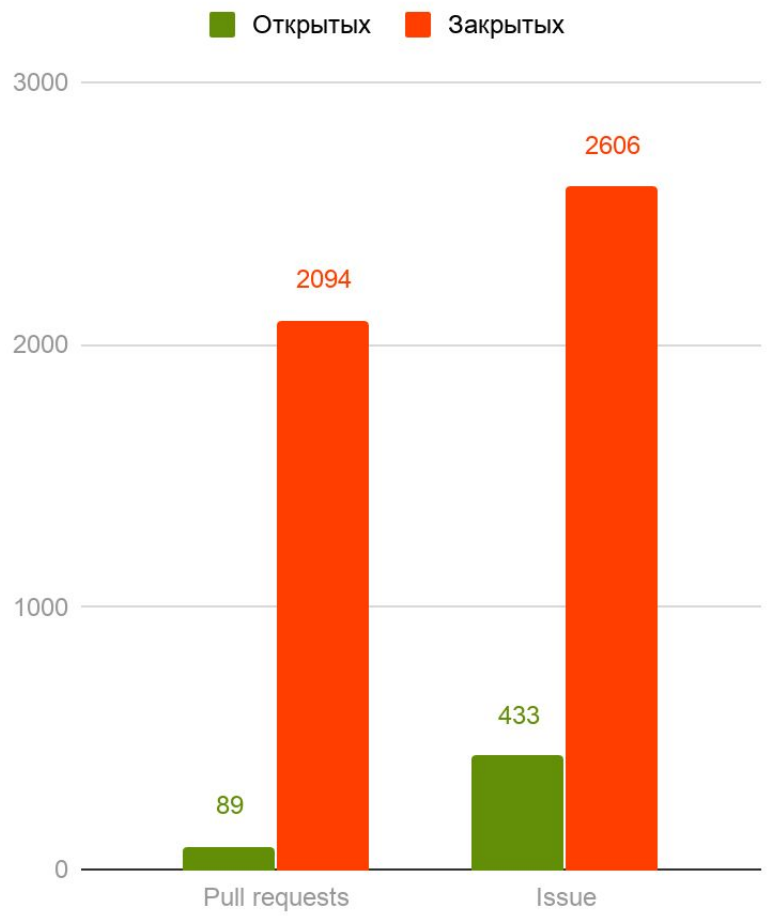
Упоминания на hh.ru



- Svelte: 32 упоминания
  0 в заголовке
- Vue: 1 929 упоминаний
  203 в заголовке
- Angular: 2 497 упоминаний
  256 в заголовке
- React: 4 219 упоминаний
  672 в заголовке

Данные на 14.08.2020

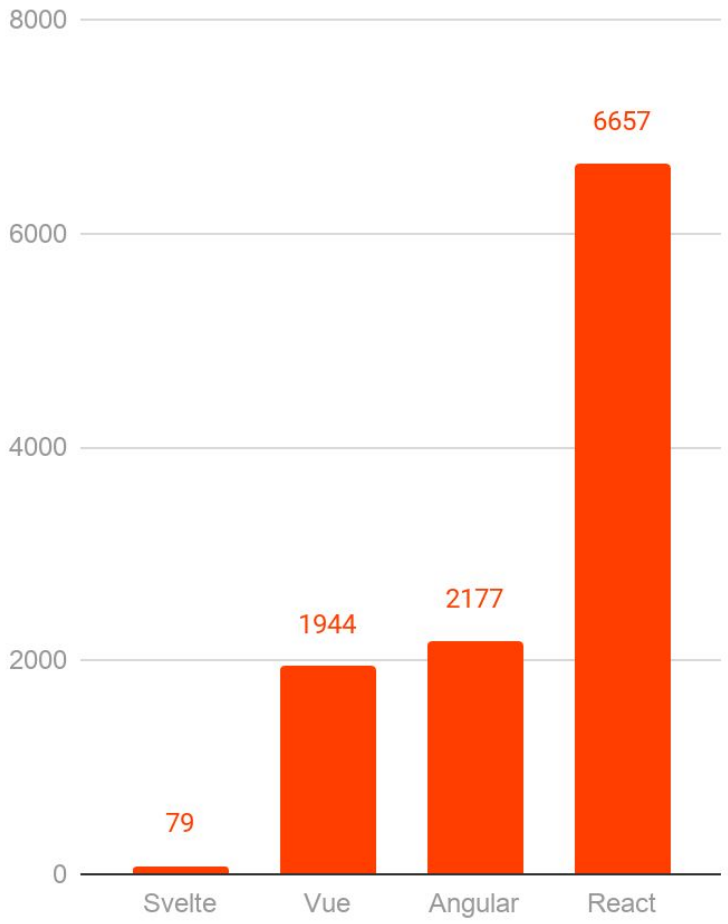
Активность в репозитории



- Новые коммиты почти каждый день.
- 433 открытых ишью на 2 606 закрытых.
- 89 открытых пул реквестов на 2 094 закрытых.

Данные на 14.08.2020

Количество пакетов в npm



- Svelte: 79 упоминаний.
- Vue: 1 944 упоминаний.
- Angular: 2 177 упоминаний.
- React: 6 657 упоминаний.

Данные на 14.08.2020

Компиляция

- Экономит 90 КБ в сравнении с Vue,
- 130 КБ в сравнении с React + React-DOM,
- 300 КБ в сравнении с Angular.

Подключение в React

```
import Widget from './Widget.svelte';

export default class extends PureComponent {
  componentDidMount() {
    this.widget = new Widget({
      target: this.el,
      data: { username: this.props.username }
    });
  }

  render() {
    return <div ref={ el => this.el = el; }/>;
  }
}
```

Простой компонент на Svelte

```
<script>
  export let name;
</script>

{#if name}
  Hello, {name}!
{/if}

<style>
  .button { display: block; }
</style>
```

CSS в Svelte

```
<div class="block">
  <div class="block__element">
    Text
  </div>
</div>

<style>
  .block {
    padding: 10px;
  }

  .block__element {
    background-color: red;
  }
</style>
```

CSS в Svelte

```
<div class="block svelte-iyiy2j">
  <div class="block__element svelte-iyiy2j">
    Text
  </div>
</div>

<style>
  .block.svelte-iyiy2j {
    padding: 10px;
  }

  .block__element.svelte-iyiy2j {
    color: red;
  }
</style>
```

CSS в Svelte

```
<style>
  :global(.class-for-mixing) {
    color: red;
  }
</style>
```

JS в Svelte

```
<script>
  export let name;                // Так объявляются пропсы
  export let phone = '';          // Можно передать им значение по умолчанию

  $: nickName = name + phone;     // Реактивное объявление переменной
</script>
```

HTML в Svelte

```
{#if name}
  Hello, {name}!
{/if}

<ul>
  {#each items as item}
    <li>{item}</li>
  {/each}
</ul>
```

Асинхронность в React

```
async componentDidMount() {
  const data = await getDataFromAPI();
  this.setState({ data });
}

render() {
  if (!data.name) return <div>Loading...<div>;

  return <div>Hello {data.name}!</div>;
}
```

Асинхронность в Svelte

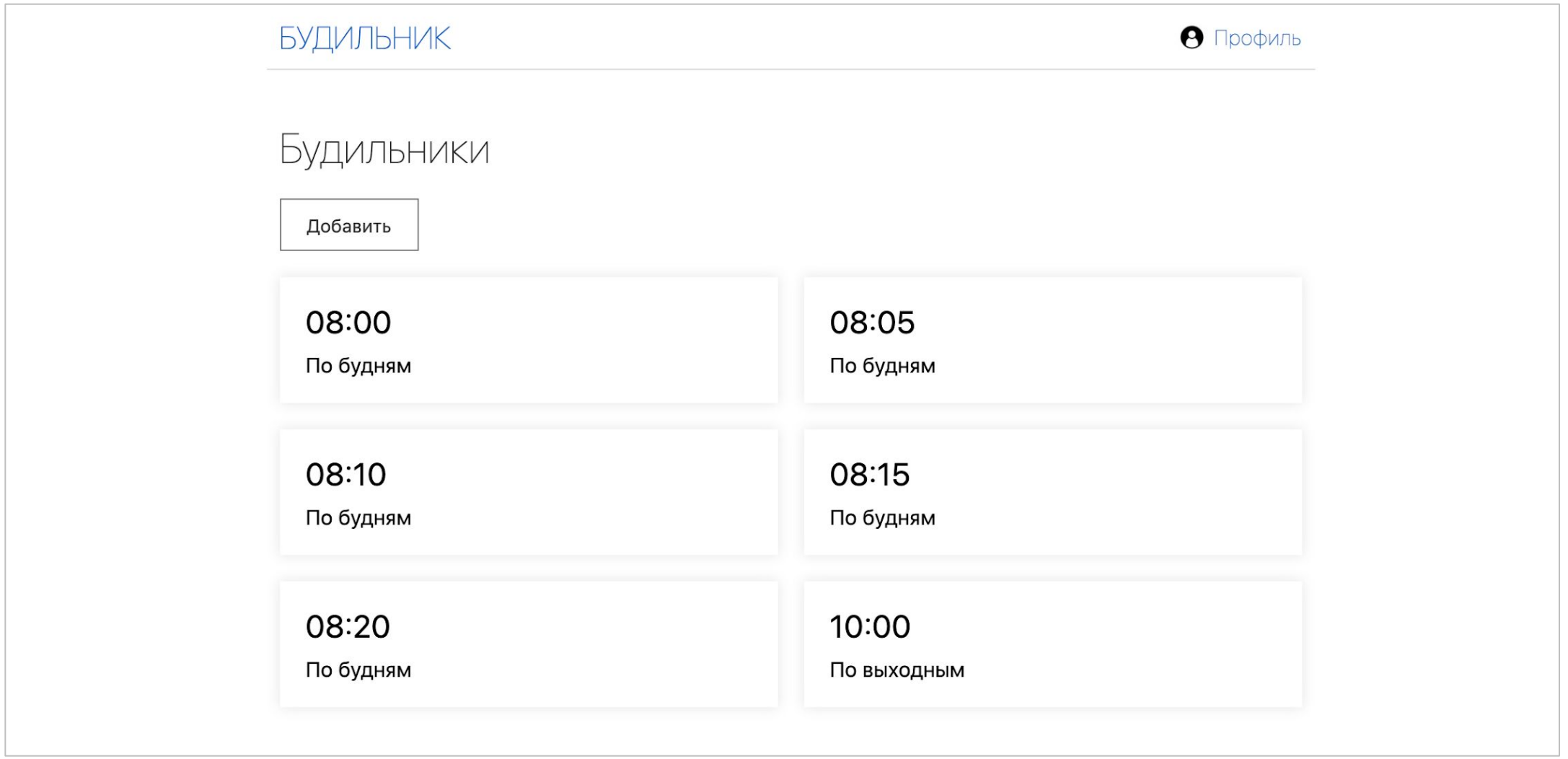
```
<script>
  import { onMount } from 'svelte';

  let data = {};

  onMount(async () => {
    data = await getDataFromAPI();
  });
</script>

{#if !data.name}
  Loading...
{:else}
  Hello {data.name}!
{/if}
```

Асинхронность в Svelte

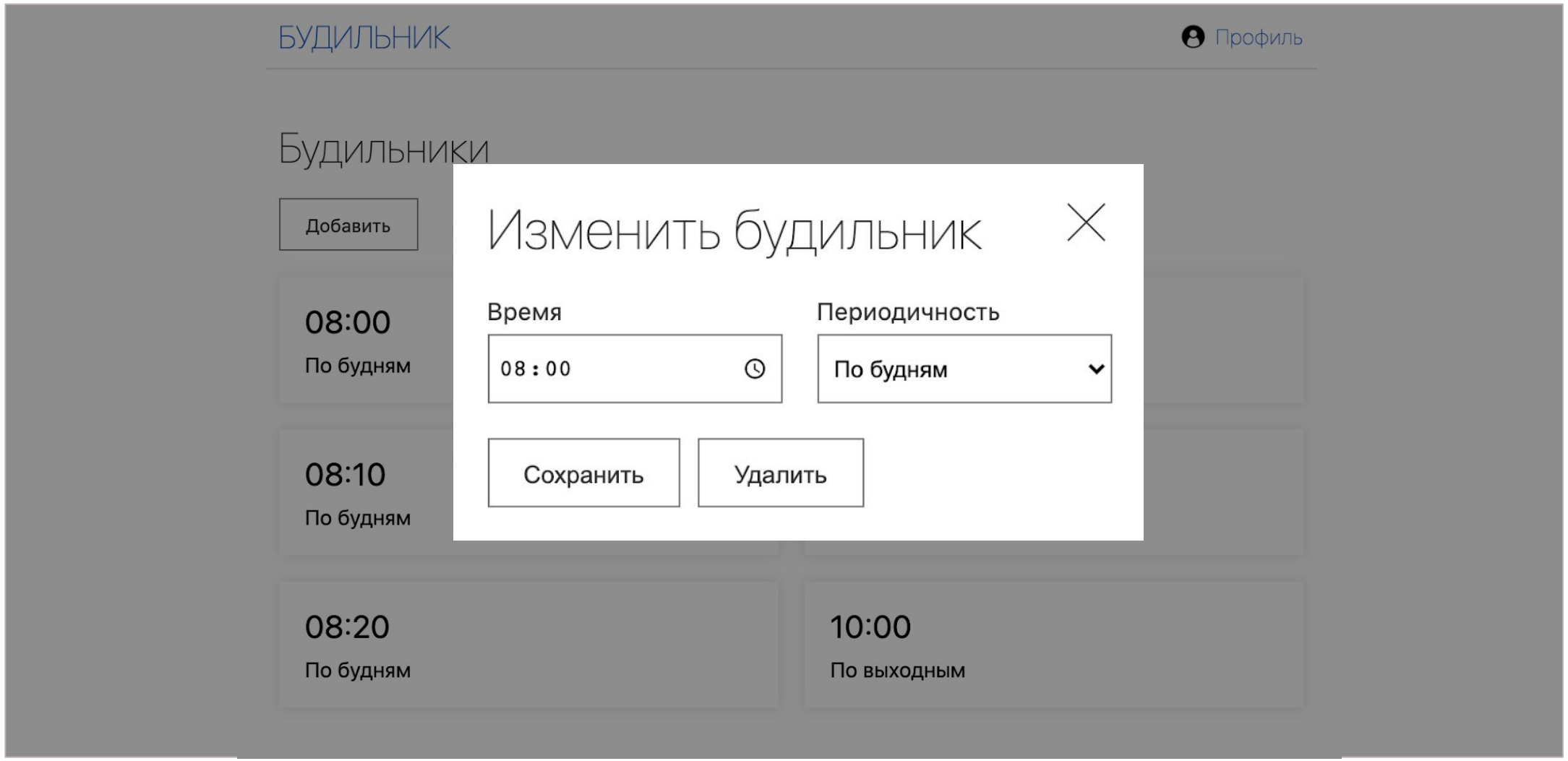
```
<script>
  const dataPromise = getDataFromAPI();
</script>

{#await dataPromise}
  Loading...
{:then data}
  Hello {data.name}!
{/await}
```


Доброе утро!

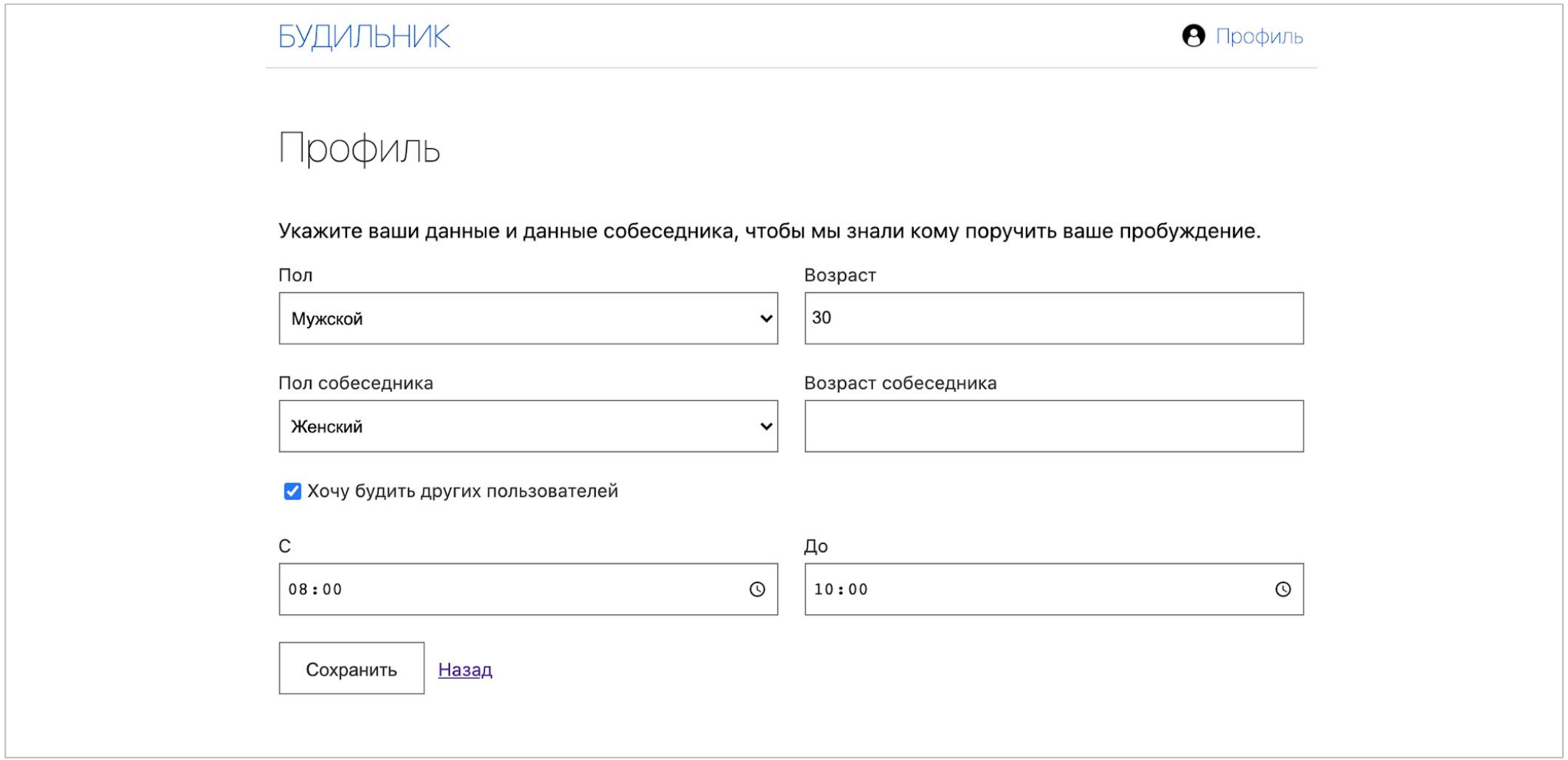
Пробное приложение



Пробное приложение



Пробное приложение



Сборка проекта на Svelte

```
# Шаблон проекта с Rollup
degit sveltejs/template my-svelte-project
cd my-svelte-project
npm install
npm run dev

# Шаблон проекта с Webpack
degit sveltejs/template-webpack my-svelte-project
cd my-svelte-project
npm install
npm run dev
```

Сборка проекта на Svelte

```
npm install -D svelte-preprocess node-sass
```

Сборка проекта на Svelte

```
npm install -D svelte-preprocess node-sass
```

```
const preprocess = require('svelte-preprocess');

{
  test: /\.svelte$/,
  use: {
    loader: 'svelte-loader',
    options: {
      preprocess: preprocess({})
    }
  }
}
```

Сборка проекта на Svelte

```
<script src="./component.js"></script>
<style src="./component.scss"></style>
<template src="./component.html"></template>
```

Компиляция простого компонента

```
<div>Hello, world!</div>
```


```
import {
  SvelteComponent,
  detach,
  element,
  init,
  insert,
  noop,
  safe_not_equal
} from "svelte/internal";


function create_fragment(ctx) {
  let div;


  return {
    c() {
      div = element("div");
      div.textContent = "Hello, world!";
    },
    m(target, anchor) {
      insert(target, div, anchor);
    },
    p: noop,
    i: noop,
    o: noop,
    d(detaching) {
      if (detaching) detach(div);
    }
  };
}


export default class App extends SvelteComponent {
  constructor(options) {
    super();
    init(this, options, null, create_fragment, safe_not_equal, {});
  }
}
```

Компиляция простого компонента

36 строк
компонента из одной
строки текста

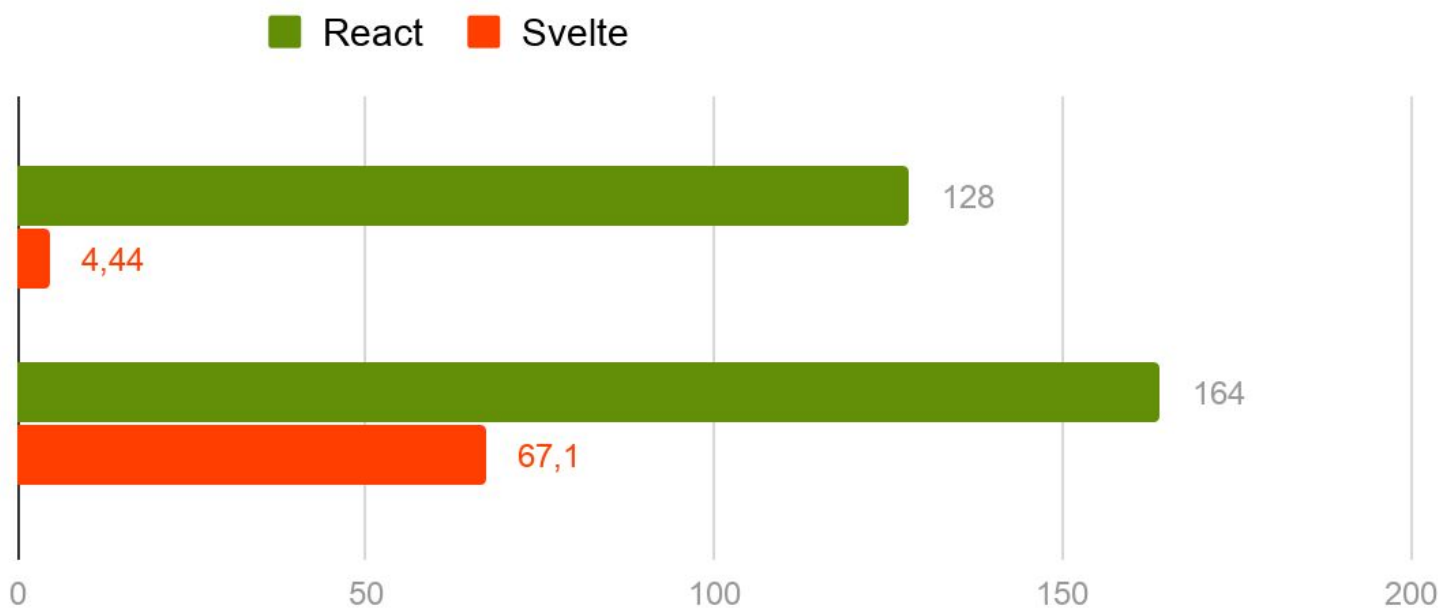
+ 1 680 строк
импортов
из Svelte

Размер бандла и время сборки

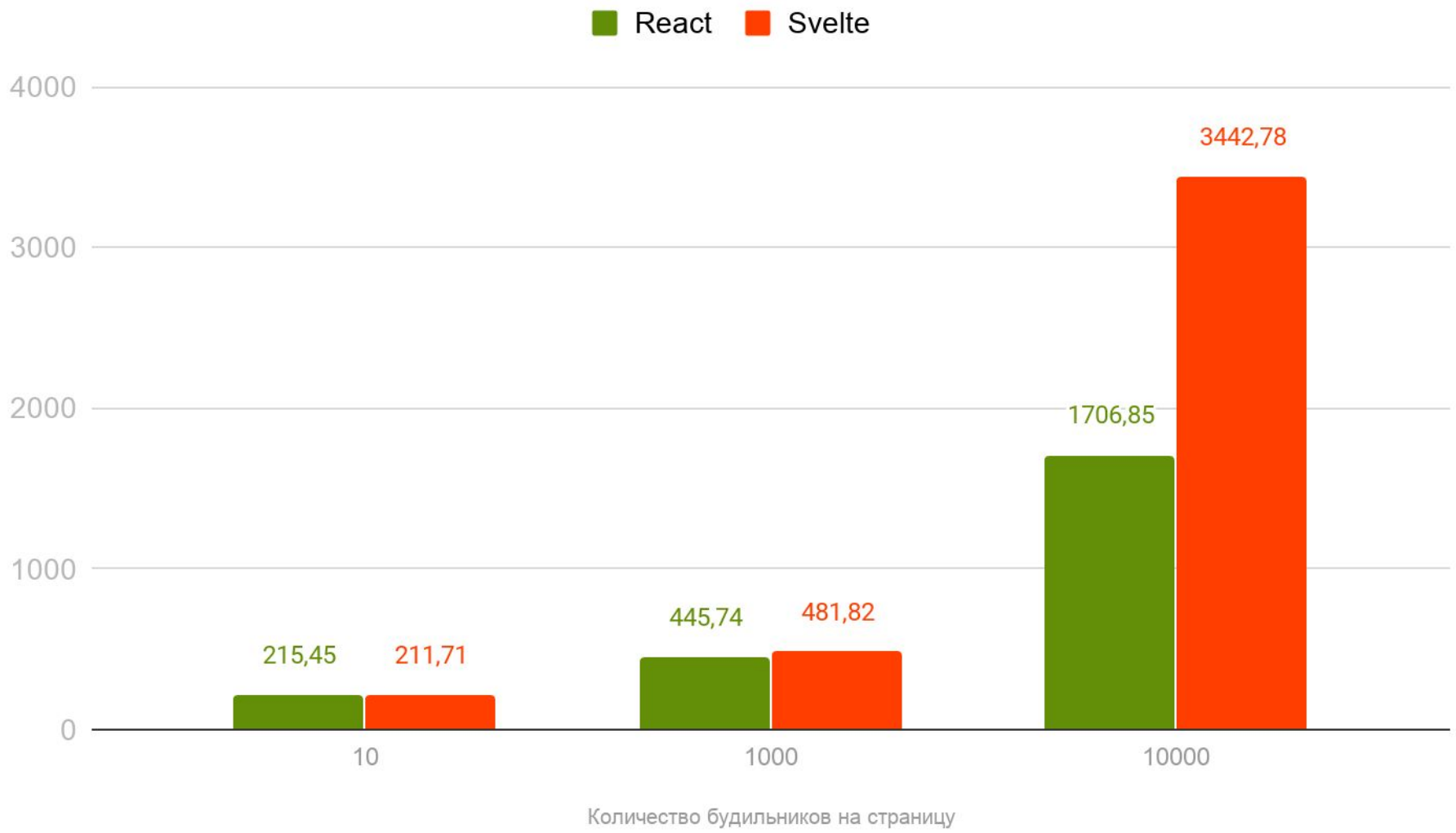
	React	Svelte	Разница
Среднее время сборки приложения **Будильник** (10 запусков)	**1 783.1 ms**	**1 441.6 ms**	**341.5 ms**

Размер бандла приложения «Hello world!», КБ

Размер бандла приложения «Будильник», КБ



Время рендеринга списка будильников, мс



Обработка ошибок

```
// Вызов функции из переменной foo, значение которой undefined
Uncaught TypeError: foo(...) is not a function

// Обращение к свойству bar переменной, значение которой undefined
Uncaught TypeError: Cannot read property 'bar' of undefined

// Обращение к необъявленной переменной baz
Uncaught ReferenceError: baz is not defined
```

Дополнительные материалы

Результаты исследования

Репозиторий приложения «Будильник» на Svelte

Репозиторий приложения «Будильник» на React

Данные по сравнительному тестированию React и Svelte

Пруфы к некоторым утверждениям из доклада

Опрос JS-разработчиков в 2017 году

Статистика использования интернета за 2020 год

Статьи и видео

Легенда о Фреймворке Всевластия — как подключить Svelte-компонент в React-приложение.

Три истории о Svelte — видео доклада Ильи Климова. Основная мысль: Svelte хорош для написания приложений под слабые устройства.

SvelteJS: Релиз второй версии — сравнение второй версии с первой для тех, кому интересна история развития Svelte.