Table of Contents   ❯

refreshNuxtData

`refreshNuxtData` re-fetches all data from the server and updates the page as well as invalidates the cache of `useAsyncData` , `useLazyAsyncData` , `useFetch` and `useLazyFetch` .

Type

```
refreshNuxtData(keys?: string | string[])
```

Parameters:

- `keys` :
  Type: `String | String[]`
  `refreshNuxtData` accepts a single or an array of strings as `keys` that are used to fetch the data. This parameter is **optional**. All `useAsyncData` and `useFetch` are re-fetched when no `keys` are specified.

Examples

Refresh All data

This example below refreshes all data being fetched using `useAsyncData` and `useFetch` on the current page.

pages/some-page.vue
```ts
<script setup lang="ts">
const refreshing = ref(false)
```

```
const refreshAll = async () => {
  refreshing.value = true
  try {
    await refreshNuxtData()
  } finally {
    refreshing.value = false
  }
}
</script>

<template>
  <div>
    <button :disabled="refreshing" @click="refreshAll">
      Refetch All Data
    </button>
  </div>
</template>
```

Refresh Specific Data

This example below refreshes only data where the key matches to count .

```
pages/some-page.vue
<script setup lang="ts">
const { pending, data: count } = await useLazyAsyncData('count', () => $fetch('/api/count'))
const refresh = () => refreshNuxtData('count')
</script>

<template>
  <div>
    {{ pending ? 'Loading' : count }}
  </div>
  <button @click="refresh">Refresh</button>
</template>
```

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

🖉 Edit on Github

Enterprise

Design Kit

NuxtLabs

Nuxt Studio