Babel

▶ **Examples**

Next.js includes the `next/babel` preset to your app, which includes everything needed to compile React applications and server-side code. But if you want to extend the default Babel configs, it's also possible.

Adding Presets and Plugins

To start, you only need to define a `.babelrc` file (or `babel.config.js`) in the root directory of your project. If such a file is found, it will be considered as the *source of truth*, and therefore it needs to define what Next.js needs as well, which is the `next/babel` preset.

Here's an example `.babelrc` file:

.babelrc

```
{
  "presets": ["next/babel"],
  "plugins": []
}
```

You can take a look at this file ↗ to learn about the presets included by `next/babel`.

To add presets/plugins **without configuring them**, you can do it this way:

.babelrc

```
{
  "presets": ["next/babel"],
  "plugins": ["@babel/plugin-proposal-do-expressions"]
}
```

Customizing Presets and Plugins

To add presets/plugins **with custom configuration**, do it on the `next/babel` preset like so:

.babelrc

```json
{
  "presets": [
    [
      "next/babel",
      {
        "preset-env": {},
        "transform-runtime": {},
        "styled-jsx": {},
        "class-properties": {}
      }
    ]
  ],
  "plugins": []
}
```

To learn more about the available options for each config, visit babel's documentation↗ site.

Good to know:

- Next.js uses the **current** Node.js version for server-side compilations.
- The `modules` option on `"preset-env"` should be kept to `false`, otherwise webpack code splitting is turned off.