# gatsby-plugin-gatsby-cloud

Automatically generates a `_headers.json` file and a `_redirects.json` file at the root of the public folder to configure Headers and Redirects on Gatsby Cloud

By default, the plugin will add some basic security headers. You can easily add or replace headers through the plugin config.

## Install

```
npm install --save gatsby-plugin-gatsby-cloud
```

## How to use

```
// In your gatsby-config.js
plugins: [`gatsby-plugin-gatsby-cloud`]
```

## Configuration

If you just need the critical assets, you don't need to add any additional config. However, if you want to add headers, remove default headers, or transform the given headers, you can use the following configuration options.

```
plugins: [
  {
    resolve: `gatsby-plugin-gatsby-cloud`,
    options: {
      headers: {}, // option to add more headers. `Link` headers are transformed by the bel
      allPageHeaders: [], // option to add headers for all pages. `Link` headers are transf
      mergeSecurityHeaders: true, // boolean to turn off the default security headers
      mergeLinkHeaders: true, // boolean to turn off the default gatsby js headers
      mergeCachingHeaders: true, // boolean to turn off the default caching headers
      transformHeaders: (headers, path) => headers, // optional transform for manipulating
      generateMatchPathRewrites: true, // boolean to turn off automatic creation of redirec
    },
  },
]
```

### Headers

You should pass in an object with string keys (representing the paths) and an array of strings for each header.

An example:

```
{
  options: {
    headers: {
```

```
      "/*": [
        "Basic-Auth: someuser:somepassword anotheruser:anotherpassword",
      ],
      "/my-page": [
        // matching headers (by type) are replaced by Gatsby Cloud with more specific route:
        "Basic-Auth: differentuser:differentpassword",
      ],
    },
  }
}
```

Link paths are specially handled by this plugin. Since most files are processed and cache-busted through Gatsby (with a file hash), the plugin will transform any base file names to the hashed variants. If the file is not hashed, it will ensure the path is valid relative to the output `public` folder. You should be able to reference assets imported through javascript in the `static` folder.

Do not specify the public path in the config, as the plugin will provide it for you.

The `_headers.json` file does not inherit headers, and it will replace any matching headers it finds in more specific routes. For example, if you add a link to the root wildcard path (`/*`), it will be replaced by any more specific path. If you want a resource to put linked across the site, you will have to add to every path. To make this easier, the plugin provides the `allPageHeaders` option to inject the same headers on every path.

```
{
  options: {
    allPageHeaders: [
      "Link: </static/my-logo.png>; rel=preload; as=image",
    ],
    headers: {
      "/*": [
        "Basic-Auth: someuser:somepassword anotheruser:anotherpassword",
      ],
    },
  }
}
```

### Redirects

You can create redirects using the `createRedirect` action.

In addition to the options provided by the Gatsby API, you can pass these options specific to this plugin:

| Attribute | Description |
|---|---|
| statusCode | Overrides the HTTP status code which is set to 302 by default or 301 when `isPermanent` is `true`. You can set one here. For example, 200 for rewrites, or 404 for a custom error page. |

An example:

```
createRedirect({ fromPath: "/old-url", toPath: "/new-url", isPermanent: true })
createRedirect({ fromPath: "/url", toPath: "/zn-CH/url", Language: "zn" })
createRedirect({
  fromPath: "/url_that_is/not_pretty",
  toPath: "/pretty/url",
  statusCode: 200,
})
createRedirect({
  fromPath: "/packages/*",
  toPath: "/plugins/*",
})
```

Redirect rules are automatically added for client only paths. The plugin uses the matchPath syntax to match all possible requests in the range of your client-side routes and serves the HTML file for the client-side route. Without it, only the exact route of the client-side route works.

If those rules are conflicting with custom rules or if you want to have more control over them you can disable them in configuration by setting `generateMatchPathRewrites` to `false`.

An asterisk, `*`, will match anything that follows. i.e. `/packages/gatsby-plugin-gatsby-cloud/` will be redirected to `/plugins/gatsby-plugin-gatsby-cloud/`.

**HTTP Strict Transport Security**

HSTS Cheatsheet

Since this header is an opt-in security enhancement with permanent consequences we don't include it as a default feature but use the `allPagesHeaders` to include it.

```
{
  options: {
    allPageHeaders: [
      "Strict-Transport-Security: max-age=31536000; includeSubDomains; preload",
    ],
  }
}
```