



\$fetch

Nuxt uses `ofetch` to expose globally the `$fetch` helper for making HTTP requests within your Vue app or API routes.

During server-side rendering, calling `$fetch` to fetch your internal API routes will directly call the relevant function (emulating the request), **saving an additional API call**.

However, using `$fetch` in components without wrapping it with `useAsyncData` causes fetching the data twice: initially on the server, then again on the client-side during hydration, because `$fetch` does not transfer state from the server to the client. Thus, the fetch will be executed on both sides because the client has to get the data again.

We recommend to use `useFetch` or `useAsyncData` + `$fetch` to prevent double data fetching when fetching the component data.

```
<script setup lang="ts">
// During SSR data is fetched twice, once on the server and once on the client.
const dataTwice = await $fetch('/api/item')

// During SSR data is fetched only on the server side and transferred to the client.
const { data } = await useAsyncData('item', () => $fetch('/api/item'))

// You can also useFetch as shortcut of useAsyncData + $fetch
const { data } = await useFetch('/api/item')
</script>
```

👉 [Read more in Docs > Getting Started > Data Fetching.](#)

You can use `$fetch` for any method that are executed only on client-side.

```
<script setup lang="ts">
function contactForm() {
  $fetch('/api/contact', {
    method: 'POST',
```

```
      body: { hello: 'world ' }
    })
  }
</script>

<template>
  <button @click="contactForm">Contact</button>
</template>
```

`$fetch` is the preferred way to make HTTP calls in Nuxt instead of `@nuxt/http` and `@nuxtjs/axios` that are made for Nuxt 2.

[Edit on Github](#)



© 2016-2023 Nuxt - MIT
License

Enterprise

Design Kit
Nuxt Studio

NuxtLabs

