a

Journey to Nuxt Icon

How we engineered the best icon solutions for Nuxt



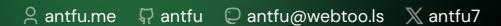
Anthony Fu

Core team member of ♥ Vite ∨ Vue and ▲ Nuxt

Creator of \checkmark Vitest Slidev UnoCSS Type Challenges Flk

Maintainer of O ESLint Stylistic S Shiki S Twoslash

Working at 🛆 NuxtLabs







Engineering Challenges



Adapt the color on the fly

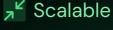










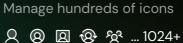


Resize as ease









Large Amount



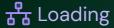
Optimize bundle











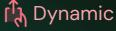
Fetch icons on-demand











Icons not known at compile time



Solution 1:

Icons as images



Fugue Icons by Yusuke Kamiyamane

- Simple, mo additional setup
- Native browser support
- No runtime dependency

♥ Cons

- ♠ No color control
- Pixelated on scaling
- ▲ Large amount of requests
- ⚠ Flash of invisible icons
- Verbose to write
- ▲ Assets & paths management

Solution 2: Web Fonts

Font Awesome 4 for example:

SVG icons are converted to font glyphs and assigned a special unicode character for each icon.

Use CSS classes to apply the icon.

```
<i class="fa fa-address-book-o" aria-hidden="true"></i>
```

```
/* font-awesome.css */
@font-face {
  font-family: 'FontAwesome';
  src: url('../fonts/fontawesome-webfont.eot?v=4.7.0');
}
.fa {
  font: normal normal normal 14px/1 FontAwesome;
}
/* class for each icon */
.fa-address-book-o:before {
  content: '\f2ba';
}
```


- Easy to use
- Colorable
- Scalable, SVG based
- Single request

Cons

- △ Large font file, all icons loaded upfront
- A Hard to customize
- Flash of invisible icons

Solution 3: Components

Example of using MDI icons:

```
<script setup>
import SvgIcon from '@jamescoyle/vue-icon'
import { mdiAccount } from '@mdi/js'
</script>

<template>
    <SvgIcon type="mdi" :path="mdiAccount" />
</template>
```

Example of using Tabler icons:

```
<script setup>
import { IconHome } from '@tabler/icons-vue'
</script>

<template>
     <IconHome />
     </template>
```


- Colorable, Scalable
- Full features of SVG
- No requests, no flash of invisible icons
- SSR friendly

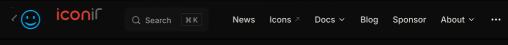
Cons

- ⚠ Creates a lot of SVG DOM elements
- Bundle size
- A Requires specific integration support
- ⚠ Vendor lock-in, hard to switch

Iconify (:) iconify

- One solution for all popular icon sets
- Unified collections of data for consuming
- 100+ icon sets, 200,000+ icons
- Iconify API for dynamic icons





Freedom to choose icons

All popular icon sets, one framework.

Over 200,000 open source vector icons.

Iconify makes it easy to avoid vendor lock-in.



components:

You can use many open source icon sets with a large choice of open source icon components. Thousands of high-quality icons from 100+ icon sets, all validated, cleaned up, optimised and always up to date.

Material Design, Phosphor, Remix, Carbon, Bootstrap, Tabler, Feather, Fluent, IconPark, Octicons and many other icon sets. Twitter Emoji, Fluent Emoji, EmojiOne, Noto Emoji are also available as icon sets.

Use them with the same tools, same reusable and customisable icon



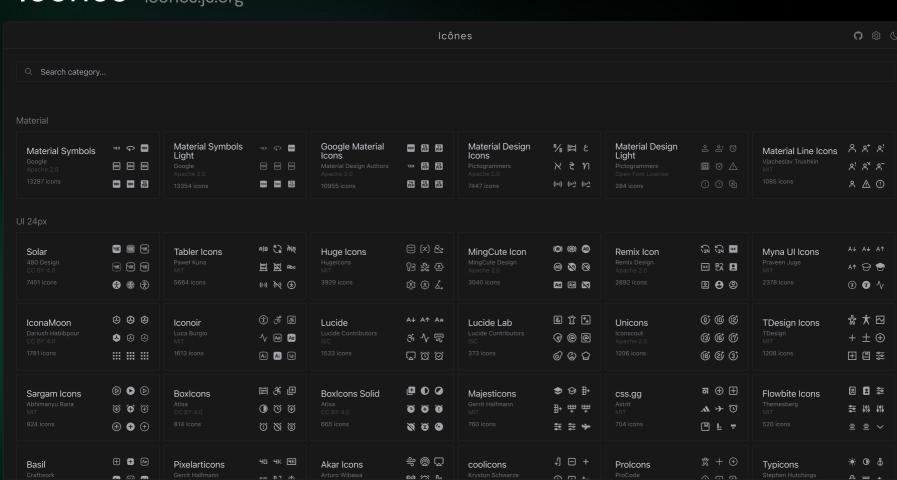
On this page >

- <u>Iconify Icon web component</u> (HTML with or without UI frameworks. Works great with SSR).
- Iconify framework native components (React, Vue, Svelte, Ember).

You can also get raw SVG <u>using various tools</u> and embed them in your pages without any components.

Iconify is supported by a growing community. In addition to Iconify icon components, there are now more ways to use icons, created by amazing open source developers:

Icônes icones.js.org



Solution 4: Iconify Runtime

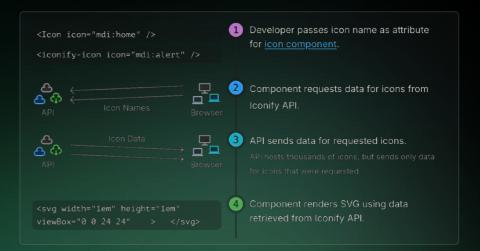
Import the Iconify library on entry (or CDN):

```
import 'iconify-icon'
```

Usage as Web Components:

```
<iconify-icon icon="mdi:home"></iconify-icon>
```

Upon usage, iconify will fetch the icon from their API to render it.



- ⊗ Support dynamic icons on-demand
- Does not require build setup
- Benfits of using SVG icons



- Runtime dependency
- Flash of invisible icons
- A Requires cross site network requests
- ⚠ Not SSR nor PWA friendly

Solution 5: On-demand Components

Introduce of (unplugin-icons

- Vue 3 + Vite + 🐧 Carbon Icons
- React + N Next.js + M Material Design Icons
- Vue 2 + ∧ Nuxt.js + Unicons
- React + i Webpack + i Twemoji
- 💶 🥭 Solid + 🍑 Vite + 윏 Tabler
- Js Vanila + Rollup + BoxIcons
- Web Components + Vite + Ant Design Icons
- Svelte + S SvelteKit + III EOS Icons
- O Any + ② Any + ② Any

- All benifits of using SVG icons
- Use any icons, and bundle for what you use
- Supports most of the popular frameworks
- Optional auto-import

- Creates a lot of SVG DOM elements
- ⚠ Bundle size
- △ Only work for build-time known icons

Solution 6: Pure CSS Icons

Create icons with only CSS, with inline SVG as data URL

```
.i-ph-acorn-duotone {
  mask-image: url(data:...) no-repeat;
  mask-size: 100% 100%;
  background: currentColor;
  height: 1em;
  width: 1em;
}

Catp

<div class="i-ph-acorn-duotone"></div>
```



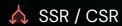
- Olorable, Scalable
- One DOM element per icon
- No requests, no flash of invisible icons
- Zero runtime
- SSR friendly

△ Only work for build-time known icons

▲ Cannot change elements inside icons



INTEGRATIONS CHALLENGES



Seamlessly support both modes, no flashes

Dynamic Icons

Support dynamic known icons, e.g. from Nuxt Content

Performant

No compromise on performance

& Custom Icons

Support loading user-provided custom icons

SOLUTIONS

Iconify Runtime

- Dynamic Icons
- X No SSR
- × No Custom Icons

CSS Icons

- ✓ SSR / CSR
- Perfoment
- X No Dynamic Icons



Dual Rendering Modes

CSS Mode

```
<Icon mode="css" name="ph:arrow-down-duotone" />

div class="i-ph-arrow-down-duotone iconify"></div>

i-ph-arrow-down-duotone {
  mask-image: url(data:...) no-repeat;
  background: currentColor;
  /* ... */
}
```

SVG Mode

Both mode can be serialized on SSR with no runtime cost

Bundles

Bundles to client

- No additional requests
- Renders instantly
- Bundle size is a concern
- Only known icons

Bundles only the static known icons on-demand

Bundles to server

- Bundle size insensitive
- ① Can include full icon sets
- No requests when SSR

Server Bundle



Bundles full sets for frequently used icons, serve on-demand via SSR or client requests

Fetch from Iconify

- All icons on-demand
- Additional requests
- Servers can be far away

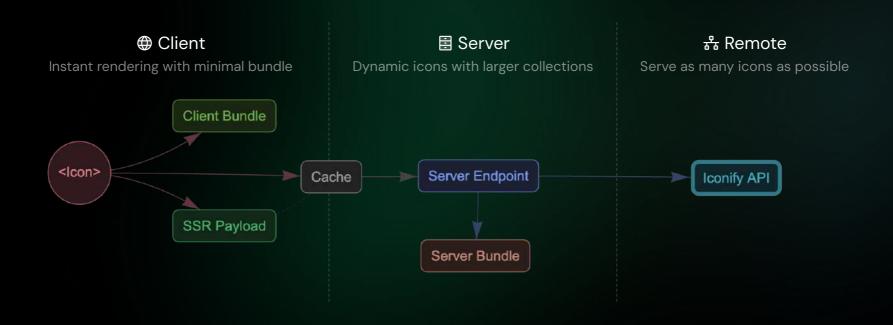
Iconify API



Fallback to serve as many icons as possible

Data Flow

Load icons efficiently while supporting dynamic icons







Icon module for Nuxt with 200,000+ ready to use icons from Iconify.

```
pnpm i -D @nuxt/icon
```

```
export default defineNuxtConfig({
  modules: [
     '@nuxt/icon'
  ]
})
```

```
<template>
    <Icon name="ph:arrow-down-duotone" />
</template>
```

Thank You!

Slides can be found on antfu.me