podlodka

React Crew
25 – 29 августа

Паттерны и практики

Николай Шабалин
Сберздоровье, Ведущий разработчик

Доклад
View Transition API + React: паттерны анимаций и подводные камни

Коля Шабалин

Веду канал и пишу статьи

Работаю на двух работах

Сберздоровье

Помогают быть здоровыми — в любое время, в любом месте.

HTML Academy

Помогает людям найти своё профессиональное призвание

https://github.com/nikolai-shabalin

Все ссылки тут

GET OFF
THE BEATEN PATH

DISCOVER HIDDEN CORNERS AND DELIGHTFUL
EXPERIENCES NEARBY

Пример для показа VTA

LOREM IPSUM DOLOR SIT AMET CONSECTETUR ADIPISICING ELIT. ILLO, PRAESENTIUM. VEL IPSAM ADIPISCI MOLLITIA SIMILIQUE VOLUPTATUM REICIENDIS CONSEQUATUR SINT EX, VOLUPTATEM, QUOD ASPERNATUR DEBITIS ISTE VOLUPTAS, DOLOREMQUE QUOS NECESSITATIBUS MINIMA.

Анимированный переход

Поддержка

View Transitions API (single-document) - WD

Usage % of all users ?

Global 87.06%

Limited availability across major browsers

Provides a mechanism for easily creating animated transitions between different DOM states, while also updating the DOM contents in a single step. This API is specific to single-document transitions.

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-110	12-110	3.1-17.6		10-96			3.2-17.7	4-22								
111-138	111-138	18.0-18.4	2-141	97-116	6-10		18.0-18.4	23-27		12-12.1		2.1-4.4.4				2.5
139	139	18.6	142	117	11	139	18.6	28	all	80	15.5	139	141	14.9	13.52	3.1
140-142		26.0-TP	143-145				26.0									

Notes | Test on a real browser | Known issues (0) | Resources (6) | Feedback

All major browser engines are working on implementing this spec.

Поддержка браузерами

Не стоит переживать

у нас есть interop и прогрессивное улучшение

Interop 2025

Ещё месяц назад поддержка Firefox 6.7%

Active Focus Areas					INTEROP
backdrop-filter	95.9%	62.1%	82.4%	81.0%	52.7%
Core Web Vitals	75.1%	75.9%	57.5%	4.4%	4.4%
CSS anchor positioning	97.0%	90.8%	14.7%	14.6%	14.6%
<details> element	99.3%	99.3%	51.4%	72.4%	51.4%
Layout	99.7%	96.1%	95.9%	94.4%	88.2%
Modules	94.6%	94.6%	15.0%	85.6%	11.8%
Navigation API	98.1%	97.6%	0.4%	0.4%	0.4%
Pointer and mouse events	90.4%	81.3%	92.8%	79.6%	71.6%
Remove mutation events	100%	100%	40.0%	30.0%	30.0%
@scope	100%	96.8%	14.6%	79.1%	11.5%
scrollend event	100%	100%	92.8%	2.6%	2.6%
Storage Access API	69.6%	60.7%	81.2%	33.0%	28.4%
text-decoration	91.1%	91.1%	100%	73.8%	73.8%
URLPattern	97.7%	97.7%	0.1%	0.1%	0.1%
View Transition API	96.0%	88.1%	6.7%	87.7%	4.8%
WebAssembly	50.0%	50.0%	50.0%	0.0%	0.0%
Web compat	92.6%	92.6%	73.9%	64.9%	56.4%
WebRTC	11.8%	11.8%	80.1%	73.1%	4.1%
Writing modes	100%	85.7%	97.6%	71.3%	61.3%
TOTAL	87.3%	82.7%	55.1%	49.8%	29.9%

Descriptions of all focus areas

Interop 2025

Уже сегодня 50.8%. Ждём конца года, когда будет 95%+

Active Focus Areas					INTEROP
backdrop-filter	95.9%	62.1%	82.4%	81.0%	52.7%
Core Web Vitals	97.4%	98.3%	71.7%	4.4%	4.4%
CSS anchor positioning	99.1%	92.9%	14.7%	67.4%	13.8%
<details> element	100%	100%	81.5%	72.4%	59.3%
Layout	99.9%	96.3%	96.1%	94.4%	88.2%
Modules	94.6%	94.6%	98.9%	85.6%	79.1%
Navigation API	99.0%	97.8%	0.4%	0.4%	0.4%
Pointer and mouse events	90.4%	90.4%	95.3%	79.6%	74.8%
Remove mutation events	100%	100%	100%	100%	100%
@scope	100%	96.8%	84.3%	79.2%	73.8%
scrollend event	100%	100%	92.8%	2.6%	2.6%
Storage Access API	69.6%	60.7%	98.8%	44.3%	37.6%
text-decoration	91.1%	91.1%	100%	78.4%	78.4%
URLPattern	97.7%	97.7%	0.1%	100%	0.1%
View Transition API	98.0%	90.5%	50.8%	87.7%	42.7%
WebAssembly	50.0%	50.0%	81.6%	0.0%	0.0%
Web compat	97.5%	92.6%	88.9%	69.8%	61.3%
WebRTC	74.4%	74.4%	80.1%	79.9%	57.6%
Writing modes	100%	85.7%	100%	72.7%	63.3%
TOTAL	92.3%	87.9%	74.6%	63.1%	46.8%

Descriptions of all focus areas

View Transition

VTA

Способ анимирования перехода между двумя состояниями DOM — даже между загрузками страниц

Shared Element Transitions

Раньше спецификация называлась SET, особенно в статьях 2021 и 2022

Peyto Lake, Canada

Two Jack Lake, Canada

Несколько карточек

При клике показывается увеличенная версия

```
1   image.addEventListener("click", () =>{
2     moveImageToModal(image);
3   });
```

Клик по картинке открывает большую версию

```
1   image.addEventListener("click", () =>{
2     document.startViewTransition(() => {
3       moveImageToModal(image);
4     })
5   });
```

Обернём функцию в startViewTransition()

```
1   image.addEventListener("click", () =>{
2     if (!document.startViewTransition) {
3       moveImageToModal(image);
4       return;
5     }
6
7     document.startViewTransition(() => {
8       moveImageToModal(image);
9     })
10  });
```

Добавим проверку на работоспобность

Что только что произошло?

```
document.startViewTransition(() => { })
```

```
document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

R >

```
1    document.startViewTransition(() => { })
```

```
1    document.startViewTransition(() => { })
```

Как браузер понимает, что именно снимать?

```
1    :root {
2      view-transition-name: root;
3    }
```

Умолчания браузера

Два состояния

::view-transition-old(root)

Можем обратиться к старому состоянию

::view-transition-new(root)

Можем обратиться к новому состоянию

```
1  ::view-transition {} /* 1 */
2    ::view-transition-group(root) {} /* 2 */
3      ::view-transition-image-pair(root) {} /* 3 */
4        ::view-transition-old(root) {} /* 4 */
5        ::view-transition-new(root) {} /* 5 */
```

Псевдодерево в DOM

```
1  :root::view-transition {}
2    ::view-transition-group(root) {}
3        ::view-transition-image-pair(root) {}
4          ::view-transition-old(root) {}
5          ::view-transition-new(root) {}
```

document.startViewTransition(() ⇒ {})

Настраиваем анимации

```
1   :root::view-transition-old(*) {
2     animation: -ua-fade-out;
3   }
4
5   :root::view-transition-new(*) {
6     animation: -ua-fade-in;
7   }
8
```

Дефолтная crossfade анимация

```
1   ::view-transition-group(*) {
2     animation-duration: 2s;
3   }
4
```

Замедляем анимацию

```
1  ::root::view-transition
2    ::view-transition-group(root)/* 2s */
3      ::view-transition-image-pair(root)/* inherit */
4        ::view-transition-old(root)/* inherit */
5        ::view-transition-new(root)/* inherit */
```

Анимация наследуется

```
1   @keyframes scale-down {
2     to { scale: 0; }
3   }
4
5   @keyframes scale-up {
6     to { scale: 1; }
7   }
8
9   ::view-transition-old(*) {
10    animation-name: scale-down;
11  }
12
13  ::view-transition-new(*) {
14    animation-name: scale-up;
15  }
16
```

Заменяем анимацию

Снимки — это пиксели

```
1
2   ::view-transition-old(*) {
3     color: red; /* 🚫 */
4   }
5
6   ::view-transition-new(*) {
7     animation-delay: var(--delay);  /* 🚫 */
8   }
9
```

Это не сработает

Заголовок

Заголовок

Пиксели — это добро

Меняем несколько элементов

Коротко: можно

```
1
2   .photo {
3     view-transition-name: product-photo;
4   }
5
6   .info {
7     view-transition-name: product-info;
8   }
9
```

Именуем разные элементы

```
1   ::view-transition-old(product-photo) {
2     /* scaleDown */
3   }
4   ::view-transition-new(product-photo) {
5     /* scaleUp */
6   }
7   ::view-transition-group(product-info) {
8     /* crossfade */
9   }
10
```

Разные анимации

```
1   ::view-transition
2     ::view-transition-group(root)
3         ::view-transition-image-pair(root)
4           ::view-transition-old(root)
5           ::view-transition-new(root)
6     ::view-transition-group(product-photo)
7         ::view-transition-image-pair(product-photo)
8           ::view-transition-old(product-photo)
9           ::view-transition-new(product-photo)
10    ::view-transition-group(product-info)
11        ::view-transition-image-pair(product-info)
12          ::view-transition-old(product-info)
13          ::view-transition-new(product-info)
```

Алгоритм использования VTA

Алгоритм использования

1. **Именовать анимированный элемент view-transition-name**

   Имя должно быть уникальным

2. **Запустить переход между представлениями**

   С помощью startViewTransiton

React

Это эксперимент

Установить, чтобы попробовать

react@experimental
react-dom@experimental

Компонент
<ViewTransition >

```
1  import {
2    unstable_ViewTransition as ViewTransition
3  } from 'react';
4  import { startTransition } from 'react';
5
6  <ViewTransition> // crossfade
7    <div>...</div>
8  </ViewTransition>
```

Импортируем необходимый минимум

```
1   // Открытие модального окна
2    const openModal = () => {
3     startTransition(() => setIsOpen(true))
4    }
5
6   // Закрытие модального окна
7    const closeModal = () => {
8     startTransition(() => setIsOpen(false))
9    }
```

Запускаем с помощью startTransotion

Где уже работает

Практически во всех популярных фреймворках

Спасибо за внимание

Время для вопросов