SVELTEKIT • ADVANCED

Link options

ON THIS PAGE

In SvelteKit, `<a>` elements (rather than framework-specific `<Link>` components) are used to navigate between the routes of your app. If the user clicks on a link whose `href` is 'owned' by the app (as opposed to, say, a link to an external site) then SvelteKit will navigate to the new page by importing its code and then calling any `load` functions it needs to fetch data.

You can customise the behaviour of links with `data-sveltekit-*` attributes. These can be applied to the `<a>` itself, or to a parent element.

These options also apply to `<form>` elements with `method="GET"`.

data-sveltekit-preload-data

Before the browser registers that the user has clicked on a link, we can detect that they've hovered the mouse over it (on desktop) or that a `touchstart` or `mousedown` event was triggered. In both cases, we can make an educated guess that a `click` event is coming.

SvelteKit can use this information to get a head start on importing the code and fetching the page's data, which can give us an extra couple of hundred milliseconds — the difference between a user interface that feels laggy and one that feels snappy.

We can control this behaviour with the `data-sveltekit-preload-data` attribute, which can have one of two values:

`"hover"` means that preloading will start if the mouse comes to a rest over a link. On mobile, preloading begins on `touchstart`

registered

The default project template has a `data-sveltekit-preload-data="hover"` attribute applied to the `<body>` element in `src/app.html`, meaning that every link is preloaded on hover by default:

```
<body data-sveltekit-preload-data="hover">
  <div style="display: contents">%sveltekit.body%</div>
</body>
```

Sometimes, calling `load` when the user hovers over a link might be undesirable, either because it's likely to result in false positives (a click needn't follow a hover) or because data is updating very quickly and a delay could mean staleness.

In these cases, you can specify the `"tap"` value, which causes SvelteKit to call `load` only when the user taps or clicks on a link:

```
<a data-sveltekit-preload-data="tap" href="/stonks">
  Get current stonk values
</a>
```

You can also programmatically invoke `preloadData` from `$app/navigation`.

Data will never be preloaded if the user has chosen reduced data usage, meaning `navigator.connection.saveData` is `true`.

data-sveltekit-preload-code

Even in cases where you don't want to preload *data* for a link, it can be beneficial to preload the *code*. The `data-sveltekit-preload-code` attribute works similarly to `data-sveltekit-preload-data`, except that it can take one of four values, in decreasing 'eagerness':

`"eager"` means that links will be preloaded straight away

`"tap"` - as above, except that only code is preloaded

Note that `viewport` and `eager` only apply to links that are present in the DOM immediately following navigation — if a link is added later (in an `{#if ...}` block, for example) it will not be preloaded until triggered by `hover` or `tap`. This is to avoid performance pitfalls resulting from aggressively observing the DOM for changes.

> Since preloading code is a prerequisite for preloading data, this attribute will only have an effect if it specifies a more eager value than any `data-sveltekit-preload-data` attribute that is present.

As with `data-sveltekit-preload-data`, this attribute will be ignored if the user has chosen reduced data usage.

data-sveltekit-reload

Occasionally, we need to tell SvelteKit not to handle a link, but allow the browser to handle it. Adding a `data-sveltekit-reload` attribute to a link...

```
<a data-sveltekit-reload href="/path">Path</a>
```

...will cause a full-page navigation when the link is clicked.

Links with a `rel="external"` attribute will receive the same treatment. In addition, they will be ignored during prerendering.

data-sveltekit-replacestate

Sometimes you don't want navigation to create a new entry in the browser's session history. Adding a `data-sveltekit-replacestate` attribute to a link...

```
Docs
```

...will replace the current `history` entry rather than creating a new one with `pushState` when the link is clicked.

data-sveltekit-keepfocus

Sometimes you don't want focus to be reset after navigation. For example, maybe you have a search form that submits as the user is typing, and you want to keep focus on the text input. Adding a `data-sveltekit-keepfocus` attribute to it...

```
<form data-sveltekit-keepfocus>
  <input type="text" name="query">
</form>
```

...will cause the currently focused element to retain focus after navigation. In general, avoid using this attribute on links, since the focused element would be the `<a>` tag (and not a previously focused element) and screen reader and other assistive technology users often expect focus to be moved after a navigation. You should also only use this attribute on elements that still exist after navigation. If the element no longer exists, the user's focus will be lost, making for a confusing experience for assistive technology users.

data-sveltekit-noscroll

When navigating to internal links, SvelteKit mirrors the browser's default navigation behaviour: it will change the scroll position to 0,0 so that the user is at the very top left of the page (unless the link includes a `#hash`, in which case it will scroll to the element with a matching ID).

In certain cases, you may wish to disable this behaviour. Adding a `data-sveltekit-noscroll` attribute to a link...

```
Docs
```

Disabling options

To disable any of these options inside an element where they have been enabled, use the `"false"` value:

```
<div data-sveltekit-preload-data>
  <!-- these links will be preloaded -->
  <a href="/a">a</a>
  <a href="/b">b</a>
  <a href="/c">c</a>

  <div data-sveltekit-preload-data="false">
    <!-- these links will NOT be preloaded -->
    <a href="/d">d</a>
    <a href="/e">e</a>
    <a href="/f">f</a>
  </div>
</div>
```

To apply an attribute to an element conditionally, do this:

```
<div data-sveltekit-preload-data={condition ? 'hover' : false}>
```

Edit this page on GitHub

PREVIOUS
Errors

NEXT
Service workers