Sub-org Name

Plone

Student Info

Name : Alok Kumar
Github-username : iFlameing
Email : ialokkumarsingh0@gmail.com
Telephone : +91 8082805834
Country : India
Time-zone : UTC +5:30
Primary Language : English

Code Samples

I have contributed to the gatsby-source-plugin and implemented fetching image from Plone CMS which is in private mode, if user passes token in configuration of gatsby-source-plone plugin. I also refactored the code of gatsby-starter-blog into gatsby-theme-plone and POC of gatsby-theme-plone is completed. Looking forward to complete that project also.

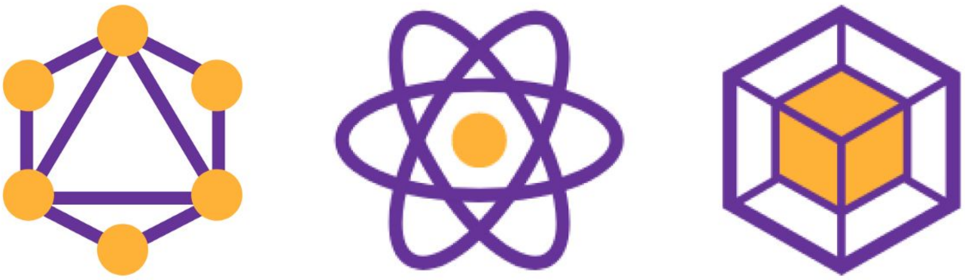
- Fetching private image from plone site (merged)
    - https://github.com/collective/gatsby-source-plone/pull/165
- Refactored gatsby-starter-blog into gatsby-theme-plone(it is also a preliminary idea for gsoc. POC is completed)
    - https://github.com/collective/gatsby-theme-plone
- You can see all commits regarding the gatsby-theme-plone at
    - https://github.com/collective/gatsby-theme-plone/commits?author=iFlameing
- Created example-repository for gatsby-theme-plone
    - https://github.com/iFlameing/gatsby-theme-plone-Example-website

- Released the npm for gatsby-theme-plone
    - https://www.npmjs.com/package/gatsby-theme-plone

- Adding header to createRemoteFileNode(review process)
    - https://github.com/gatsbyjs/gatsby/pull/11682

What is Gatsby?

It’s another static site generator like Hugo, Jekyll and so on. So what makes it special?
Gatsby can be used to build static sites from any source(since 2018 also from Plone CMS) that are Progressive Web Apps, follow the latest web standards, and are optimized to be highly performant. It makes use of the latest and popular technologies including ReactJS, Webpack, GraphQL, modern ES6+ JavaScript and CSS.



GraphQL + React + Webpack = ❤

Project Info

GatsbyJs Preview for plone

Currently gatsby-source-plone plugin is unable to update,update, delete and create node after fetching data from plone site unless GatsbyJs development server is manually restarted or GatsbyJs site is rebuilt. This project plan is to enhance existing project "gatsby-source-plone" with features that allow "gatsby develop" development

server to provide up-to-date instant live previews of pages that contain data from a Plone CMS source. Plone is an enterprise open-source CMS solution written in Python. The aim is to enhance "gatsby-source-plone" to provide a GatsbyJS development experience matching the experience provided by "gatsby-source-filesystem" based plugins, which can create, update and delete nodes from filesystem events.
This idea matters because It will add ability to instantly preview how changes in Plone CMS content affect the eventual GatsbyJs built site. Site built with the help of GatsbyJs is considered crucial feature in marketing Plone as CMS backend for GatsbyJS powered web sites.

TechStack

This project requires good knowledge of javascript, GatsbyJs, NodeJs and understanding how gatsby-source-filesystem works under the hood.

Importance of Project

- User don't have to wait for see the changes they have made in database on their site.
- They don't have to build their app continuously to see the changes.
- Enhance developing experience with gatsby-source-plugin.
- This would make gatsby-source-plugin the best available CMS plugin for GatsbyJs.

Implementation

GatsbyJS is a React-based, GraphQL powered static site generator. It creates the fastest web sites possible. But this feature is missing in current gatsby-source-plugin which works as gatsby-source-filesystem plugin for fetching data from Plone CMS. Currently gatsby-source-plugin don't fetch the latest changes from database which is a problem because developer has to rebuild the whole site to see the changes. Let us assume that a developer have 1000 page website and if everytime if he has to rebuild the whole project to see the latest changes can consume lots of time. That's not cool.

The project plan is to enhance the existing project gatsby-source-plone with features that allow the gatsby develop development server to provide up-to-date, instant, live previews of pages with data from a Plone CMS source. This makes it possible to use gatsby develop to provide a preview of the changes and only run the static site rebuild when the author is ready. This would make gatsby-source-plone the best available CMS plugin for GatsbyJS.

Initial solution could simply poll the configured Plone site for every few seconds and smartly update affected GatsbyJS nodes (content nodes and their, breadcrumbs and navigation nodes) and pages. Because this cannot detect deleted pages, the eventual solution requires support for a configurable websocket endpoint where the plugin could listen for updates. During GSOC this websocket endpoint could be simulated with a custom NodeJS implementation. As we are using express-js we can also provide HTTP POST webhook endpoints where current Plone could make a call using https://pypi.org/project/collective.webhook/. We can use this webhook by simply adding egg in our buildout file i.e :

```
[buildout]

...

eggs =
    collective.webhook
```

And then running bin/buildout.

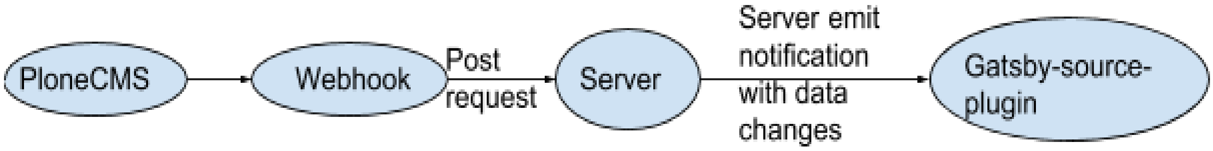
This way we could configure Plone site to notify gatsby-source-plone via ExpressJs-webhook. Already(Plone has feature called “content rules” that allow configuring automated actions to be triggered on events like add, change and delete. Collective.webhook add-on adds webhook calling support for it).

Phase 1

Dependencies used

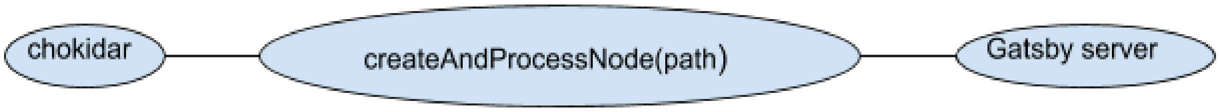
- Express Js
- SocketIo

In this phase we mainly work on our server side that is implementing mockup server which provide services to Plone CMS and gatsby-source-plugin. We will create server with the help of express(a popular node framework which is used for creating server and custom endpoints). For notifying changes to gatsby-source-plugin we are using socketIo(*Socket.IO* is a JavaScript library for realtime web applications. It enables real time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js). When something changes in Plone CMS the webhook make a http post request with notification of event which is occured to the server. When server receive this http request it emits notification to gatsby-source-plugin with the event which is occurred. The gatsby-source-plugin handle this notification and changes the data which is needed to change based on request.



Phase 2

Now as we configure the initial setup. Now we are good to implement the gatsby api for adding changed data. In this phase we generally implement the api related to add, update, delete api. As this feature is same as the feature provided by gatsby-source-filesystem. So, understanding the gatsby-source-filesystem is more important than anything. Currently gatsby-source-filesystem handle these changes by using a npm library called chokidar and gatsby api(**createAndProcessNode()**). Chokidar library is used for getting file change events. After that gatsby api register these chokidar events at source node and pushes changes to gatsby development server. The simple implementation is shown below and code is in this link. We also have to implement the same feature by using same api but instead of chokidar we have socketIo endpoint.



Phase 3

Now as we configure the endpoints at both server side, client side and we also understand how gatsby-source-filesystem handle all the changes gracefully, and completed the POC of this project. In this phase we mainly work for remaining configuration of the gatsby-source-plone plugin with our mockup server and updating the node which is related to endpoint. We also have to solve all the issues which will be arising in second phase and this phase. Refactoring the code to future-proof. Writing documentation and test which is left for all modification we have done in the gsoc period.

Timeline:

Mainly Gsoc have three phases and also have a period of community bonding in addition.
So, As implementation is described above, I classified the above implementation in community bonding and the three phases.

Community Bonding Period (May 7 - May 26)

- Interact with the members of the community.
- Discuss on the design system of project.
- Clear all the doubts, and concepts regarding to my project which helps me in coding period.
- Finalizing all the technology stack going to be used.
- Learning SocketIo which helps me in implementing the CRUD feature in gatsby-source-plugin.
- Discovering api which is used in the gatsby-source-filesystem
- Read the code of gatsby-source-plone

Coding Period Begins :

Phase 1 (May 27 - June 24)

May 27 to June 6 (09 days)

- Start Implementing the nodeJs mockup server.
- I will dig deep into how we can implement all the APIs in efficient way.
- Get myself comfortable with packages like socketIo.
- Configuring socketIo with NodeJs

June 6 to June 14 (09 days)

- Creating endpoint for change, add and deletion.
- Reading about the discovered api of gatsby which is used in this project.
- Understanding how gatsby-source-file system handle changes.

June 15 to June 24(10 days)

- Configuring gatsby-source-plone with socketIo-client.
- Implementing websocket endpoint for add, change, delete.
- Configuring webhook to Plone CMS.

First Evaluation

Phase 2 (June 28 - July 24)

June 28 to July 6 (09 days)

- Implementing add api to gatsby-source-plone.
- When we add something to Plone CMS we can instantly trigger gatsby-source-plone to fetch the added data and instantly show the preview.
- Adding test for the added changes
- Writing documentation for this api

July 8 to July 16(09 days)

- Implementing change api to gatsby-source-plone
- When we modify data to Plone CMS, we again fetch the modify data and show the preview
- Adding test for the added api
- Writing documentation for this api
- Start implementing delete api to gatsby-source-plone

July 17 to July 24(08 days)

- Fully implemented delete api to gatsby-source-plone.
- Solving all issue related to all implemented api.
- Complete Proof of concept before the second evaluation period.
- Test all the implementation is working fine or not.

Second Evaluation

Phase 3 (July 27 - August 19)

July 27 to August 3(09 days)

- Writing Documentation for all the changes made to repo.
- Writing all the test related to all the changes.
- Refactoring code for future proof.

July 5 to August 13(09 days)

- Make a example site using all these feature.
- Writing blog post related to all these changes.

July 14 to August 19(06 days)

- Finish up all the pending issue, documentation and test related to repo.

Final Evaluation

After GSOC

After GSOC I will keep contributing to plone organisation. I already completed the POC of gatsby-theme-plone. I want to complete this project as well and become the core contributor of the plone organisation. I am also interested to contribute in volto project which is based on reactJs.

What is the most interesting software project you've worked on while in school?

I have worked on many important projects in university. What’s really crucial for me when starting one is to get very clear on the goals right at start and then create a plan with milestones. For example I have created a website called “React-app” for learning react and firebase. It took me around a month to learn and deploy it in production. I am also creating a clone of twitter with chat as additional feature using technology node, express, mongodb, jwt and react as frontend. It is in developing phase and now I changed my mind to refactor react code to react-hooks code before deploy it in production.

- React app : https://my-first-react-app-84542.firebaseapp.com/
- Tweet-chat-app : https://github.com/iFlameing/Tweet-Chat-App

---

Have you ever contributed to an open source project? How?

Yes, I have contributed to open source project in past. I have implemented autosave feature, refactoring code of github oAuth in Iodide mozilla project. I also contributed to taskcluster which is another mozilla project. I have implemented a signIn button in app bar with all feature. I also contributed to gatsby-source-plone and refactored gatsby-starter-blog into gatsby-theme-plone.

- In Iodide I commited 13 commits. You can see all commit at
    - https://github.com/iodide-project/iodide/commits?author=iFlameing

- In taskcluster I commited 7 commits and one approved pull request. You can see all commit at
    - https://github.com/taskcluster/taskcluster/commits?author=iFlameing

---

Have you used Plone before? What for?

No, I have not used Plone before. But for creating gatsby-theme-plone I have created my own local site using plone.

Why do you want to do this project?

I want to this project because I’ve always loved building and configuring apps. I am excited to see there’s a lot of opportunity to use my testing and configuring skills in this project. I think I am the most suitable candidate for this project because I have experience with gatsby, node, react and with gatsby-source-plone. Using gatsby-source-plone I made gatsby-theme-plone.

Apart from this, I am pretty familiar with the workflow which is followed by Plone and have enough experience with git and Web development. I am pretty proficient in javascript, gatsby, node and react which are the required skills for this project.

Benefits I will be getting from this project :

- Will get to work with really great people.
- Get to know more about Plone itself.
- A lifetime experience.
- Establish myself as a core contributor!

Other Commitments

- **Do you have any other commitments during the main GSoC time period?**
    - No, I don’t have any commitments during the GSOC period.

- **Have you applied with any other organizations?**
    - No, I am only applying to Plone.

- During the entire GSOC period, I am willing to devote at least 40 hrs per week to my project.

Extra Information:

University info:

- **University name:** Shri Mata Vaishno Devi University, Katra, Jammu and kashmir
- **Major:** Computer Science & Engineering
- **Current year:** 2nd Year
- **Expected graduation date:** August 2021
- **Degree:** Bachelor of Technology (B.Tech)