Seams-CMS is a headless CMS system in which you can manage everything from blog posts to complex data structures like products and translations. After defining your data types, you can add content based on your specific structures. Seams-CMS allows you to localize your content and has an advanced security and workflow model. Your content can be retrieved through different JSON APIs and GraphQL endpoints, which allows you to use your content as you see fit.

Seams-CMS makes it possible use your content in different scenarios: on your website, in your blogs, and in your apps while managing your content in a single location.

Creating a Seams-CMS account is free for hobby or small sites. The code samples in this guide assume you're using the demo workspace that is automatically generated when you create a new account. Make sure you add an API key to this workspace in the API key settings. You will need your workspace ID and the (read-only) API key shortly.

Setup

Create a new Gatsby site with the default starter and install the source plugin.

Run this in your terminal:

```
gatsby new seamscms-site
cd seamscms-site
npm install seams-cms-gatsby-source
```

Configuring the source plugin

Once you have installed the plugin, it's time to configure it. Add the following code to the `gatsby-config.js` file found in the root of your project.

```
module.exports = {
  ...
  plugins: [
    ...
    {
      resolve: 'seams-cms-gatsby-source',
      options: {
        workspace: <WORKSPACE_ID>,
        apiKey: <API_KEY>,
        contentTypes: ['blogpost'],
      },
    }
    ...
  ],
  ...
}
```

Change `<WORKSPACE_ID>` and `<API_KEY>` into the ID of the workspace from your own Seams-CMS account and the newly generated API key inside that workspace. Within the `contentTypes` array, you can list all the IDs of the content types you'd like to import from Seams-CMS into your Gatsby site. For this tutorial, you'll only want to import the `blogpost` content.

Checking your connection

Run `gatsby develop` to fetch your blog posts from Seams-CMS:

```
gatsby develop
```

To check if it worked, you can check the GraphQL playground at `http://localhost:8000/___graphql` and look for an `allSeamsCmsBlogpost` section. If found, it means that your blog posts are correctly fetched and stored in the GraphQL layer of Gatsby, ready to be consumed.

To see what kind of data it holds, try running the following query:

```
query MyQuery {
  allSeamsCmsBlogpost {
    edges {
      node {
        content {
          title {
            value
          }
          content {
            value
          }
          author {
            value {
              content {
                name {
                  value
                }
              }
            }
          }
          categories {
            value {
              content {
                name {
                  value
                }
              }
            }
          }
        }
      }
    }
  }
}
```

This will return the title, content, author name, and categories for each blog post.

Generating a blog post page

You can create a site with a single page holding all blog posts. You can do this by creating a template blog index file at `src/templates/blog-index.js` with the following content:

```
import React from "react"

import Layout from "../components/layout"

const BlogIndex = ({ pageContext: { blogPosts } }) => (
  <Layout>
    {blogPosts.map(blogPost => (
      <>
        <h4>Title: {blogPost.node.content.title.value}</h4>
        <h4>
          Author: {blogPost.node.content.author.value[0].content.name.value}
        </h4>
        <p>{blogPost.node.content.content.value}</p>
        <hr />
      </>
    ))}
  </Layout>
)

export default BlogIndex
```

This template fetches your blog posts from the `pageContext` (passing your data along with this variable), and iterates over all the posts. For each post, it prints the title, the author's name, the content, and whatever you'd like to add that is present in the GraphQL data. For instance, you might also fetch things like tags, categories, snippets, image headers, etc.

Once you have created your template, connect your data to it! Check out the `gatsby-node.js` file found in your root directory. Note that you will need a GraphQL query that you can generate automatically through the web interface. Select all the fields you want to use in your template and copy/paste this query into the new file:

```
const blogQuery = `
   query MyQuery {
     allSeamsCmsBlogpost {
       edges {
         node {
           content {
             title {
               value
             }
             content {
               value
             }
             author {
               value {
                 content {
                   name {
                     value
```

```
                    }
                  }
                }
              }
              categories {
                value {
                  content {
                    name {
                      value
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
`
```

You're almost there! Next, create and export a `createPages` function in the same `gatsby-node.js` file that will create your page:

```
exports.createPages = async ({ graphql, actions: { createPage } }) => {
  const query = await graphql(blogQuery)

  createPage({
    path: `/blogs`,
    component: require.resolve("./src/templates/blog-index.js"),
    context: {
      blogPosts: query.data.allSeamsCmsBlogpost.edges,
    },
  })
}
```

Finally, when you run `gatsby develop` again, it should generate a `/blogs` page with your content. You can visit this at `http://localhost:8000/blogs`.

Other resources

For more detailed information about setting up Seams-CMS with Gatsby, see the Seams-CMS blog.