

Technischer Bericht

Bugfix: iOS Safari — Schaltflächen reagieren beim ersten Antippen nach dem Scrollen nicht

Projekt: Digital & Mobil in Deutschland

Datum: 21. April 2026

1. Problembeschreibung

Auf der Website `digital-mobil-deutschland.vercel.app` wurden Schaltflächen vom Typ `.btn-main` (HTML-Element `<a href= "... ">`) auf iOS Safari nicht beim ersten Antippen aktiviert, wenn der Nutzer zuvor auf der Seite gescrollt hatte.

Symptome:

- Beim ersten Antippen einer Schaltfläche nach dem Scrollen wurde die CSS-Animation (`:active`) ausgelöst, aber der Übergang zur verlinkten Seite fand nicht statt.
- Beim zweiten Antippen funktionierte alles normal.
- Das Problem trat ausschließlich auf iOS Safari auf (iPhone), auf Desktop-Browsern war alles einwandfrei.

2. Ursachenanalyse

iOS Safari hat ein eingebautes Verhalten, bei dem der erste Fingertipp nach dem Scrollen als „Scroll-Stopp“ interpretiert wird und nicht als Klick. Das bedeutet:

- Das `touchstart`-Ereignis wird ausgelöst (daher die Animation).
- Das `click`-Ereignis wird jedoch nicht weitergeleitet.
- Die CSS-Eigenschaft `touch-action: manipulation`, die für alle `<a>`-Elemente gesetzt war, hat das Problem nicht vollständig behoben, da das Scrollen selbst den ersten Tap blockierte.

3. Lösungsweg

Schritt 1: CSS-Fix (`touch-action`)

In `components.css` wurde `touch-action: manipulation` zur Klasse `.btn-main` hinzugefügt:

```
.btn-main {  
    touch-action: manipulation; /* iOS tap delay fix */  
}
```

Dies reduziert die 300-ms-Verzögerung, behebt aber nicht das Scroll-Blocker-Problem vollständig.

Schritt 2: JavaScript-Fix (touchend + Bewegungserkennung)

Am Ende von tabs.js wurde folgender Code ergänzt:

```
document.querySelectorAll('a.btn-main').forEach(link => {
  let startY = 0;
  let moved = false;

  link.addEventListener('touchstart', function(e) {
    startY = e.touches[0].clientY;
    moved = false;
  }, { passive: true });

  link.addEventListener('touchmove', function() {
    moved = true;
  }, { passive: true });

  link.addEventListener('touchend', function(e) {
    if (moved) return;
    e.preventDefault();
    const href = this.getAttribute('href');
    if (href) setTimeout(() => window.location.href = href, 180);
  }, { passive: false });
});
```

4. Erklärung der Lösung

- touchstart — merkt sich die Y-Position des Fingers und setzt moved = false.
- touchmove — wenn der Finger sich bewegt, wird moved = true gesetzt. Das bedeutet: der Nutzer scrollt, kein Klick.
- touchend — wenn moved = true, wird der Übergang nicht ausgeführt. Andernfalls wird e.preventDefault() aufgerufen, um das native Click-Ereignis zu unterdrücken, und nach 180 ms erfolgt die Navigation.
- Die 180-ms-Verzögerung ermöglicht es der CSS :active-Animation, vollständig abzuspielen, bevor die Seite wechselt.

5. Ergebnis

- Alle .btn-main-Schaltflächen mit href reagieren nun beim ersten Antippen auf iOS Safari.
- Die Scroll-Geste löst keine unbeabsichtigte Navigation aus.
- Die Klick-Animation bleibt erhalten.
- Das Verhalten auf Desktop-Browsern wurde nicht verändert.

6. Geänderte Dateien

- css/components.css — touch-action: manipulation zu .btn-main hinzugefügt
- js/tabs.js — iOS touchend-Fix am Ende der Datei ergänzt

Технический отчёт

Исправление бага: iOS Safari — кнопки не реагируют на первое нажатие после прокрутки

Проект: Digital & Mobil in Deutschland

Дата: апрель 2026

1. Описание проблемы

На сайте `digital-mobil-deutschland.vercel.app` кнопки типа `.btn-main` (элемент `<a href="...">`) на iOS Safari не срабатывали с первого нажатия, если перед этим пользователь пролистывал страницу.

Симптомы:

- При первом нажатии CSS-анимация (`:active`) запускалась, но переход по ссылке не происходил.
- При втором нажатии всё работало нормально.
- Проблема возникала только на iOS Safari (iPhone). На десктопных браузерах всё работало корректно.

2. Анализ причины

iOS Safari имеет встроенное поведение: первое нажатие пальцем после прокрутки воспринимается как «остановка скролла», а не как клик. Это означает:

- Событие `touchstart` срабатывает (поэтому анимация видна).
- Но событие `click` не пробрасывается.
- CSS-свойство `touch-action: manipulation`, прописанное для всех `<a>`-элементов, не решило проблему полностью.

3. Процесс решения

Шаг 1: CSS-фикс (`touch-action`)

В `components.css` добавлено свойство `touch-action: manipulation` в класс `.btn-main`:

```
.btn-main {  
    touch-action: manipulation; /* iOS tap delay fix */  
}
```

Это убирает 300-миллисекундную задержку, но не решает проблему скролла полностью.

Шаг 2: JavaScript-фикс (touchend + определение движения)

В конец tabs.js добавлен следующий код:

```
document.querySelectorAll('a.btn-main').forEach(link => {
  let startY = 0;
  let moved = false;

  link.addEventListener('touchstart', function(e) {
    startY = e.touches[0].clientY;
    moved = false;
  }, { passive: true });

  link.addEventListener('touchmove', function() {
    moved = true;
  }, { passive: true });

  link.addEventListener('touchend', function(e) {
    if (moved) return;
    e.preventDefault();
    const href = this.getAttribute('href');
    if (href) setTimeout(() => window.location.href = href, 180);
  }, { passive: false });
});
```

4. Пояснение решения

- touchstart — запоминает Y-позицию пальца и устанавливает moved = false.
- touchmove — если палец движется, moved = true. Значит: пользователь скроллит, не нажимает.
- touchend — если moved = true, переход не выполняется. Иначе — e.preventDefault() блокирует нативный click, а через 180 мс происходит навигация.
- Задержка 180 мс позволяет CSS-анимации :active полностью отыграть перед сменой страницы.

5. Результат

- Все кнопки .btn-main с href срабатывают с первого нажатия на iOS Safari.
- Жест скролла не вызывает случайного перехода.
- Анимация нажатия сохранена.
- Поведение на десктопных браузерах не изменилось.

6. Изменённые файлы

- css/components.css — добавлено touch-action: manipulation в .btn-main
- js/tabs.js — добавлен iOS touchend-фикс в конец файла