

А так ли нужен Virtual DOM?

Сергей Чикуёнок
«Одноклассники»

Как работает отрисовка в React

JSX →

```
class Example extends React.Component {  
  render() {  
    return <div id="main">  
      <h1>Hello world</h1>  
      <p>My sample component</p>  
    </div>;  
  }  
}
```

vDOM/Fibers → diff → DOM

```
{  
  type: 'div',  
  props: {  
    id: 'main',  
    children: [{  
      type: 'h1',  
      children: 'Hello world'  
    }, {  
      type: 'p',  
      children: 'My sample component'  
    }]  
  }  
}
```

```
<div id="main">  
  <h1>Hello world</h1>  
  <p>My sample component</p>  
</div>
```

Какую задачу выполняет Virtual DOM?

Нужен для более быстрого обновления DOM

Какую задачу выполняет Virtual DOM?

~~Нужен для более быстрого обновления DOM~~

Пример

JSX

```
class Example extends React.Component {  
  render() {  
    return <div id="main">  
      <h1>Hello world</h1>  
      <p>My sample component</p>  
    </div>;  
  }  
}
```

DOM

```
<div id="main">  
  <h1>Hello world</h1>  
  <p>My sample component</p>  
</div>
```

Пример

JSX

```
class Example extends React.Component {  
  render() {  
    return <div id="main">  
      <h1 title={this.props.title}>Hello world</h1>  
      { this.state.enabled && <p>enabled</p>}  
      <p>My sample component</p>  
    </div>;  
  }  
}
```

DOM

```
<div id="main">  
  <h1 title="test">Hello world</h1>  
  <p>enabled</p>  
  <p>My sample component</p>  
</div>
```

Пример

JSX

```
class Example extends React.Component {
  render() {
    return React.createElement(
      "div",
      { id: "main" },
      React.createElement(
        "h1",
        { title: this.props.title },
        "Hello world"
      ),
      this.state.enabled && React.createElement(
        "p",
        null,
        "enabled"
      ),
      React.createElement(
        "p",
        null,
        "My sample component"
      )
    );
  }
}
```

DOM

```
<div id="main">
  <h1 title="test">Hello world</h1>
  <p>enabled</p>
  <p>My sample component</p>
</div>
```

React не может связать
исходный JSX-элемент
с DOM-результатом

Virtual DOM нужен для
решения задач *React*
в выбранном шаблонизаторе*

* *JavaScript*

Virtual DOM — это:

- не про производительность, а про *определение изменений*;
- довольно медленный и ресурсоёмкий;
- добавляет новый класс проблем: `componentShouldUpdate`.
- хорошая абстракция для *императивного* шаблона.

Как можно определять
изменения, если сделать шаблон
полностью декларативным?

```
<div>
  <h1>
    Hello world
  </h1>
  <if test="enabled">
    <p>
      enabled
    </p>
  </if>
  <p>
    My sample component
  </p>
</div>
```

Инструкции на создание элемента

```
<div>  
  <h1>  
    Hello world  
  </h1>  
  <if test="enabled">  
    <p>  
      enabled  
    </p>  
  </if>  
  <p>  
    My sample component  
  </p>  
</div>
```

Контрольная инструкция

Шаг 1: представим результирующие узлы в виде однонаправленного списка



```
<div>
<h1>
Hello world
</h1>
<if test="enabled">
<p>
enabled
</p>
</if>
<p>
My sample component
</p>
</div>
```

Шаг 2: пронумеруем каждый элемент списка

1 ○
2 ○
3 ○
4 ○
5 ○
6 ○
7 ○

```
<div>  
<h1>  
Hello world  
</h1>  
<if test="enabled">  
<p>  
enabled  
</p>  
</if>  
<p>  
My sample component  
</p>  
</div>
```

Шаг 3: отрисуем шаблон и сохраним результат в виде такого же списка

Шаблон

```
1 <div>
2 <h1>
3 Hello world
  </h1>
  <if test="enabled">
4 <p>
5 enabled
  </p>
  </if>
6 <p>
7 My sample component
  </p>
</div>
```

DOM enabled \equiv true

```
1 <div>
2 <h1>
3 Hello world
  </h1>
4 <p>
5 enabled
  </p>
6 <p>
7 My sample component
  </p>
</div>
```


Шаг 4: отрисуем шаблон при `enabled === false` и согласуем списки

Шаблон

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
 `</h1>`
 `<if test="enabled">`
 `<p>`
 `enabled`
 `</p>`
 `</if>`
6 ☐ `<p>`
7 ☐ `My sample component`
 `</p>`
 `</div>`

DOM `enabled === false`

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
 `</h1>`
4 ☐ `<p>`
5 ☐ `enabled`
 `</p>`
6 ☐ `<p>`
7 ☐ `My sample component`
 `</p>`
 `</div>`

Шаг 4: отрисуем шаблон при `enabled ≡ false` и согласуем списки

Шаблон

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
 `</h1>`
 `<if test="enabled">`
 `<p>`
 `enabled`
 `</p>`
 `</if>`
6 ☐ `<p>`
7 ☐ `My sample component`
 `</p>`
 `</div>`

DOM `enabled ≡ false`

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
 `</h1>`
4 ☒ `<p>`
5 ☒ `enabled`
 `</p>`
6 ☐ `<p>`
7 ☐ `My sample component`
 `</p>`
 `</div>`

Шаг 4: рисуем шаблон при enabled \equiv false и согласуем списки

Шаблон

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
`</h1>`
`<if test="value">`
`<p>`
`enabled`
`</p>`
`</if>`
6 ☐ `<p>`
7 ☐ `My sample component`
`</p>`
`</div>`

DOM enabled \equiv false

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
`</h1>`

6 ☐ `<p>`
7 ☐ `My sample component`
`</p>`
`</div>`

Шаг 5: отрисуем шаблон при `enabled ≡ true` И СНОВА СОГЛАСУЕМ

Шаблон

```
1 <div>
2 <h1>
3 Hello world
  </h1>
  <if test="value">
4 <p>
5 enabled
  </p>
  </if>
6 <p>
7 My sample component
  </p>
</div>
```

DOM `enabled ≡ true`

```
1 <div>
2 <h1>
3 Hello world
  </h1>
6 <p>
7 My sample component
  </p>
</div>
```

Шаг 5: отрисуем шаблон при `enabled ≡ true` И СНОВА СОГЛАСУЕМ

Шаблон

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
 ☐ `</h1>`
 ☐ `<if test="value">`
4 ☐ `<p>`
5 ☐ `enabled`
 ☐ `</p>`
 ☐ `</if>`
6 ☐ `<p>`
7 ☐ `My sample component`
 ☐ `</p>`
 ☐ `</div>`

DOM `enabled ≡ true`

1 ☐ `<div>`
2 ☐ `<h1>`
3 ☐ `Hello world`
 ☐ `</h1>`
4 ☒ `<p>`
5 ☒ `enabled`
 ☐ `</p>`
6 ☐ `<p>`
7 ☐ `My sample component`
 ☐ `</p>`
 ☐ `</div>`

Так работает алгоритм
согласования DOM в нашем
новом фронт-энд стэке

Особенности нашего нового СТЭКа

- Основан на веб-компонентах
- Использует только Web API, добавляя недостающие части: шаблонизатор и передача объектов в атрибутах
- Полностью декларативное описание компонента
- `componentShouldUpdate` не нужен

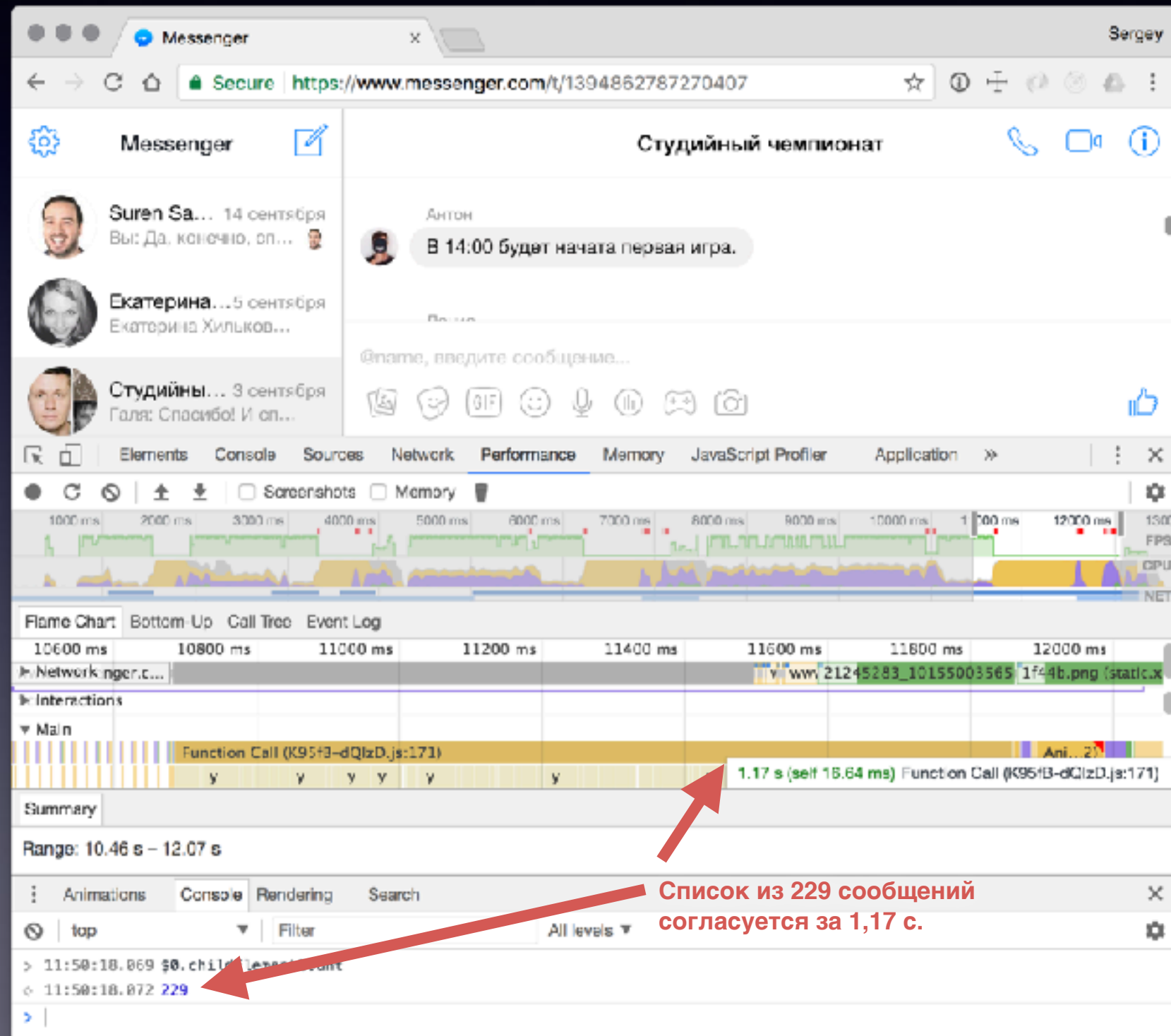
Особенности нашего нового СТЭКа

- Автоматическая оптимизация при сборке
- Изоляция CSS «из коробки»
- Маленький размер: 6 КБ gzip
- Серверный рендеринг на любом языке программирования

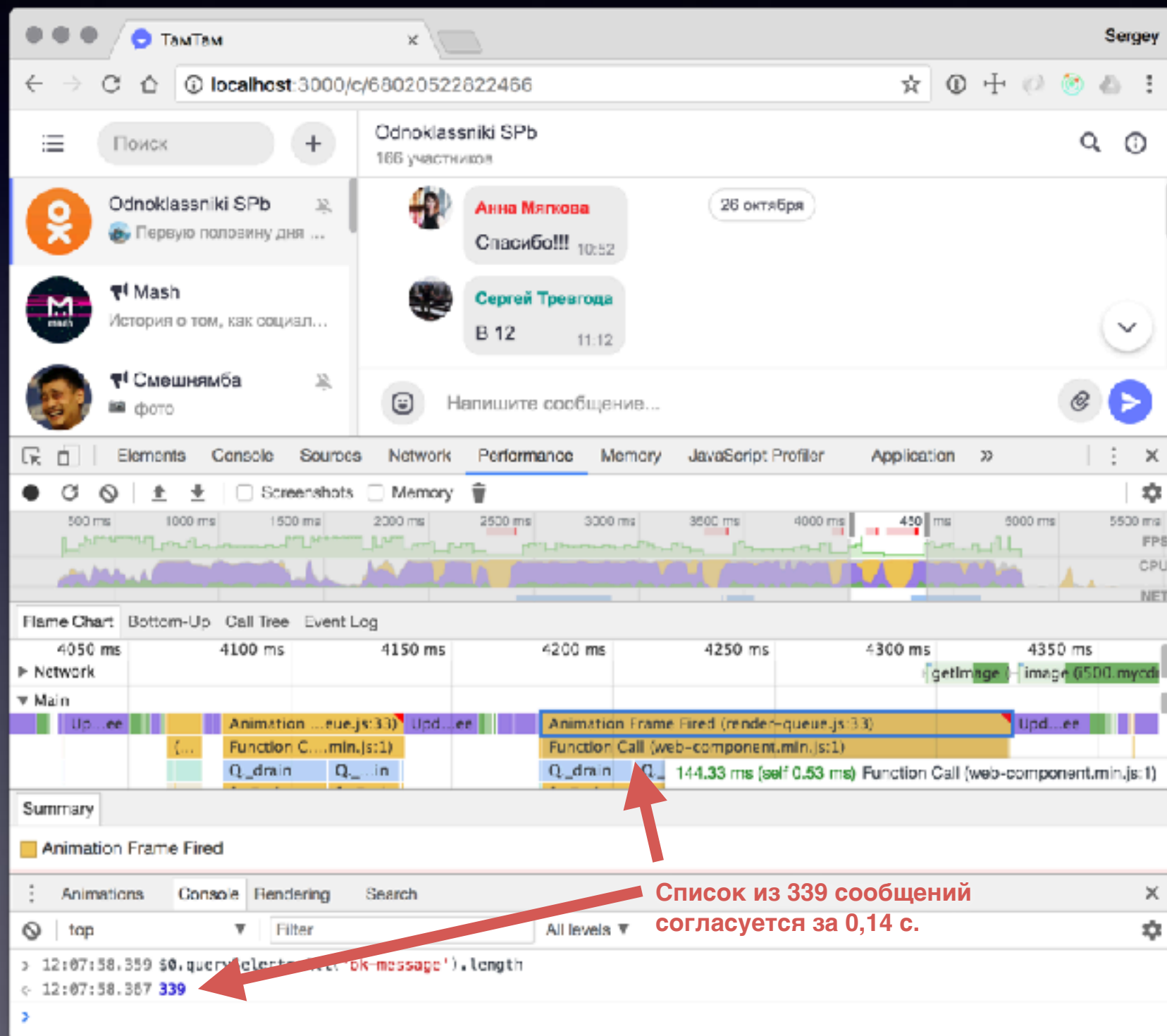
Что будет, если убрать
Virtual DOM из
рендеринга?

messenger.com

React 16



tamtam.chat



Где это можно
пощупать?

Приходите к нам
работать! 🕶️

Вопросы?

serge.che@gmail.com
@chikuyonok