gatsby-plugin-google-tagmanager

Easily add Google Tagmanager to your Gatsby site.

Install

```
npm install gatsby-plugin-google-tagmanager
```

How to use

```
// In your gatsby-config.js
plugins: [
  {
    resolve: "gatsby-plugin-google-tagmanager",
    options: {
      id: "YOUR_GOOGLE_TAGMANAGER_ID",

      // Include GTM in development.
      //
      // Defaults to false meaning GTM will only be loaded in production.
      includeInDevelopment: false,

      // datalayer to be set before GTM is loaded
      // should be an object or a function that is executed in the browser
      //
      // Defaults to null
      defaultDataLayer: { platform: "gatsby" },

      // Specify optional GTM environment details.
      gtmAuth: "YOUR_GOOGLE_TAGMANAGER_ENVIRONMENT_AUTH_STRING",
      gtmPreview: "YOUR_GOOGLE_TAGMANAGER_ENVIRONMENT_PREVIEW_NAME",
      dataLayerName: "YOUR_DATA_LAYER_NAME",

      // Name of the event that is triggered
      // on every Gatsby route change.
      //
      // Defaults to gatsby-route-change
      routeChangeEventName: "YOUR_ROUTE_CHANGE_EVENT_NAME",
      // Defaults to false
      enableWebVitalsTracking: true,
      // Defaults to https://www.googletagmanager.com
      selfHostedOrigin: "YOUR_SELF_HOSTED_ORIGIN",
    },
  },
]
```

If you like to use data at runtime for your defaultDataLayer you can do that by defining it as a function.

```
// In your gatsby-config.js
plugins: [
  {
    resolve: "gatsby-plugin-google-tagmanager",
    options: {
      // datalayer to be set before GTM is loaded
      // should be a stringified object or object
      //
      // Defaults to null
      defaultDataLayer: function () {
        return {
          pageType: window.pageType,
        }
      },
    },
  },
]
```

This plugin only initiates the tag manager *container*. If you want to use Google Analytics, please also add `gatsby-plugin-google-analytics`.

If you want to link analytics use with anything inside the container (for example, a cookie consent manager such as OneTrust), you will need to ensure that the tag manager script comes *before* the analytics script in your `gatsby-config.js`.

Tracking routes

This plugin will fire a new event called `gatsby-route-change` (or as in the `gatsby-config.js` configured `routeChangeEventName`) whenever a route is changed in your Gatsby application. To record this in Google Tag Manager, we will need to add a trigger to the desired tag to listen for the event:

1. Visit the Google Tag Manager console and click on the workspace for your site.
2. Navigate to the desired tag using the 'Tags' tab of the menu on the right hand side.
3. Under "Triggering", click the pencil icon, then the "+" button to add a new trigger.
4. In the "Choose a trigger" window, click on the "+" button again.
5. Choose the trigger type by clicking the pencil button and clicking "Custom event". For event name, enter `gatsby-route-change` (or as in the `gatsby-config.js` configured `routeChangeEventName`).

This tag will now catch every route change in Gatsby, and you can add Google tag services as you wish to it.

Tracking Core Web Vitals

Optimizing for the quality of user experience is key to the long-term success of any site on the web. Capturing Real user metrics (RUM) helps you understand the experience of your user/customer. By setting `enableWebVitalsTracking` to `true`, GTM will get "core-web-vitals" events with their values.

You can save this data in Google Analytics or any database of your choosing.

We send three metrics:

- **Largest Contentful Paint (LCP)**: measures loading performance. To provide a good user experience, LCP should occur within 2.5 seconds of when the page first starts loading.

- **First Input Delay (FID)**: measures interactivity. To provide a good user experience, pages should have a FID of 100 milliseconds or less.
- **Cumulative Layout Shift (CLS)**: measures visual stability. To provide a good user experience, pages should maintain a CLS of 0.1. or less.

Note

Out of the box this plugin will simply load Google Tag Manager on the initial page/app load. It's up to you to fire tags based on changes in your app. See the above "Tracking routes" section for an example.