Adding Offline Support with a Service Worker

If you've run an audit with Lighthouse, you may have noticed a lackluster score in the "Progressive Web App" category. Let's address how you can improve that score.

1. You can add a manifest file. Ensure that the manifest plugin is listed *before* the offline plugin so that the offline plugin can cache the created `manifest.webmanifest`.
2. You can also add offline support, since another requirement for a website to qualify as a PWA is the use of a service worker. Gatsby's offline plugin makes a Gatsby site work offline–and makes it more resistant to bad network conditions–by creating a service worker for your site.

What is a service worker?

A service worker is a script that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction. They increase your site availability in spotty connections, and are essential to making a nice user experience.

It supports features like push notifications and background synchronization.

Using service workers in Gatsby with `gatsby-plugin-offline`

Gatsby provides an awesome plugin interface to create and load a service worker into your site: gatsby-plugin-offline.

We recommend using this plugin together with the manifest plugin. (Don't forget to list the offline plugin after the manifest plugin so that the manifest file can be included in the service worker).

Installing `gatsby-plugin-offline`

```
npm install gatsby-plugin-offline
```

Add this plugin to your `gatsby-config.js`

```
{
  plugins: [
    {
      resolve: `gatsby-plugin-manifest`,
      options: {
        ...
      }
    },
    'gatsby-plugin-offline'
  ]
}
```

That's all you need to add offline support to your Gatsby site.

Note: Service worker registers only in production builds (`gatsby build`).

Displaying a message when a service worker updates

To display a custom message once your service worker finds an update, you can use the `onServiceWorkerUpdateReady` browser API in your `gatsby-browser.js` file. The following code will display a confirm prompt asking the user whether they would like to refresh the page when an update is found:

```
export const onServiceWorkerUpdateReady = () => {
  const answer = window.confirm(
    `This application has been updated. ` +
      `Reload to display the latest version?`
  )

  if (answer === true) {
    window.location.reload()
  }
}
```

Using a custom service worker in Gatsby

You can add a custom service worker if your use case requires something that `gatsby-plugin-offline` doesn't support.

Add a file called `sw.js` in the `static` folder.

Use the `registerServiceWorker` browser API in your `gatsby-browser.js` file.

```
export const registerServiceWorker = () => true
```

That's all! Gatsby will register your custom service worker.

Removing the service worker

If you would like to fully remove the service worker from your site, you can use the plugin `gatsby-plugin-remove-serviceworker` in place of `gatsby-plugin-offline`. See the README for `gatsby-plugin-offline` for instructions how to do this.

References

- Service Workers: an Introduction
- Service Worker API