

next.config.js

For custom advanced configuration of Next.js, you can create a `next.config.js` or `next.config.mjs` file in the root of your project directory (next to `package.json`).

`next.config.js` is a regular Node.js module, not a JSON file. It gets used by the Next.js server and build phases, and it's not included in the browser build.

Take a look at the following `next.config.js` example:

```
/**
 * @type {import('next').NextConfig}
 */
const nextConfig = {
  /* config options here */
}

module.exports = nextConfig
```

If you need [ECMAScript modules](#), you can use `next.config.mjs` :

```
/**
 * @type {import('next').NextConfig}
 */
const nextConfig = {
  /* config options here */
}

export default nextConfig
```

You can also use a function:

```
module.exports = (phase, { defaultConfig }) => {
  /**
   * @type {import('next').NextConfig}
   */
  const nextConfig = {
    /* config options here */
  }
  return nextConfig
}
```

Since Next.js 12.1.0, you can use an async function:

```
module.exports = async (phase, { defaultConfig }) => {
  /**
   * @type {import('next').NextConfig}
   */
  const nextConfig = {
```

```

    /* config options here */
  }
  return nextConfig
}

```

`phase` is the current context in which the configuration is loaded. You can see the [available phases](#). Phases can be imported from `next/constants`:

```

const { PHASE_DEVELOPMENT_SERVER } = require('next/constants')

module.exports = (phase, { defaultConfig }) => {
  if (phase === PHASE_DEVELOPMENT_SERVER) {
    return {
      /* development only config options here */
    }
  }

  return {
    /* config options for all phases except development here */
  }
}

```

The commented lines are the place where you can put the configs allowed by `next.config.js`, which are [defined in this file](#).

However, none of the configs are required, and it's not necessary to understand what each config does. Instead, search for the features you need to enable or modify in this section and they will show you what to do.

Avoid using new JavaScript features not available in your target Node.js version. `next.config.js` will not be parsed by Webpack, Babel or TypeScript.