FrontConf 2k19 — Munich — 27/04/2019

Fast by Default: Extending GatsbyJS with Plugins

& /phacks

👋

Hi! My name is Nicolas Goutay. I am a Web Performance Evangelist at Theodo. I love building side-projects, and I'm a top GatsbyJS contributor. I ~~have stage fright~~ am really excited to be with all of you today! ☺️

You can find me on Twitter/GitHub at @phacks.

What is GatsbyJS?

What is GatsbyJS?

Static Site Generators (SSGs)

Jekyll, Hugo, Middleman

😎 Build performant websites

📟 Little to no backend

🎨 Off-the-shelf themes

Progressive Web Apps (PWAs)

React, Vue, Angular

📱 App-like, snappy experience

🏗 Easily reuse code & components

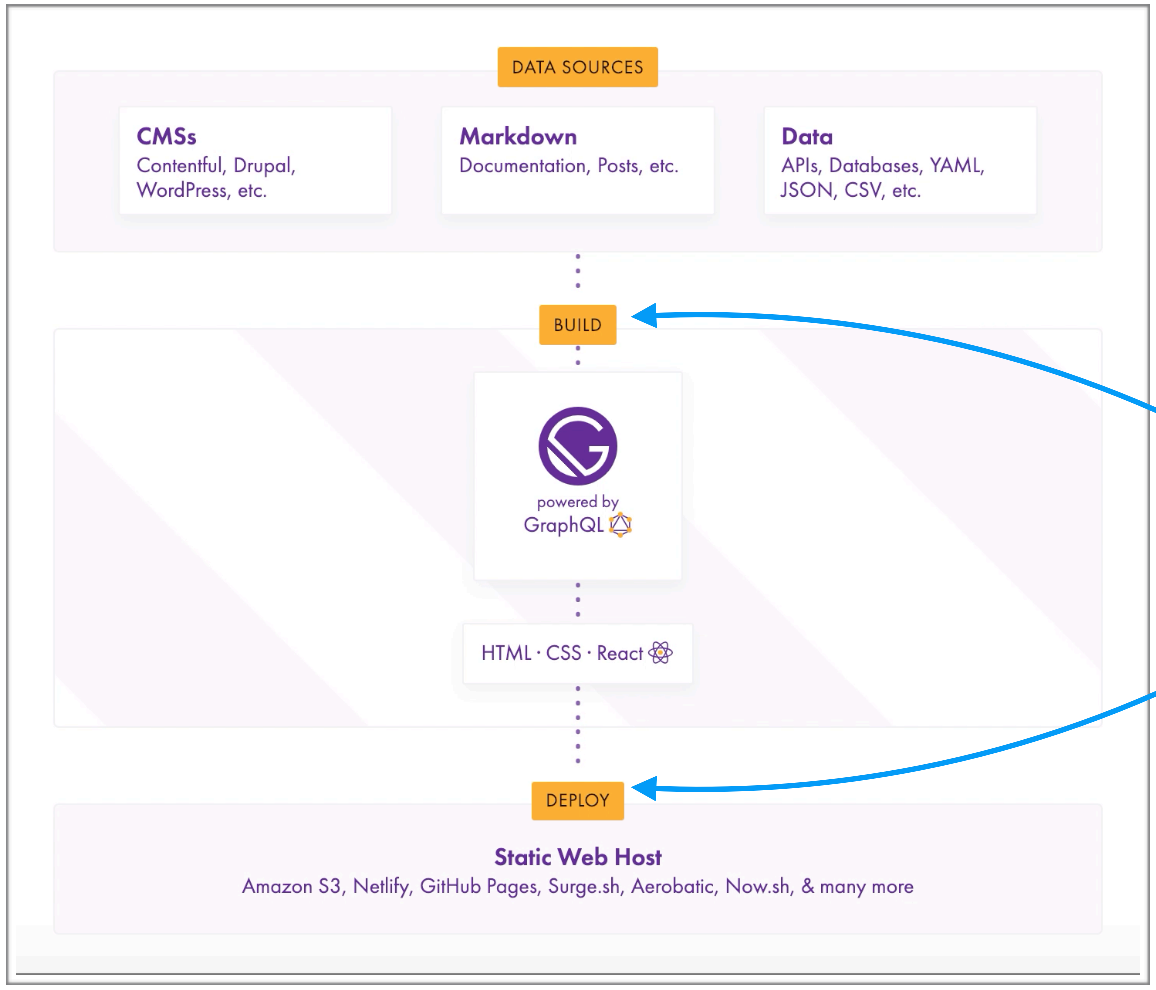
✨ Build interactive websites

What is GatsbyJS?

Static Progressive Web Apps

- 😎 Build performant websites
- 📟 Little to no backend
- 🎨 Off-the-shelf themes
- 📱 App-like, snappy experience
- 🏗 Easily reuse code & components
- ✨ Build interactive websites

What is GatsbyJS?

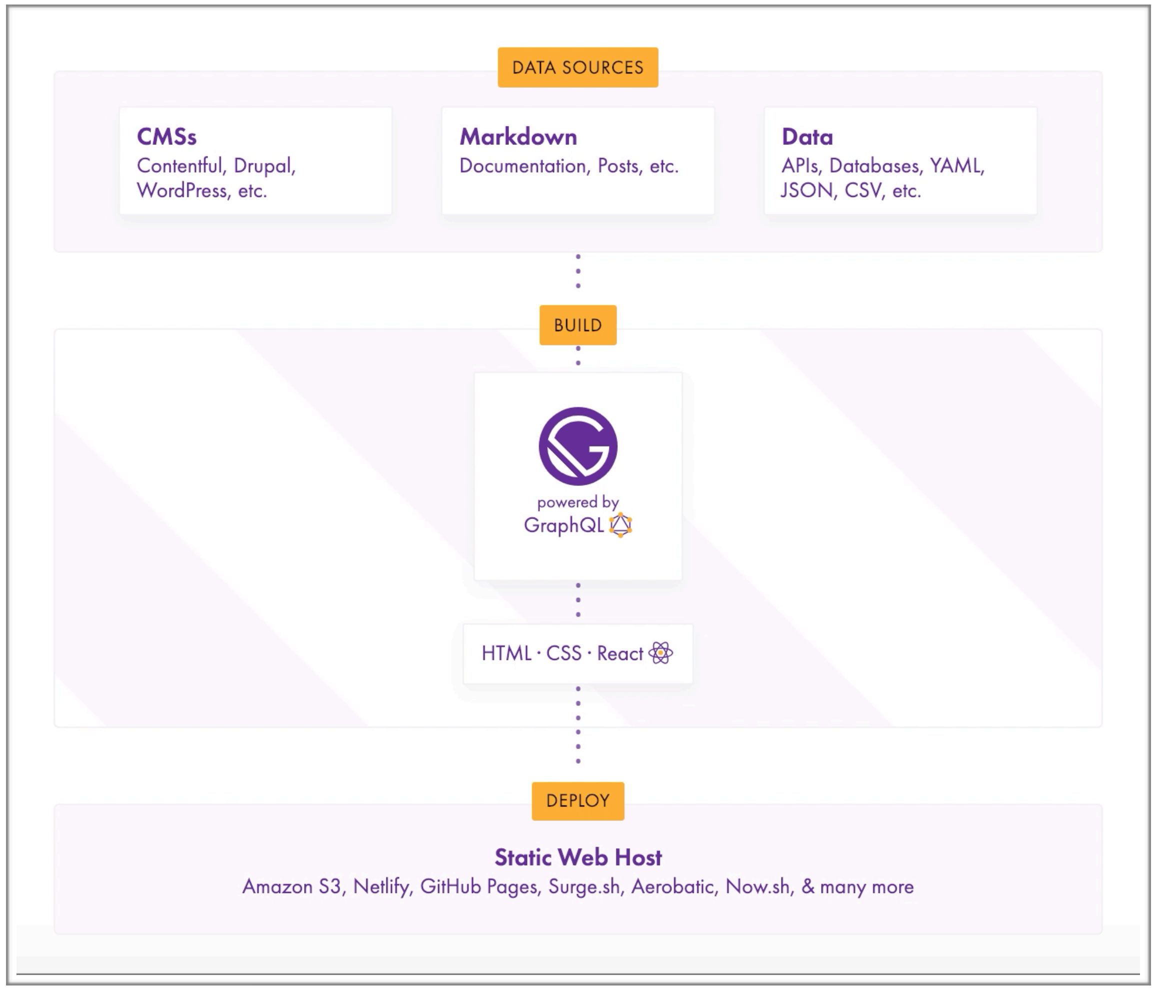

DATA SOURCES
CMSs
Contentful, Drupal, WordPress, etc.
Markdown
Documentation, Posts, etc.
Data
APIs, Databases, YAML, JSON, CSV, etc.
BUILD
powered by GraphQL
HTML · CSS · React
DEPLOY
Static Web Host
Amazon S3, Netlify, GitHub Pages, Surge.sh, Aerobatic, Now.sh, & many more

The SSG part relies on NodeJS and GraphQL

The SPA part is a fully-featured React application

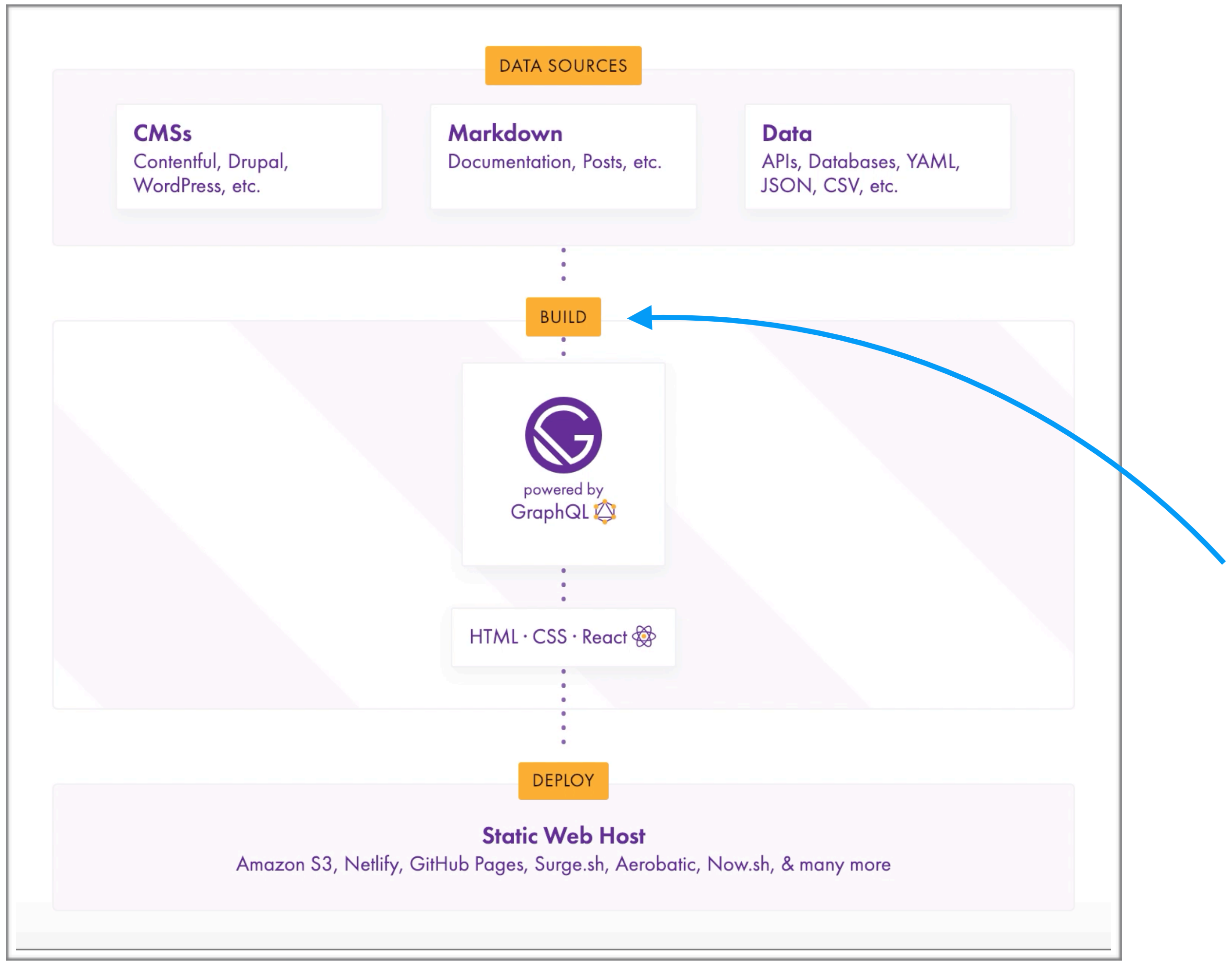
Gatsby Plugins

Gatsby Plugins


DATA SOURCES
CMSs
Contentful, Drupal, WordPress, etc.
Markdown
Documentation, Posts, etc.
Data
APIs, Databases, YAML, JSON, CSV, etc.
BUILD
powered by GraphQL
HTML · CSS · React
DEPLOY
Static Web Host
Amazon S3, Netlify, GitHub Pages, Surge.sh, Aerobatic, Now.sh, & many more

There are two ways to inject external data into a Gatsby website:

Gatsby Plugins

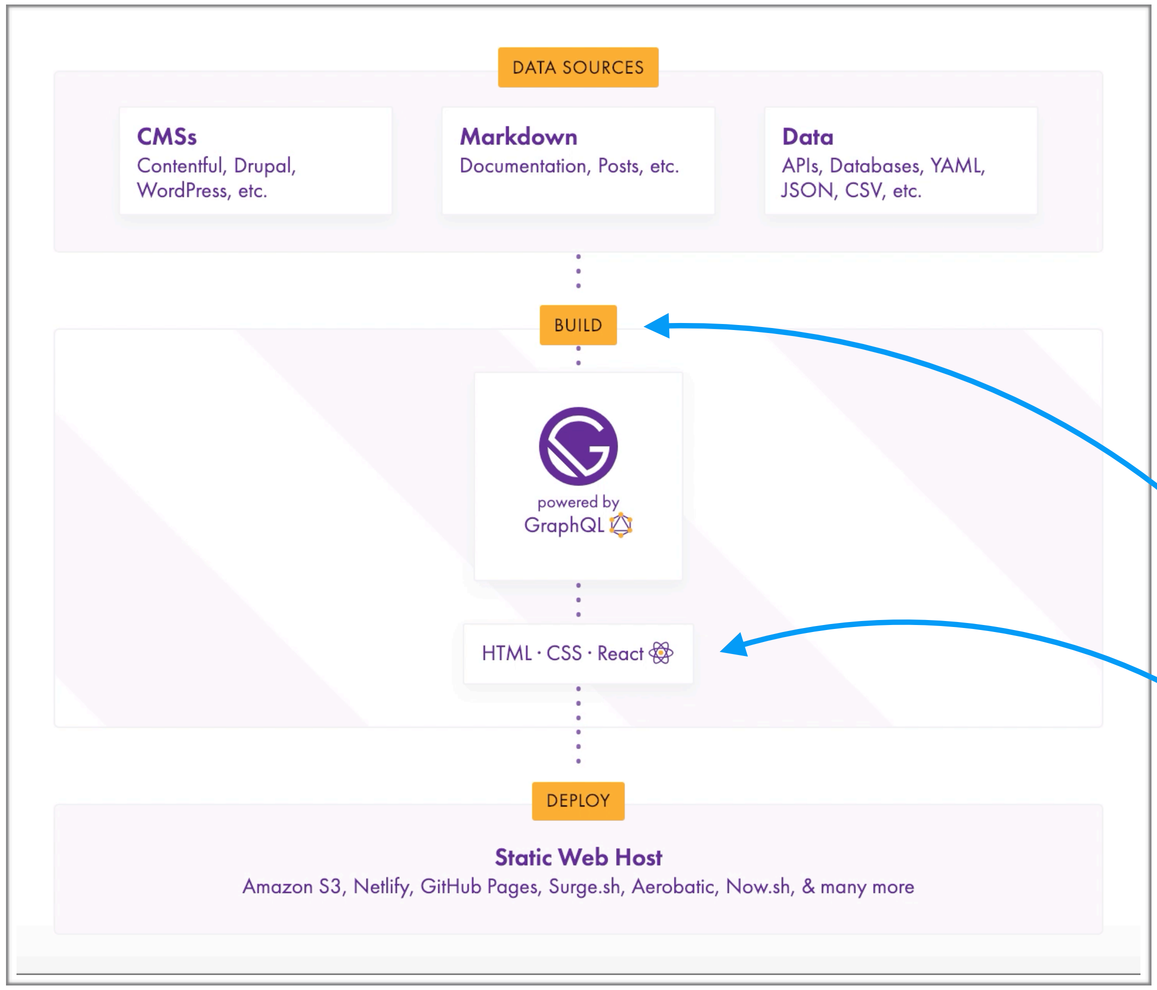

DATA SOURCES
CMSs
Contentful, Drupal, WordPress, etc.
Markdown
Documentation, Posts, etc.
Data
APIs, Databases, YAML, JSON, CSV, etc.
BUILD
powered by GraphQL
HTML · CSS · React
DEPLOY
Static Web Host
Amazon S3, Netlify, GitHub Pages, Surge.sh, Aerobatic, Now.sh, & many more

There are two ways to inject external data into a Gatsby website:

🛠️ At build-time (GraphQL)

& /phacks

Gatsby Plugins


DATA SOURCES
CMSs
Contentful, Drupal, WordPress, etc.
Markdown
Documentation, Posts, etc.
Data
APIs, Databases, YAML, JSON, CSV, etc.
BUILD
powered by GraphQL
HTML · CSS · React
DEPLOY
Static Web Host
Amazon S3, Netlify, GitHub Pages, Surge.sh, Aerobatic, Now.sh, & many more

There are two ways to inject external data into a Gatsby website:

🛠 At build-time (GraphQL)

🏃‍♀️ At runtime (React, fetch...)

& /phacks

Gatsby Plugins

🛠 **What kind of data are suitable for build time injection?**

Static data (e.g. blog posts) or weakly dynamic data (where one can wait ~1 minute for them to deploy to production, e.g. e-commerce catalog updates).

Gatsby Plugins

🛠 **What kind of data are suitable for build time injection?**

Static data (e.g. blog posts) or weakly dynamic data (where one can wait ~1 minute for them to deploy to production, e.g. e-commerce catalog updates)

🏃‍♀️ **What kind of data are suitable for runtime injection?**

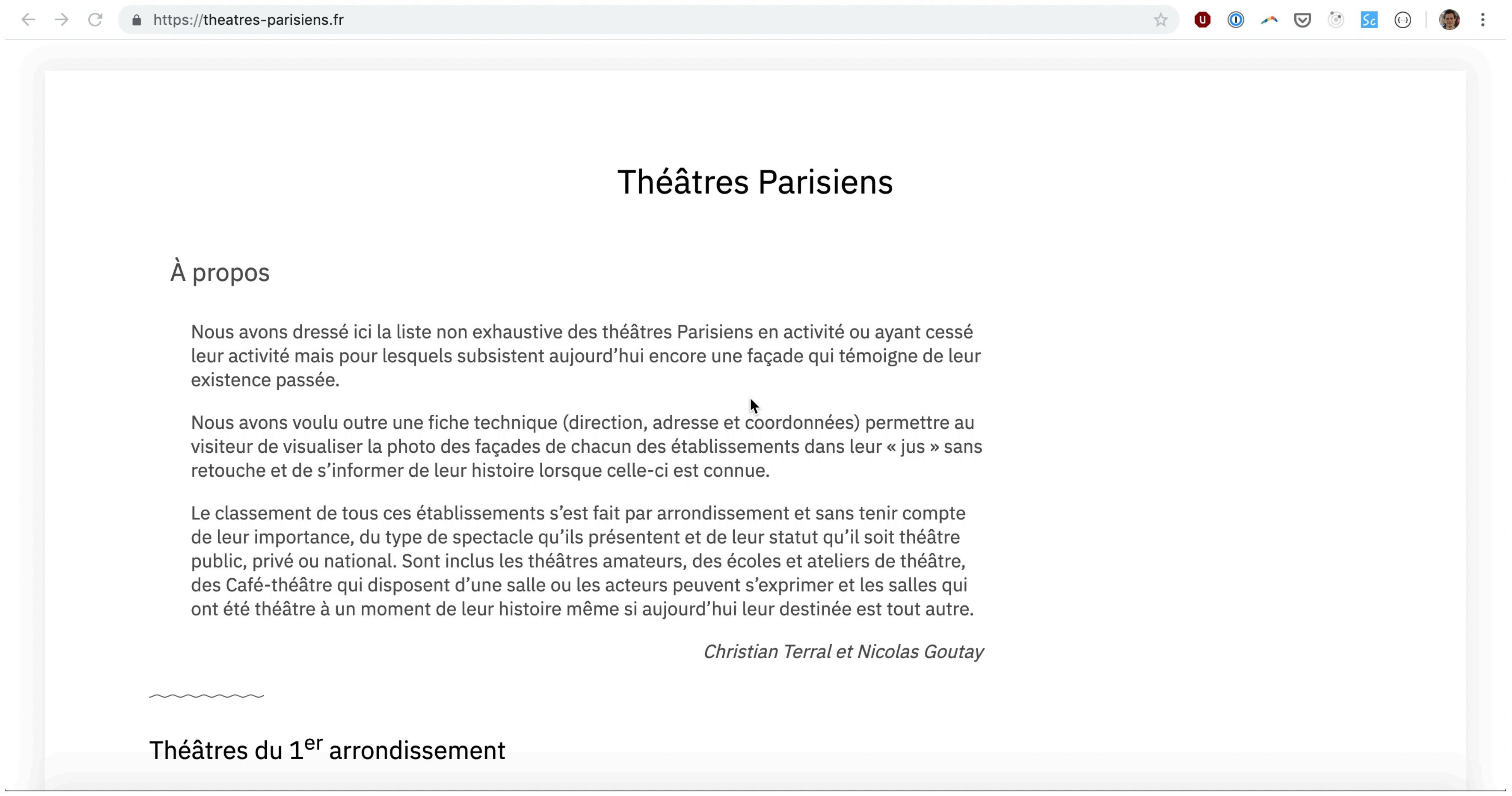
Highly dynamic data (e.g. real-time notifications, data controlled by users).

Gatsby Plugins

Gatsby plugins allows one to inject data from any datasource (remote API, database, filesystem…), through a GraphQL interface, into your Gatsby website's React components.

They abstract the complexity of remote datasources away from the frontend developer.

Gatsby Plugins


https://theatres-parisiens.fr

Théâtres Parisiens

À propos

Nous avons dressé ici la liste non exhaustive des théâtres Parisiens en activité ou ayant cessé leur activité mais pour lesquels subsistent aujourd'hui encore une façade qui témoigne de leur existence passée.

Nous avons voulu outre une fiche technique (direction, adresse et coordonnées) permettre au visiteur de visualiser la photo des façades de chacun des établissements dans leur « jus » sans retouche et de s'informer de leur histoire lorsque celle-ci est connue.

Le classement de tous ces établissements s'est fait par arrondissement et sans tenir compte de leur importance, du type de spectacle qu'ils présentent et de leur statut qu'il soit théâtre public, privé ou national. Sont inclus les théâtres amateurs, des écoles et ateliers de théâtre, des Café-théâtre qui disposent d'une salle ou les acteurs peuvent s'exprimer et les salles qui ont été théâtre à un moment de leur histoire même si aujourd'hui leur destinée est tout autre.

Christian Terral et Nicolas Goutay

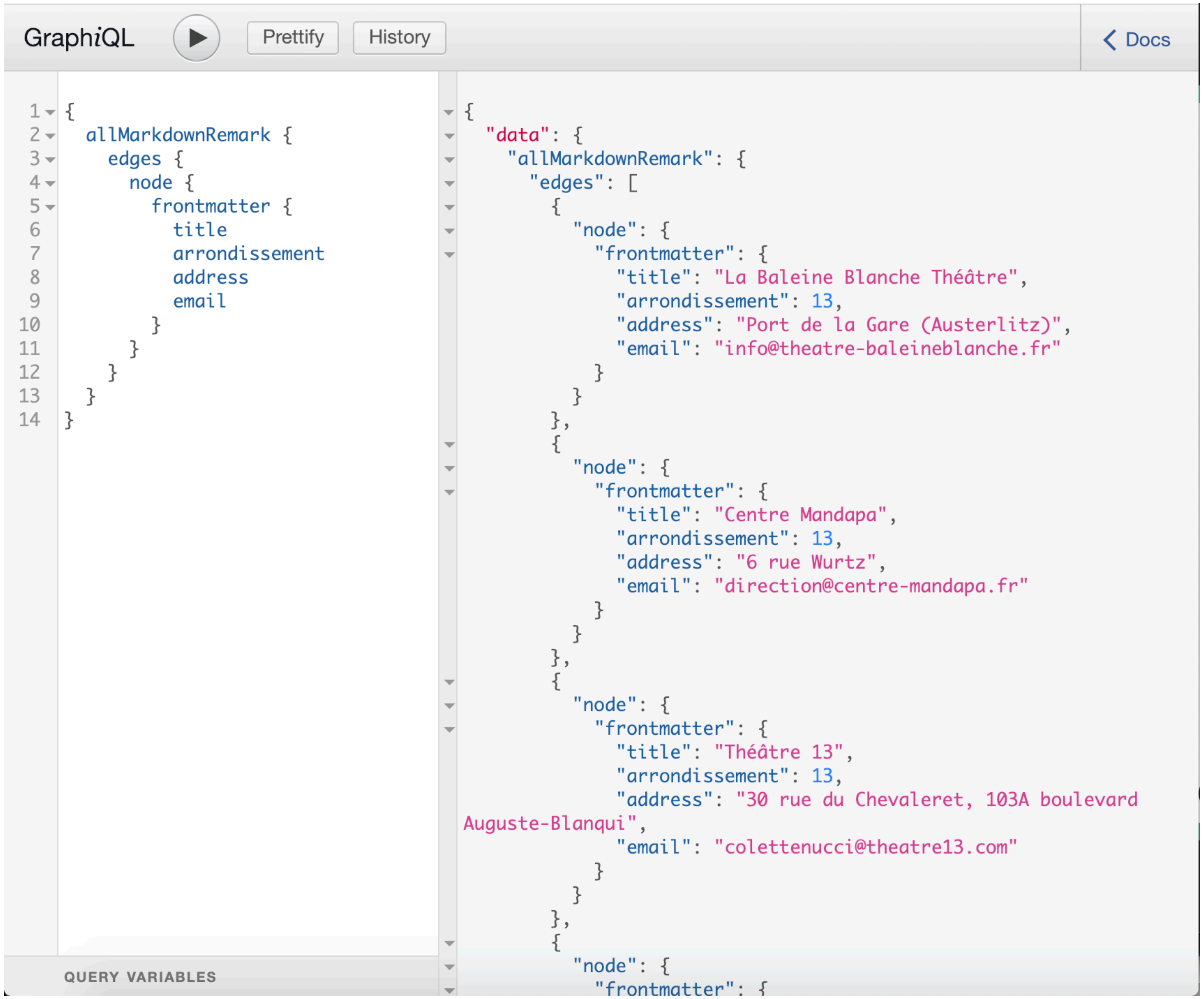
Théâtres du 1er arrondissement

https://theatres-parisiens.fr

https://github.com/phacks/theatres-parisiens

& /phacks

Gatsby Plugins



```
{
  allMarkdownRemark {
    edges {
      node {
        frontmatter {
          title
          arrondissement
          address
          email
        }
      }
    }
  }
}
```

```
{
  "data": {
    "allMarkdownRemark": {
      "edges": [
        {
          "node": {
            "frontmatter": {
              "title": "La Baleine Blanche Théâtre",
              "arrondissement": 13,
              "address": "Port de la Gare (Austerlitz)",
              "email": "info@theatre-baleineblanche.fr"
            }
          }
        },
        {
          "node": {
            "frontmatter": {
              "title": "Centre Mandapa",
              "arrondissement": 13,
              "address": "6 rue Wurtz",
              "email": "direction@centre-mandapa.fr"
            }
          }
        },
        {
          "node": {
            "frontmatter": {
              "title": "Théâtre 13",
              "arrondissement": 13,
              "address": "30 rue du Chevaleret, 103A boulevard
Auguste-Blanqui",
              "email": "colettenucci@theatre13.com"
            }
          }
        },
        {
          "node": {
            "frontmatter": {
```

& /phacks

Gatsby Plugins

WordPress

M

& /phacks

Gatsby Plugins

vimeo

WordPress

GitHub

M

& /phacks

Gatsby Plugins

vimeo

WordPress

GitHub

mongoDB

M

& /phacks

A small story:
From Digital to Analog and Back Again

From Digital to Analog and Back Again

&/phacks

From Digital to Analog and Back Again

СТЕРЕО С10 24467 009
BACH
7 29 109
7 29 022
ETERNA
THE BEATLES
STEREO C10 29771 000 K. SH
СТЕРЕО А10 00605 001 П. ЧАЙКОВ
СТЕРЕО А10 00537 009 Л. ВАН БЕТХОВ
STEREO A10 00641 002 THE
7-29 050

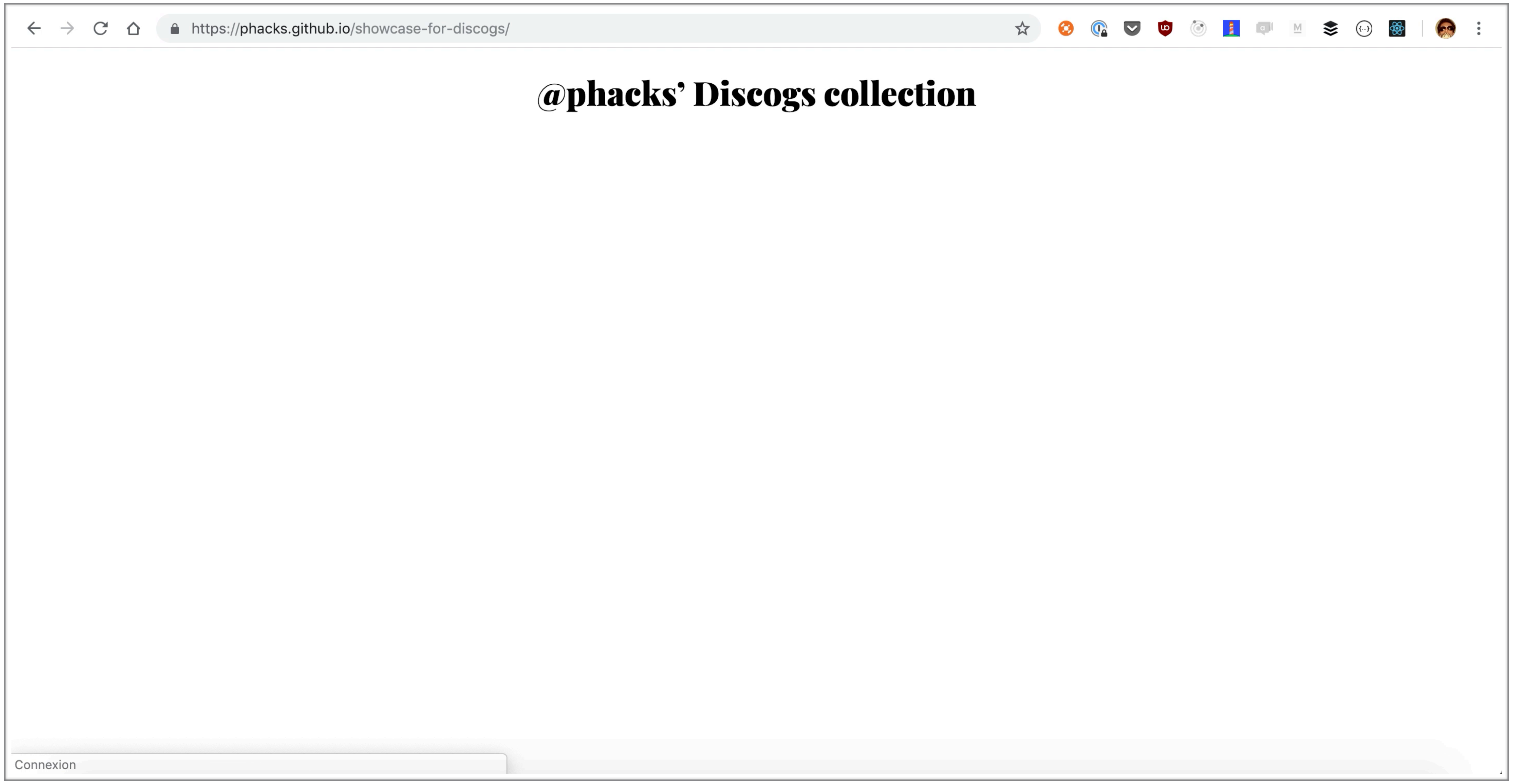
From Digital to Analog and Back Again

&/phacks

From Digital to Analog and Back Again

Discogs®

From Digital to Analog and Back Again



https://phacks.github.io/showcase-for-discogs/

@phacks' Discogs collection

https://phacks.github.io/showcase-for-discogs

https://github.com/phacks/showcase-for-discogs

& /phacks

From Digital to Analog and Back Again

🤔 **Two problems with the naive approach**

- 🐌 Slow loading times as I fetch 60+ resources from the Discogs API

From Digital to Analog and Back Again

🤔 **Two problems with the naive approach**

- 🐌 Slow loading times as I fetch 60+ resources from the Discogs API
- 👆 Hitting the Discogs API rate limit means that only one person can visit the website every minute

From Digital to Analog and Back Again

🤔 **Two problems with the naive approach**

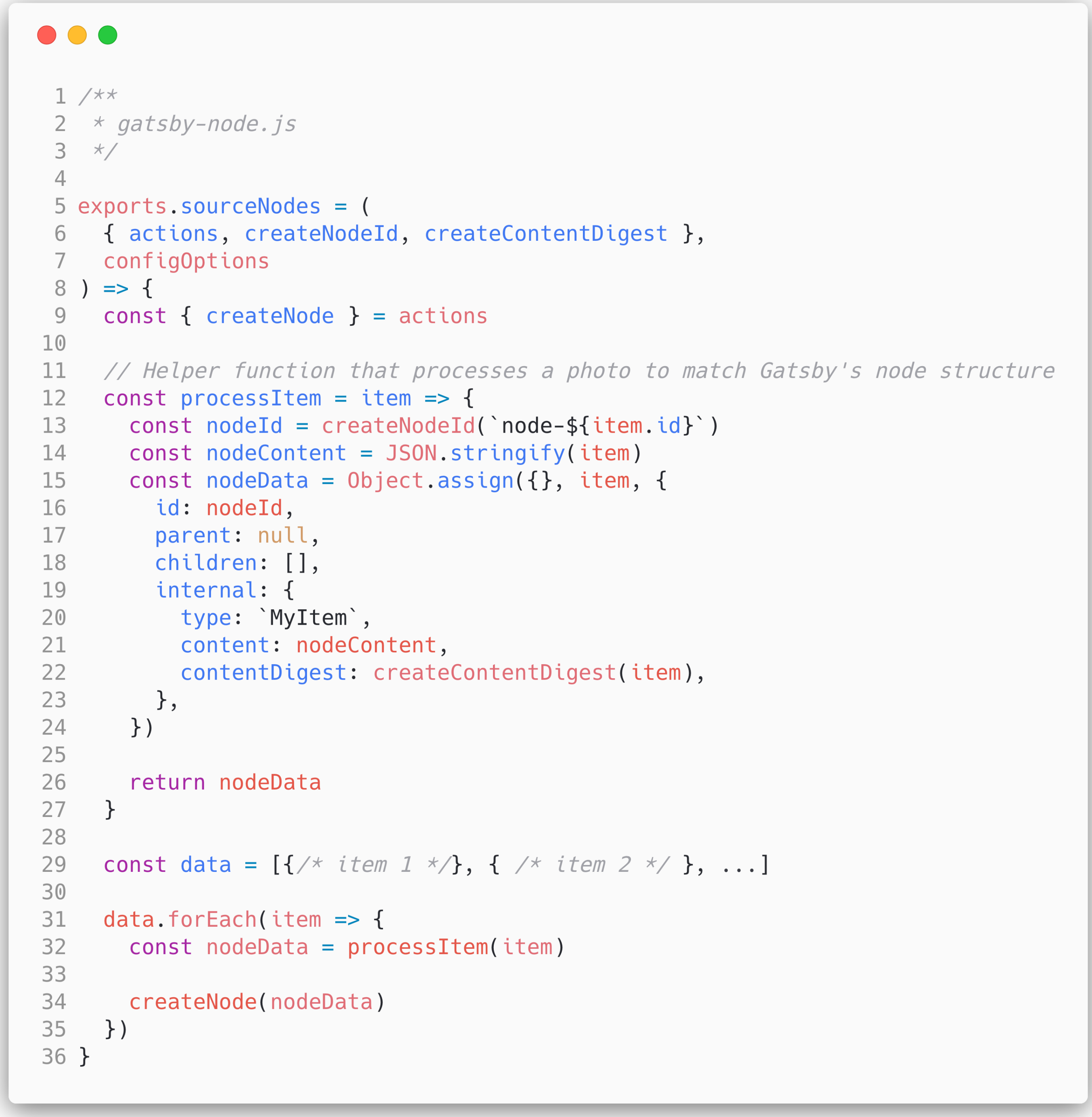
🐌 Slow loading times as I fetch 60+ resources from the Discogs API

👆 Hitting the Discogs API rate limit means that only one person can visit the website every minute

😅 *(I think I invented the litteral opposite of scalability)*

Building a Gatsby Plugin

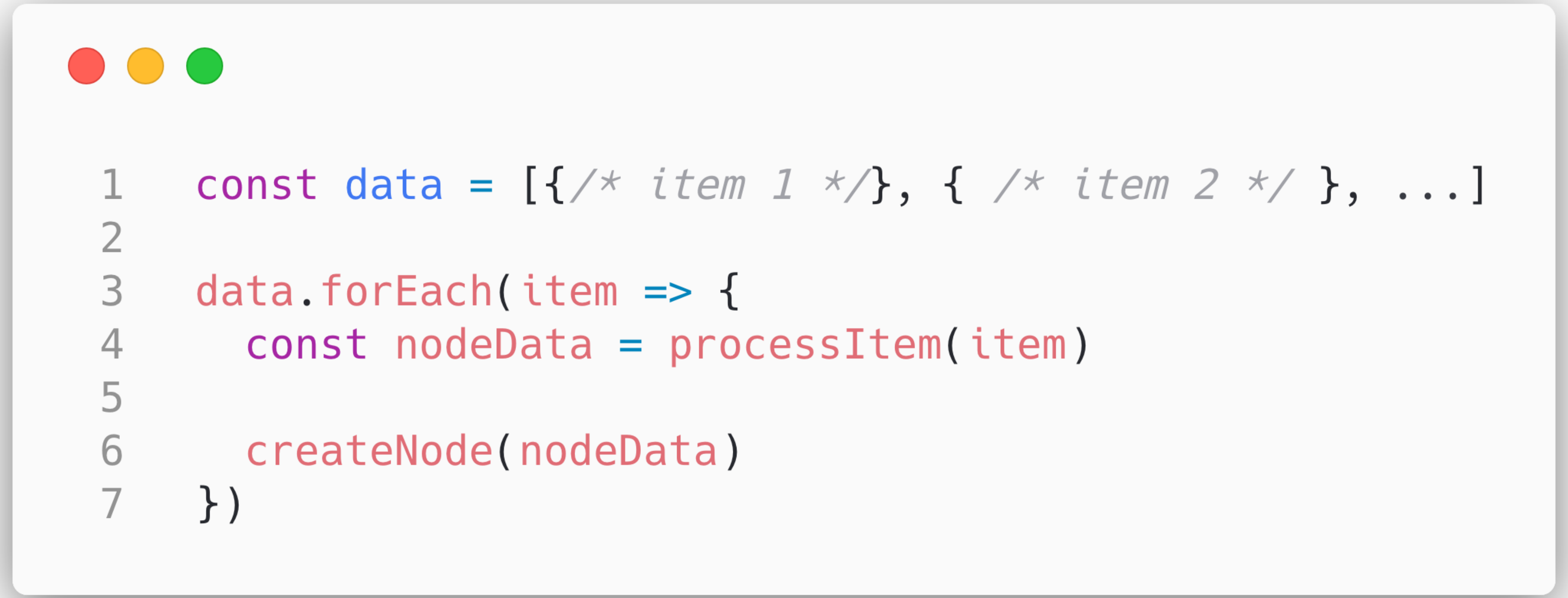
Building a Gatsby Plugin



```
/**
 * gatsby-node.js
 */

exports.sourceNodes = (
  { actions, createNodeId, createContentDigest },
  configOptions
) => {
  const { createNode } = actions

  // Helper function that processes a photo to match Gatsby's node structure
  const processItem = item => {
    const nodeId = createNodeId(`node-${item.id}`)
    const nodeContent = JSON.stringify(item)
    const nodeData = Object.assign({}, item, {
      id: nodeId,
      parent: null,
      children: [],
      internal: {
        type: `MyItem`,
        content: nodeContent,
        contentDigest: createContentDigest(item),
      },
    })

    return nodeData
  }

  const data = [{/* item 1 */}, { /* item 2 */ }, ...]

  data.forEach(item => {
    const nodeData = processItem(item)

    createNode(nodeData)
  })
}
```

& /phacks

Building a Gatsby Plugin



```
const data = [{/* item 1 */}, { /* item 2 */ }, ...]

data.forEach(item => {
  const nodeData = processItem(item)

  createNode(nodeData)
})
```

Live coding!

🙏

Thanks!

Slides, links & GitHub sources will be available later on my Twitter: @phacks ☺️