This guide walks you through the steps involved in setting up a Gatsby site that fetches content from Contentstack, a popular headless (or API-first) CMS.

Why use Contentstack with Gatsby?

Contentstack is a headless CMS that offers developers turnkey Content-as-a-Service (CaaS). It lets you create flexible content components and deliver them to any destination platform via CDN-backed Content Delivery APIs. Instead of having to create separate content for all your sites and apps, Contentstack lets you reuse content across different digital channels (web, mobile, IoT, voice, etc.) and present it optimized for its digital endpoint, be it a browser on a laptop, a smartphone, a smartwatch, a kiosk, or a jumbotron.

Gatsby, with its powerful UI toolset, serves as the "head" for your websites while Contentstack – being the headless CMS – acts as the "body". Combining these two best-in-class tools provide a powerful solution to create modern, enterprise-grade sites and apps.

Note: *This guide uses the `gatsby-source-contentstack` plugin to create a Gatsby-powered site.*

Build a site with Contentstack and Gatsby

Step 1: Clone the starter repo and install dependencies

Pull down the Contentstack starter with this command:

```
gatsby new gatsby-starter-contentstack
```

This downloads the required files and initializes the site.

Step 2: Create env file(s) with stack configuration

Create copies of the `env.sample` file for `.env.development` , `.env.production` and any other environment, then add the following stack details to each respective file:

```
CONTENTSTACK_API_KEY='apikey'
CONTENTSTACK_ACCESS_TOKEN='deliverytoken'
CONTENTSTACK_ENVIRONMENT='development' # or production, staging, etc.
```

These will be pulled into the `gatsby-config.js` file under the `gatsby-source-contentstack` plugin.

Note: *if you wish to preview the pages of your Gatsby site from within the Contentstack entry editor, ensure base URLs are set for each environment and language.*

Step 3: Create and publish content

To build a sample home page, perform the following steps in Contentstack:

1. Create a 'Home' content type with 'Title' and 'Body' fields
2. Create an entry for the 'Home' content type
3. Create a development environment
4. Publish the 'Home' entry to the development environment

Step 4: Run Gatsby

Navigate to your root directory and run the following command:

```
gatsby develop
```

After running this, you will be able to view your site at `http://localhost:8000/`. You can run the GraphiQL IDE at `http://localhost:8000/___graphql`. The GraphiQL IDE will help you explore the app's data, including the Contentstack APIs.

Now, you will be able to query Contentstack data. Try the query below to get the 'Home' content type data:

```
{
  contentstackHome {
    title
    body
  }
}
```

Step 5: Create a page in Gatsby

Go to the pages folder inside the src folder, and create a home.js file. Add the code below to it.

```
import React from "react"
import { graphql } from "gatsby"

export default function Home({ data }) {
  return (
    <div>
      <h1>{data.contentstackHome.title}</h1>
    </div>
  )
}

export const pageQuery = graphql`
  query HomeQuery {
    contentstackHome {
      title
    }
  }
`
```

This will display the title of your home page on `http://localhost:8000/home`. Likewise, you can query additional fields in your entry.

Closing note

The above example is meant to demonstrate how to set up a Gatsby site that sources content directly from Contentstack. Contentstack has also published a few articles that demonstrate how Contentstack works with Gatsby in their documentation:

- Getting started with Contentstack and Gatsby
- Build an example website using Gatsby and Contentstack