Composables Directory

Nuxt 3 uses the `composables/` directory to automatically import your Vue composables into your application using auto-imports!

Under the hood, Nuxt auto generates the file `.nuxt/imports.d.ts` to declare the types.

Be aware that you have to run `nuxi prepare`, `nuxi dev` or `nuxi build` in order to let Nuxt generate the types. If you create a composable without having the dev server running, TypeScript will throw an error, such as `Cannot find name 'useBar'.`

Usage

Method 1: Using named export

composables/useFoo.ts

```ts
export const useFoo = () => {
  return useState('foo', () => 'bar')
}
```

Method 2: Using default export

composables/use-foo.ts or composables/useFoo.ts

```ts
// It will be available as useFoo() (camelCase of file name without extension)
export default function () {
  return useState('foo', () => 'bar')
}
```

Usage: You can now use auto imported composable in `.js`, `.ts` and `.vue` files

app.vue

```vue
<script setup lang="ts">
```

```
const foo = useFoo()
</script>

<template>
  <div>
    {{ foo }}
  </div>
</template>
```

Read and edit a live example in Docs > Examples > Features > Auto Imports.

Examples

Nested Composables

You can use a composable within another composable using auto imports:

composables/test.ts

```
export const useFoo = () => {
  const nuxtApp = useNuxtApp()
  const bar = useBar()
}
```

Access plugin injections

You can access plugin injections from composables:

composables/test.ts

```
export const useHello = () => {
  const nuxtApp = useNuxtApp()
  return nuxtApp.$hello
}
```

How Files Are Scanned

Nuxt only scans files at the top level of the `composables/` directory, e.g.:

```
> composables
>   | - index.ts // scanned
>   | - useFoo.ts // scanned
>   | - nested
>   | --- utils.ts // not scanned
```

Only `composables/index.ts` and `composables/useFoo.ts` would be searched for imports.

To get auto imports working for nested modules, you could either re-export them (recommended) or configure the scanner to include nested directories:

Example: Re-export the composables you need from the `composables/index.ts` file:

composables/index.ts

```ts
// Enables auto import for this export
export { utils } from './nested/utils.ts'
```

Example: Scan nested directories inside the `composables/` folder:

nuxt.config.ts

```ts
export default defineNuxtConfig({
  imports: {
    dirs: [
      // Scan top-level modules
      'composables',
      // ... or scan modules nested one level deep with a specific name and file extension
      'composables/*/index.{ts,js,mjs,mts}',
      // ... or scan all modules within given directory
      'composables/**'
    ]
  }
})
```

Edit on Github