Runtime Config

> **Note**: This feature is considered legacy and does not work with Automatic Static Optimization, Output File Tracing, or React Server Components. Please use environment variables instead to avoid initialization overhead.

To add runtime configuration to your app, open `next.config.js` and add the `publicRuntimeConfig` and `serverRuntimeConfig` configs:

next.config.js

```js
module.exports = {
  serverRuntimeConfig: {
    // Will only be available on the server side
    mySecret: 'secret',
    secondSecret: process.env.SECOND_SECRET, // Pass through env variables
  },
  publicRuntimeConfig: {
    // Will be available on both server and client
    staticFolder: '/static',
  },
};
```

Place any server-only runtime config under `serverRuntimeConfig`.

Anything accessible to both client and server-side code should be under `publicRuntimeConfig`.

> A page that relies on `publicRuntimeConfig` **must** use `getInitialProps` or `getServerSideProps` or your application must have a Custom App with `getInitialProps` to opt-out of Automatic Static Optimization. Runtime configuration won't be available to any page (or component in a page) without being server-side rendered.

To get access to the runtime configs in your app use `next/config`, like so:

```js
import getConfig from 'next/config';
import Image from 'next/image';

```

```
// Only holds serverRuntimeConfig and publicRuntimeConfig
const { serverRuntimeConfig, publicRuntimeConfig } = getConfig();
// Will only be available on the server-side
console.log(serverRuntimeConfig.mySecret);
// Will be available on both server-side and client-side
console.log(publicRuntimeConfig.staticFolder);

function MyImage() {
  return (
    <div>
      <Image
        src={`${publicRuntimeConfig.staticFolder}/logo.png`}
        alt="logo"
        layout="fill"
      />
    </div>
  );
}

export default MyImage;
```