gatsby-plugin-guess-js

Gatsby plugin for integrating Guess.js with Gatsby.

Guess.js is a library for enabling data-driven user-experiences on the web.

When this plugin is added to your site, it will automatically download Google Analytics data and use this to create a model to predict which page a user is most likely to visit from a given page.

The plugin uses this information to do two things.

- When generating HTML pages, it automatically adds `<link prefetch>` for resources on pages the user is likely to visit. This means that as soon as a person visits your site, their browser will immediately start downloading in the background code and data for links they'll likely click on which can dramatically improve the site performance.
- Once the person has loaded your site and visits additional pages, the plugin will continue to predict which pages will be visited and prefetch their resources as well.

Demo

https://guess-gatsby-wikipedia-demo.firebaseapp.com

Install

```
npm install gatsby-plugin-guess-js
```

How to use

```
// In your gatsby-config.js
module.exports = {
  plugins: [
    {
      resolve: "gatsby-plugin-guess-js",
      options: {
        // Find the view id in the GA admin in a section labeled "views"
        GAViewID: `VIEW_ID`,
        // Add a JWT to get data from GA
        jwt: {
          client_email: `GOOGLE_SERVICE_ACCOUNT_EMAIL`,
          private_key: `GOOGLE_SERVICE_ACCOUNT_PRIVATE_KEY`,
        },
        minimumThreshold: 0.03,
        // The "period" for fetching analytic data.
        period: {
          startDate: new Date("2018-1-1"),
          endDate: new Date(),
        },
      },
    },
  ],
}
```

Integrating with CI

Integrating this plugin within a CI pipeline may cause errors because the plugin will prompt the user/machine to log into the Google Analytics account. To get around this you'll need to generate a JWT for your GA account to pass to the plugin, this will enable the plugin to work in prod without any human interaction to authenticate permissions

```
// In your gatsby-config.js
module.exports = {
  plugins: [
    {
      resolve: "gatsby-plugin-guess-js",
      options: {
        // Find the view id in the GA admin in a section labeled "views"
        GAViewID: `VIEW_ID`,
        minimumThreshold: 0.03,
        // Set Google Analytics jwt with Google Service Account email and private key
        jwt: {
          client_email: `GOOGLE_SERVICE_ACCOUNT_EMAIL`,
          private_key: `GOOGLE_SERVICE_ACCOUNT_PRIVATE_KEY`,
        },
        // The "period" for fetching analytic data.
        period: {
          startDate: new Date("2018-1-1"),
          endDate: new Date(),
        },
      },
    },
  ],
}
```

How to get a JWT token?

> Excerpt from https://developers.google.com/identity/protocols/oauth2/service-account

1. Open the Google Developer Console IAM & Admin
2. If prompted, select a project, or create a new one.
3. Click add **service account**.

4. on top click **Create service account**, type a name, ID, and description for the service account, then click **Create**.
5. Optional: Under **Service account permissions**, select the IAM roles to grant to the service account, then click **Continue**.
6. Optional: Under **Grant users access to this service account**, add the users or groups that are allowed to use and manage the service account.
7. Click on **three dots under actions** , then click **Create**.

- Put that file into a directory `node_modules` that is inside one of the parent directories of the script that we'll create later. That means that you can keep it out of the repository with `analytics.js`. For example, the following path is perfectly fine: `\$HOME/node_modules/myproject-3126e4caac6a.json`
  - The credentials that you created have an `client_email` address (which is displayed in the user interface and stored inside the JSON file). Copy that email address.

Go to the Admin panel in Google Analytics:

- Analytics has three scopes:
  - Account
  - Property
  - View
- Create a new user in scope "Property", via "Admin -> View User Management -> Plus icon -> Add User".
  - That user has the email address that you copied previously from the json file `client_email`.
  - Its permissions are "Read & Analyze".
- In scope "View", go to "View Settings" and write down the "View ID" (e.g. 97675673) for later.

Note: This JWT will only be valid for a finite amount of time. If you would like to generate a token that will not expire, please follow further instructions in https://2ality.com/2015/10/google-analytics-api.html

Troubleshooting

- "User does not have any Google Analytics Account": StackOverflow answer