Animate Between Pages Without A Framework

Two lines of CSS replace the router. Firefox still does not support them, and you should ship them anyway.

by Shreyas Kalyanpad

PUBLISHED: NOVEMBER 23RD · BY SHREYAS KALYANPAD

Cross document view transitions = the browser animates between two separate HTML pages, with no router in the middle.

For years, wanting a page to slide instead of blink was an honest reason to adopt a framework. You took on a router, a build step and hydration bugs, and your links moved nicely.

That trade is over for most sites. It did not end the way anyone expected.

> *Firefox shipped the half that needs JavaScript. It skipped the half that needs two lines of CSS.*

Six Things To Know Before You Delete Your Router

For each one: what teams ship, why it fails, and what to ship instead.



Most of that wall was only ever there to make links move.

1. The Feature Is Two Lines

What teams ship: a client side router, partly so that links can animate.

Why it fails: the animation never needed JavaScript. The whole opt in is `@view-transition { navigation: auto; }` sitting in a stylesheet.

Ship instead: put that rule in the global CSS. Both pages must carry it, because each side opts in on its own.

2. The Support Gap Is Real, And Backwards

What teams ship: a blog post saying this is cross browser now.

Why it fails: it is not. The same page version became Baseline on 14 October 2025, when Firefox 144 landed it. The between pages version is still limited: Chrome 126, Edge 126 and Safari 18.2, and no Firefox release at all. It is on in Nightly only.

Ship instead: treat Firefox as the no animation case, and look at your site that way first.

3. Ship It Anyway

What teams ship: nothing, until support is finished.

Why it fails: waiting made sense when the price was a framework. Here the price is two lines. A browser without support does a normal instant page load. Nothing breaks and nobody sees an error.

Ship instead: compare the failure modes. Skip the CSS and you lose an animation. Skip the framework and you lose a router, a build step and the bugs. To branch in script, the test is `'onpagereveal' in window`.

4. Same Origin, And Little Else

What teams ship: the rule, then a bug report that it never fires.

Why it fails: the conditions are narrow. Both pages must share a scheme, host and port, so a hop to a subdomain will not animate. Address bar trips, bookmarks and reloads are all excluded. In Chrome, a navigation slower than four seconds drops the transition and throws a timeout.

Ship instead: test it by clicking links. That is the only case it covers.

5. The Flash Is Yours To Fix

What teams ship: the default cross fade, plus a flicker on slower pages.

Why it fails: the new page can paint before the thing you are animating to exists.

Ship instead: hold the first paint with `<link rel="expect" blocking="render" href="#main">`. It waits for that element to be in the DOM, not for its images. Firefox does not support this one either, which is at least consistent.

6. Wrap It In Reduced Motion

What teams ship: a full page slide for every visitor.

Why it fails: a whole viewport sliding is the exact thing a motion sensitivity setting exists to stop.

Ship instead: nest the rule inside `@media (prefers-reduced-motion: no-preference)`. Chrome's own guidance writes it that way.

What Works Where

WHAT YOU WANT	HOW YOU WRITE IT	WHERE IT WORKS
Animate inside one page	`startViewTransition()`	Baseline since Oct 2025, all engines
Animate between pages	`@view-transition`	Chrome 126, Edge 126, Safari 18.2. Not Firefox
Hold the first paint	`rel="expect"`	Chrome 124, Edge 124, Safari 18.2. Not Firefox
Pick a direction	Navigation API	Baseline since Jan 2026, all engines

The Four Question Audit

Run this before you call it done.

1. Do both pages load a stylesheet that carries the rule?
2. Is every animated link same origin, port included?
3. Does the site still read well in Firefox, with no transition at all?
4. Is the rule inside a reduced motion query?

If the animation was your only reason for a router, hand that complexity back. Three engines out of four look better for it. The fourth gets the plain page load it was always giving you.

Checked against

Read on 23 November 2026.

- MDN, the @view-transition at-rule: syntax, the navigation descriptor, and a compatibility table showing Firefox in preview only
- MDN, View Transition API: the same document and cross document split, and Firefox 144 for startViewTransition
- Chrome for Developers, cross document view transitions: same origin limits, excluded navigations, and the four second timeout
- Chrome team modern web guidance, cross document transitions: Baseline dates, fallback behaviour, and the reduced motion wrapper