

Server-Side Rendering Caching Headers

This example uses [stale-while-revalidate](#) [cache-control headers](#) in combination with `getServerSideProps` for server-rendering.

`pages/index.js` uses `getServerSideProps` to forward the request header to the React component, as well as setting a response header. This `cache-control` header uses `stale-while-revalidate` to cache the server response.

`pages/index.js` is considered fresh for ten seconds (`s-maxage=10`). If a request is repeated within the next 10 seconds, the previously cached value will still be fresh. If the request is repeated before 59 seconds, the cached value will be stale but still render (`stale-while-revalidate=59`).

In the background, a revalidation request will be made to populate the cache with a fresh value. If you refresh the page, you will see the new value shown.

Deploy your own

Deploy the example using [Vercel](#) or preview live with [StackBlitz](#)



How to use

Execute [create-next-app](#) with [npm](#) or [Yarn](#) to bootstrap the example:

```
npx create-next-app --example ssr-caching ssr-caching-app
# or
yarn create next-app --example ssr-caching ssr-caching-app
# or
pnpm create next-app -- --example ssr-caching ssr-caching-app
```

Deploy it to the cloud with [Vercel](#) ([Documentation](#)).