Table of Contents >

useLazyFetch

`useLazyFetch` provides a wrapper around `useFetch` that triggers navigation before the handler is resolved by setting the `lazy` option to `true`.

Description

By default, useFetch blocks navigation until its async handler is resolved.

> `useLazyFetch` has the same signature as `useFetch`.

👉 Read more in Docs > API > Composables > Use Fetch.

Example

```
<script setup lang="ts">
/* Navigation will occur before fetching is complete.
  Handle pending and error states directly within your component's template
*/
const { pending, data: posts } = await useLazyFetch('/api/posts')
watch(posts, (newPosts) => {
  // Because posts might start out null, you won't have access
  // to its contents immediately, but you can watch it.
})
</script>

<template>
```

```
    <div v-if="pending">
      Loading ...
    </div>
    <div v-else>
      <div v-for="post in posts">
        <!-- do something -->
      </div>
    </div>
</template>
```

`useLazyFetch` is a reserved function name transformed by the compiler, so you should not name your own function `useLazyFetch`.

👉 Read more in Docs > Getting Started > Data Fetching #uselazyfetch.

Edit on Github