SVELTEKIT • ADVANCED

Snapshots

Ephemeral DOM state — like scroll positions on sidebars, the content of `<input>` elements and so on — is discarded when you navigate from one page to another.

For example, if the user fills out a form but navigates away and then back before submitting, or if the user refreshes the page, the values they filled in will be lost. In cases where it's valuable to preserve that input, you can take a *snapshot* of DOM state, which can then be restored if the user navigates back.

To do this, export a `snapshot` object with `capture` and `restore` methods from a `+page.svelte` or `+layout.svelte`:

+page.svelte

JS TS

```
<script lang="ts">
  import type { Snapshot } from './$types';

  let comment = $state('');

  export const snapshot: Snapshot<string> = {
    capture: () => comment,
    restore: (value) => comment = value
  };
</script>

<form method="POST">
  <label for="comment">Comment</label>
  <textarea id="comment" bind:value={comment} />
  <button>Post comment</button>
</form>
```

When you navigate away from this page, the `capture` function is called immediately before the page updates, and the returned value is associated with the current entry in the

value as soon as the page is updated.

The data must be serializable as JSON so that it can be persisted to `sessionStorage`. This allows the state to be restored when the page is reloaded, or when the user navigates back from a different site.

> Avoid returning very large objects from `capture` — once captured, objects will be retained in memory for the duration of the session, and in extreme cases may be too large to persist to `sessionStorage`.

Edit this page on GitHub

PREVIOUS
Server-only modules

NEXT
Shallow routing