

Instrumentation

Note: This feature is experimental. To use it, you must explicitly opt in by defining `experimental.instrumentationHook = true;` in your `next.config.js`.

If you export a function named `register` from this file, we will call that function whenever a new Next.js server instance is bootstrapped. When your `register` function is deployed, it will be called on each cold boot (but exactly once in each environment).

Sometimes, it may be useful to import a file in your code because of the side effects it will cause. For example, you might import a file that defines a set of global variables, but never explicitly use the imported file in your code. You would still have access to the global variables the package has declared.

You can import files with side effects in `instrumentation.ts`, which you might want to use in your `register` function as demonstrated in the following example:

TS /instrumentation.ts

```
1 import { init } from 'package-init';
2
3 export function register() {
4   init();
5 }
```

However, we recommend importing files with side effects using `import` from within your `register` function instead. The following example demonstrates a basic usage of `import` in a `register` function:

TS /instrumentation.ts

```
1 export async function register() {
2   await import('package-with-side-effect');
3 }
```

By doing this, you can colocate all of your side effects in one place in your code, and avoid any unintended consequences from importing files.

We call `register` in all environments, so it's necessary to conditionally import any code that doesn't support both `edge` and `nodejs`. You can use the environment variable `NEXT_RUNTIME` to get the current environment. Importing an environment-specific code would look like this:

 /instrumentation.ts



```
1 export async function register() {
2   if (process.env.NEXT_RUNTIME === 'nodejs') {
3     await import('./instrumentation-node');
4   }
5
6   if (process.env.NEXT_RUNTIME === 'edge') {
7     await import('./instrumentation-edge');
8   }
9 }
```