config

- Configuring custom Iconify API
  - Example
    - Overwriting default API
    - Using custom ID
    - Using IconifyProviders


```
title: Configuring custom Iconify API
replacements:
  - code: '/2/2.0.0/'
    value: '/${svg-framework.version.major}/${svg-framework.version.full}/'
```

Configuring custom Iconify API

This article explains who to configure Iconify icon components to use custom API server.

All Iconify icon components can connect to multiple Iconify API servers. It is done via API providers.

If you are hosting Iconify API server, there are 2 ways to use your icons:

- Replace public Iconify API with your API.
- Use your icons as an alternative icons source by using API provider.

As far as code is concerned, both choices are actually almost identical. The only difference is in first option provider id is empty string `[str]""`, so you are overwriting existing configuration instead of adding new one.

Your users can do whatever they want: overwrite default API provider or add new API provider. They can use whatever provider id they want for it, as long as they use the same provider id in icon names.

Example

For example, you have set up your Iconify API server at `[url]https://iconify.myproject.tld`.

One of icon sets you host has prefix `[icon]material-icons` and it has icon `[icon]home`.

How can your users configure it?

Overwriting default API

Users can overwrite the default API provider like this:

```
import { addAPIProvider } from '@iconify/react';

addAPIProvider('', {
	resources: ['https://iconify.myproject.tld'],
});
```

then in components use your icon like this:

```
import React from 'react';
import { Icon } from '@iconify/react';

function renderHomeIcon() {
	return <Icon icon="material-icons:home" />;
}
```

Using custom ID

Users can also set any id for your API:

```
import { addAPIProvider } from '@iconify/react';

addAPIProvider('awesome-icons', {
```

```
  resources: ['https://iconify.myproject.tld'],
});
```

then in components use your icon like this:

```
import React from 'react';
import { Icon } from '@iconify/react';

function renderHomeIcon() {
  return <Icon icon="@awesome-icons:material-icons:home" />;
}
```

Whatever your users choose, it is up to them. API server does not know what ID users use in icon names and cannot change it.

So one user can overwrite public Iconify API configuration, another user can set ID to `[icon]awesome-icons`, another user can set ID to `[icon]more-icons` and so on.

Using IconifyProviders

Instead of using `[func]addAPIProvider()`, users can use global variable `[var]IconifyProviders` to set configuration.

```
<script>
  // Define global variable
  IconifyProviders = {
    // For overwriting public API
    '': {
      resources: ['https://iconify.myproject.tld'],
    },

    // For creating alternative API provider
    // 'awesome-icons': {
    //   resources: ['https://iconify.myproject.tld'],
    // },
  };
</script>
<script src="/assets/bundle.min.js"></script>
```

This example assumes that `[url]/assets/bundle.min.js` is a bundle that contains all components that use icons and Iconify component.

Similar example for the SVG framework:

```
<head>
<script>
// Define global variable
IconifyProviders = {
// For overwriting public API
// '': {
// resources: ['https://iconify.myproject.tld'],
// },

// For creating alternative API provider
'awesome-icons': {
resources: ['https://iconify.myproject.tld'],
},
};
</script>
<script src="https://code.iconify.design/2/2.0.0/iconify.min.js"></script>
</head>
<body>
<span class="iconify" data-icon="@awesome-icons:material-icons:home"></span>
</body>
```

For more information about API providers, see API providers documentation.

#2
2026-10-31 14:15:28 UTC
2026-10-31 14:15:29 UTC