

Modern Web Development on the Jamstack

Written by: Mathias Biilmann & Phil Hawksworth

Audiobook PDF Accompaniment

Chapter 3

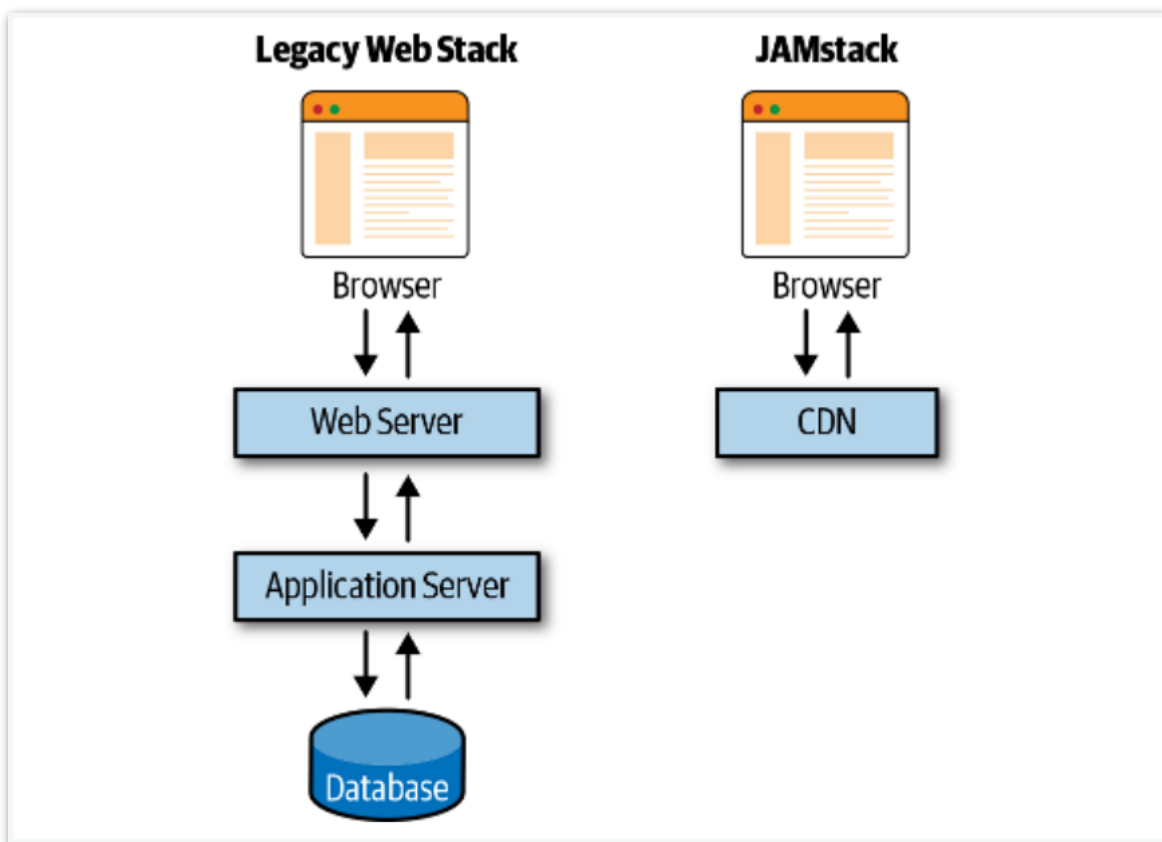


Figure 3-1. Legacy web stack versus Jamstack

Chapter 4



Figure 4-1. Markdown and the resulting HTML output

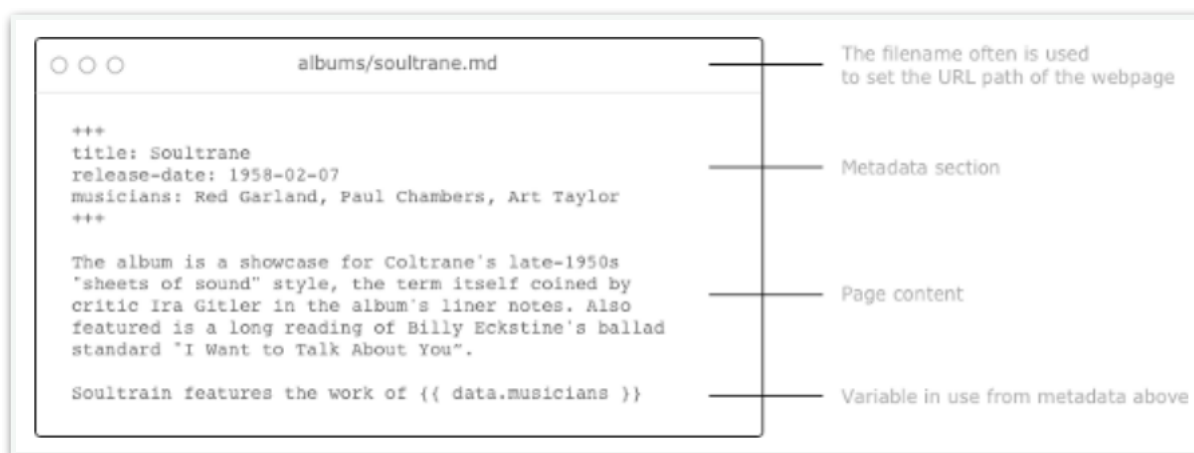


Figure 4-2. A sample markdown file with metadata

Chapter 4 (cont'd)

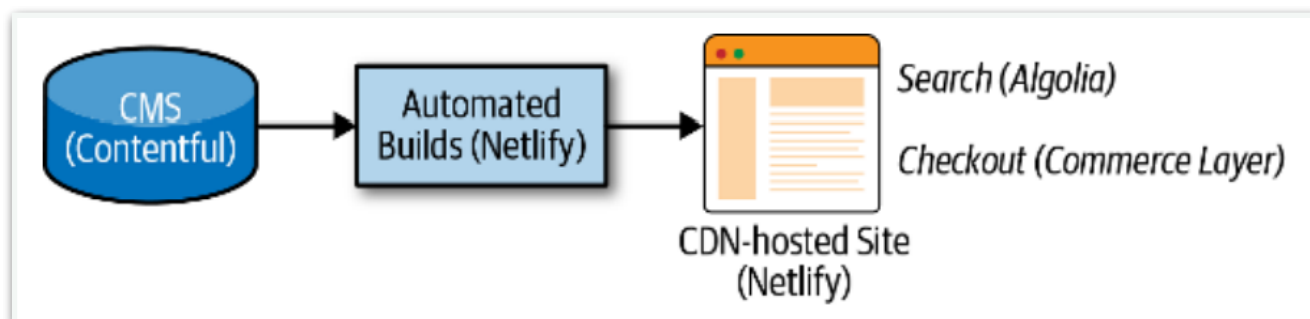


Figure 4-3. Sample Jamstack commerce setup

Chapter 5

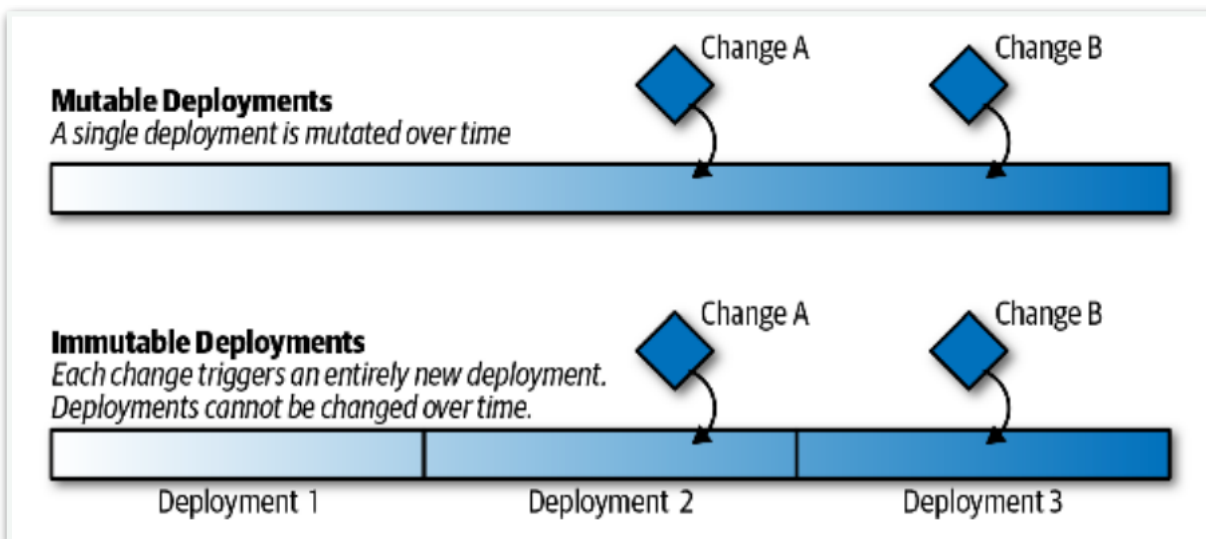


Figure 5-1. Mutable versus immutable deploys

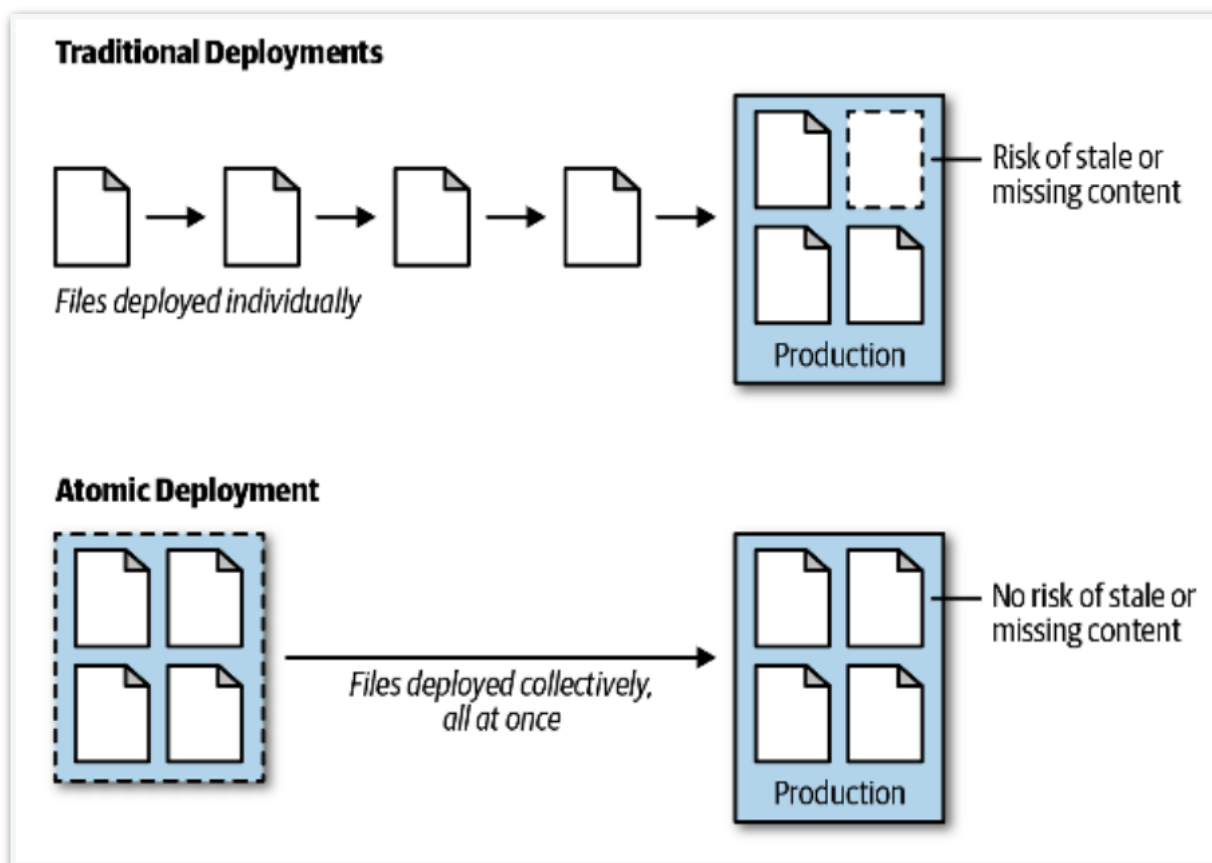


Figure 5-2. Traditional versus atomic deploys

Chapter 5 (cont'd)

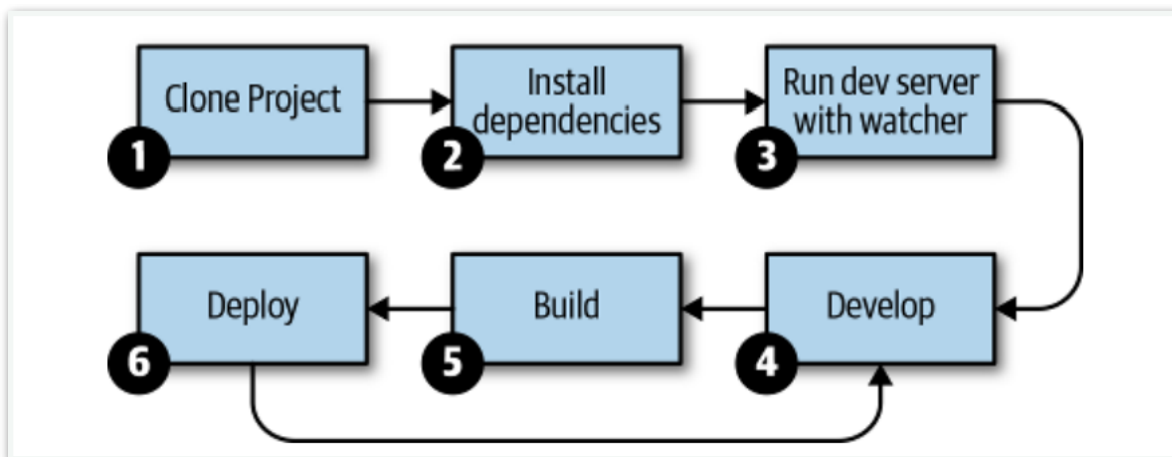


Figure 5-3. Common development workflow

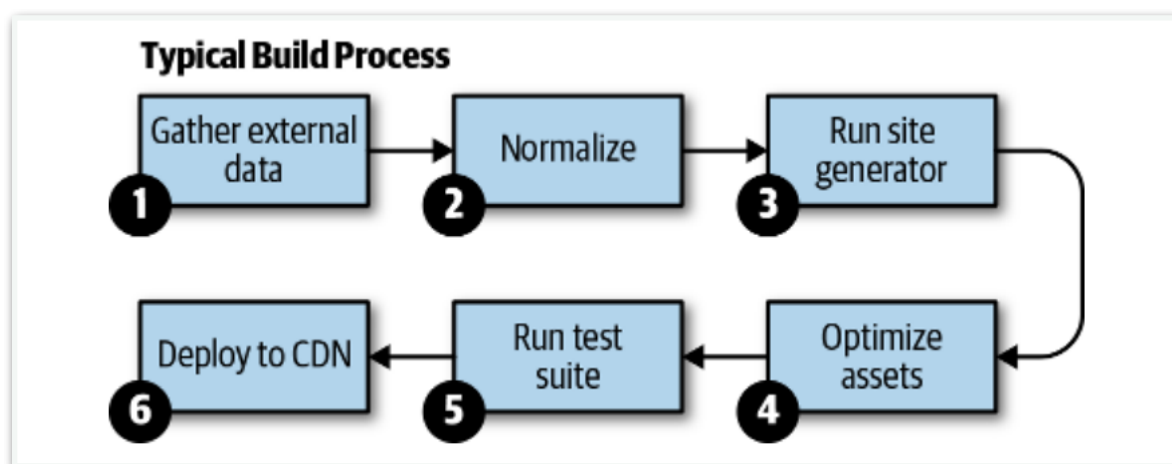


Figure 5-4. Typical build process

Chapter 6

```
/content
/articles
  /2016-01-01-some-old-article-about-css.md
  /2016-01-02-another-old-article-about-animation.md
  ...
/printed-books
  /design-systems.md
  /digital-adaption.md
  ...
/ebooks
  /content-strategy.md
  /designing-better-ux.md
  ...
/jobs
  /2018-04-01-front-end-developer.md
  /2018-05-02-architecture-astronaut.md
  ...
...
/data
  /authors
    /matt-biilmann.yml
    /phil-hawksworth.yml
    ...
  /categories
    /css.yml
    /ux.yml
```

Code 6-1. Content Migration (pg. 75)

Chapter 6 (cont'd)

```

---
title: Why Static Site Generators Are The Next Big Thing
slug: modern-static-website-generators-next-big-thing
image: 'https://cloud.netlifyusercontent.com/assets/
344dbf88- fdf9-42bb-adb4-
46f01eedd629/15bb5653-5a31-4ac0-85fc-c946795f10bd/jekyll-
opt. png'
date: 2015-11-02T23:03:34.000Z
author: mathiasbiilmannchristensen
description: >-
  Influential design-focused companies such as Nest and
  MailChimp now use static website generators for their
  primary websites. Vox Media has built a whole publishing
  system around Middleman...
categories:
  - Coding
  - Tools
  - Static Generators
-

<p> At <a href="https://www.staticgen.com">StaticGen</a>,
our open-source directory of static website generators
</strong>, we've kept track of more than a hundred
generators for more than a year now, and we've seen both
the volume and popularity of these projects take off
incredibly on GitHub during that time, going from just 50
to more than 100 generators and a total of more than
100,000 stars for static website generator repositories.
</p>

...

```

Code 6-2. Utilizing Structured Content (pg. 76)

Chapter 6 (cont'd)

```
gulp.task('copy-site', (cb) => {
  return gulp.src([
    'site/**/*', '!site/content/articles/*',
    '!site/productionarticles/*'
  ], { dot: true })
  .pipe(gulp.dest('prod', {overwrite: true}));
});

gulp.task('copy-articles', ['copy-site'], (cb) => {
  return gulp.src([
    'site/productionarticles/*'
  ])
  .pipe(gulp.dest(
    'prod/content/articles', {overwrite: true}
  ));
});
```

Code 6-3. Working with Large Sites (pg. 78)

```
<div class="article--grid__container">
  {{ first 7 (
    where .Data.Pages.ByDate.Reverse "Section" "articles"
  )}}
  ...
  {{ partial "article--grid" . }}
  ...
  {{ end }}
</div>
```

Code 6-4. Building Out the Core (pg. 79)

Chapter 6 (cont'd)

featured_articles:

- articles/2018-02-05-media-queries-in-2018.md
- articles/2018-01-17-understanding-using-rest-api.md
- articles/2018-01-31-comprehensive-guide-product-design.md
- articles/

Code 6-5. Building Out the Core (pg. 80)

```
{{ range $index $featured :=
  .Site.Data.curated.featured_articles }}
  {{ range where $.Data.Pages "Section" "articles"}}
    {{ if eq $featured .Path }}
      {{ partial "featured-article" . }}
    {{ end }}{{ end }} {{ end }}
```

Code 6-6. Building Out the Core (pg. 80)

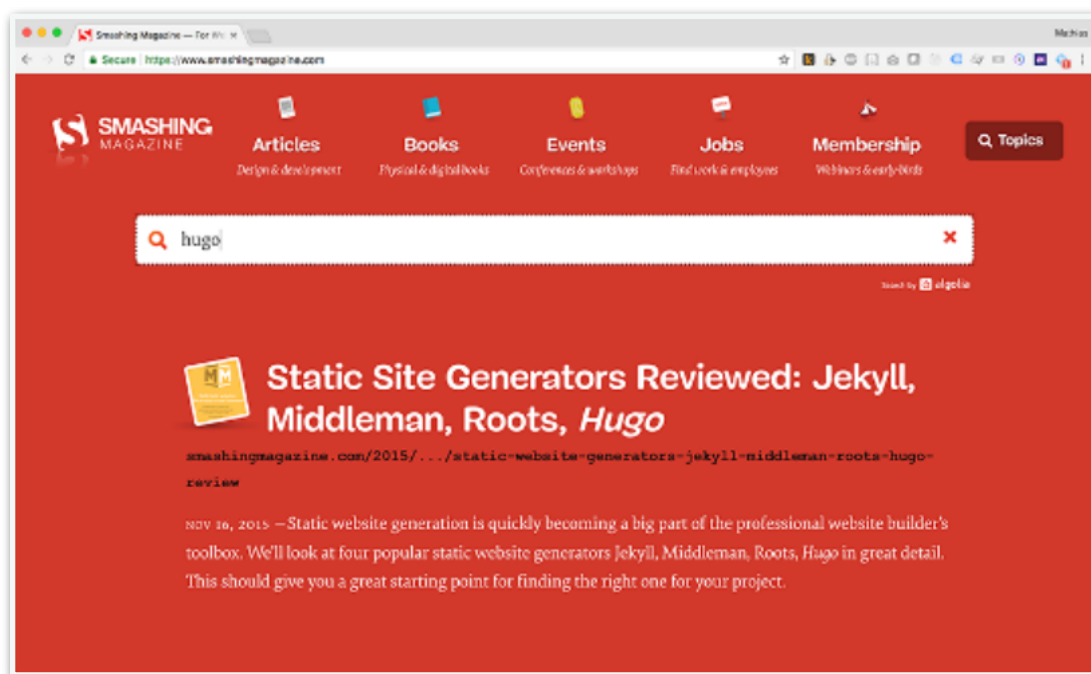


Figure 6-1. Smashing Magazine search

Chapter 6 (cont'd)

```
import algoliasearch from 'algoliasearch'
export const client = algoliasearch(ENV.algolia.appId, ENV.algolia.apiKey)
export const index = client.initIndex(ENV.algolia.indexName)
export const search = (query, page = 0) => (dispatch, getState) => {
  dispatch({
    type: types.SEARCH_START,
    query
  }) index.search(query, {
    hitsPerPage: (page + 1) * 15, page: 0
  }, (error, results) => {
    if (error) return dispatch({
      type: types.SEARCH_FAIL,
      error: formatErrorMessage(error)
    }) dispatch({
      type: types.SEARCH_SUCCESS,
      results,
      page
    })
  })
}
```

```
props.store.dispatch(search(query))
```

```
<div data-component="SearchResults" data-lite="true"></div>
```

Code 6-7. Search (pg. 82)

Chapter 6 (cont'd)

```
<form
  data-handlers="Search" method="get"
  action="https://www.google.com/webhp?q=site:smashingmagazine.com"
>
  <label for="js-search-input"></label>
  <div class="search-input-wrapper">
    <input type="search" name="q" id="js-search-input"
      autocomplete="off" placeholder="Search Smashing..."
      aria-label="Search Smashing"
      aria-controls="jssearch-results-dropdown"
    />
  </div>
</form>
```

Code 6-8. Search (pg. 82-83)

Chapter 6 (cont'd)

```
collections:
- name: articles
  label: "Smashing Articles"
  folder: "site/production-articles"
  sort: "date:desc"
  create: true # Allow users to create new documents in this collection
  slug: "{{slug}}"
  fields: # The fields each document in this collection have
    - {label: "Title", name: "title", widget: "string"}
    - {label: "Slug", name: "slug", widget: "string", required: false}
    - {label: "Author", name: "author", widget: "relation", collection:
      "authors", searchFields: ["first_name", "last_name"],
      valueField: "title"}
    - {label: "Image", name: "image", widget: "image"}
    - {label: "Publish Date", name: "date", widget: "datetime",
      format: "YYYY-MM-DD hh:mm:ss"}
    - {label: "Quick Summary", name: "summary", widget: "markdown",
      required: false}
    - {label: "Excerpt", name: "description", widget: "markdown"}
    - {label: "Store as Draft", name: "draft", widget:
      "boolean",required: false}
    - {label: "Disable Ads", name: "disable_ads", widget: "boolean",
      required: false} - {label: "Disable Panels",
      name: "disable_panels", widget: "boolean", required: false}
    - {label: "Disable Comments", name: "disable_comments",
      widget: "boolean", required: false}
    - {label: "Body", name: "body", widget: "markdown"}
    - {label: "Categories", name: "categories", widget: "list"}
```

Code 6-9. Building Out the Core (pg. 85)

Chapter 6 (cont'd)

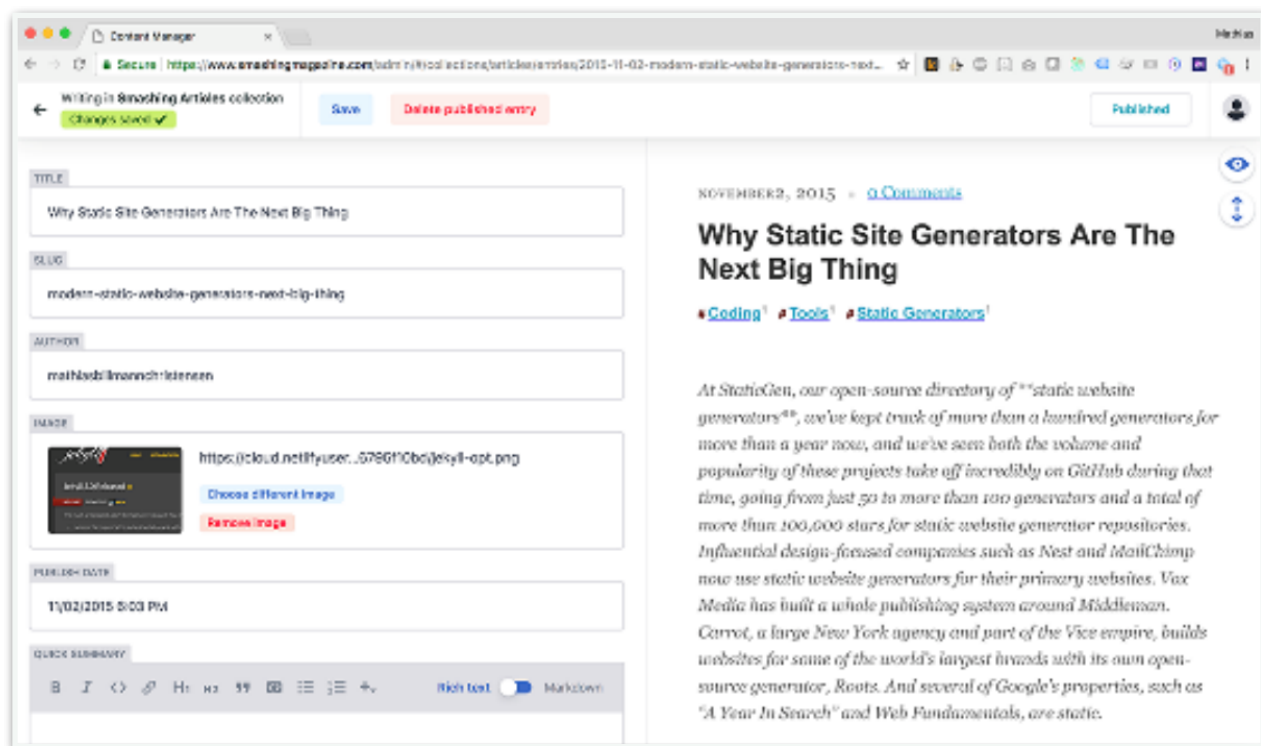


Figure 6-2. Netlify CMS

```
import React from 'react';
import CMS from "netlify-cms";
import ArticlePreview from './cms-preview-templates/Article';
import AuthorPreview from './cms-preview-templates/Author';
// ... more preview imports
import {PriceControl} from './cms-widgets/price';
import {SignaturePlugin, PullQuotePlugin, FigurePlugin}
  from './cms-editorplugins/plugins';
window.CMS.registerPreviewStyle('/css/main.css');
window.CMS.registerPreviewStyle('/css/cms-preview.css');
window.CMS.registerPreviewTemplate('articles', ArticlePreview);
window.CMS.registerPreviewTemplate('authors', AuthorPreview);
// ... more preview registrations
window.CMS.registerWidget('price', PriceControl);
window.CMS.registerEditorComponent(FigurePlugin);
window.CMS.registerEditorComponent(PullQuotePlugin);
window.CMS.registerEditorComponent(SignaturePlugin);
```

Code 6-10. Integrating Netlify CMS (pg. 86-87)

Chapter 6 (cont'd)

```
import React from 'react';
import format from 'date-fns/format';
import AuthorBio from './author-bio';
export default class ArticlePreview extends React.Component {
  render() {
    const {entry, fieldsMetaData, widgetFor, getAsset} = this.props;
    const data = entry && entry.get('data').toJS();
    const author = fieldsMetaData.getIn(['authors', data.author]);
    return <article className="block article" role="main">
      <div className="container">
        <div className="container">
          <div className="row">
            <div className="col col-12 col-article-head">
              {author && <
                AuthorBio author={author.toJS()} getAsset={getAsset}
              />}
            <header className="article__header">
              <div className="article__meta">
                <time className="article__date">
                  <span className="article__date__month">
                    {format(data.date, 'MMMM')}
                  </span>
                  {format(data.date, 'D, YYYY')}
                </time>
                <svg aria-hidden="true"
                  style={{margin: '0 0.75em'}}
                  viewBox="0 0 7 7" width="7px" height="7px">
                  <title>bullet</title>
                  <rect fill="#ddd" width="7" height="7" rx="2" ry="2"/>
                </svg>
                <span className="article__comments-count">
                  <a href="#">
                    <span className="js-comments-count">0</span>
                    Comments
                  </a></span>
                </div>
              ...
            </div>
          </div>
        </div>
      </div>
    </article>
```

Chapter 6 (cont'd)

```

...
<h2>{ data.title }</h2>
<div className="article__tags">
  {data.categories && data.categories.map
    ((category) => (
      <span className="article__tag"
        key={category}>
        <a href="#">{category}</a>
        <sup className="article__tag__count">1</sup>
      </span>
    ))}
</div>
</header>
</div>
<div className="col col-4 col-article-summary">
  <p className="article__summary">
    { data.description }
  </p>
</div>
<div className="col col-7 article__content">
  { widgetFor('body') }
</div>
</div>
</article>;
}
}

```

Code 6-11 (cont'd). Integrating Netlify CMS (pg. 87-88)

Chapter 6 (cont'd)

```
{
  "sub": "1234",
  "email": "matt@netlify.com",
  "user_metadata": {
    "name": "Matt Biilmann"
  },
  "app_metadata": {
    "roles": ["admin", "cms"],
    "subscription": {
      "id": "4321",
      "plan": "smashing"
    }
  }
}
```

Code 6-12. JWTs and Stateless Authentication

```
return auth.signup(data.email, data.password, {
  firstname: data.firstname,
  lastname: data.lastname
}).then(user => { return auth.login(data.email,
data.password, true)
  .then(user => { persistUserSession(user, noRedirect);
return user; })
```

Code 6-13. JWTs and Stateless Authentication

Chapter 6 (cont'd)

```

-
title: 'A Career On The Web: On The Road To Success'
sku: a-career-on-the-web-on-the-road-to-success
image: >-
//cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-
46f01eedd629/133e54ba-e7b5-40ff-957a-f42a84cf79ff/on-the-roadto-
success-365-large.png
description: >-
There comes a time in everyone's career when changing jobs is the natural
next step. But how can you make the most of this situation and find a job
you'll love?
price:
  eur: '4.99'
  usd: '4.99'
published_at: March 2015
isbn: 978-3-945749-17-3
files:
  pdf: >-
    //api.netlify.com/api/v1/sites/344dbf88-fdf9-42bb-adb4- 46f01eedd629/
assets/58eddda0351deb3f6122b93c/public_signature
---
<p>There comes a time in everyone's career when changing jobs is the
natural next step. Perhaps you're looking for a new challenge or you feel
like you've hit a wall in your current company? Either way, you're
standing at a crossroad, with an overwhelming amount of possibilities in
front of you. But how can you make the most of this situation? How can you
<strong> find a job you will truly love</strong>?</p>

```

Code 6-14. ECommerce

Chapter 6 (cont'd)

```
<script class="gocommerce-product" type="application/json">
{
  "sku": "a-career-on-the-web-assuming-leadership",
  "title": "A Career On The Web: Assuming Leadership",
  "image": "//cloud.netlifyusercontent.com/assets/344dbf88-fdf9- 42bb-
adb4- 46f01eedd629/3eb5e393-937e-46ac-968d-249d37aebcef/
assumingleadership-365- large.png",
  "type": "E-Book",
  "prices": [
    {"amount": "4.99", "currency": "USD"},
    {"amount": "4.99", "currency": "EUR"}
  ],
  "downloads": [
    {
      "format": "pdf",
      "url": "//api.netlify.com/api /v1/sites/344dbf88-fdf9- 42bb-
adb4-46f01eedd629/assets/58edd9f7351deb3f6122b911/public_ signature"
    }
  ]
}
</script>
```

Code 6-15. ECommerce

```
gocommerce.addToCart(item).then(cart =>{
  dispatch({
    type: types.ADD_TO_CART_SUCCESS,
    cart,
    item
  })
})
```

Code 6-16. ECommerce

Chapter 6 (cont'd)

```
export const putOrder = (dispatch, getState) => {
  const state = getState()
  const { order, cart, auth } = state dispatch(
    { type: types.ORDER_CONFIRM_START }
  )
  // Create a new order in Gocommerce
  gocommerce.order({
    email: order.details.email,
    shipping_address: formatAddress(order.details.shippingAddress),
    billing_address: formatAddress(order.details.billingAddress) })
    .then(result => {
      // Create a payment for the order with the stripe token or paypal
      payment
    return gocommerce.payment({
      order_id: result.order.id,
      amount: result.cart.total.cents,
      provider: order.paypal ? 'paypal' : 'stripe',
      stripe_token: order.key,
      paypal_payment_id: order.paypal && order.paypal.paymentID,
      paypal_user_id: order.paypal && order.paypal.payerID })
      .then(transaction => {
        // All done, clear the cart and confirm the order
        if (auth.user) dispatch(saveUserMetadata(order.details))
        dispatch(emptyCart)
        dispatch(clearOrder)
        dispatch({
          type: types.ORDER_CONFIRM_SUCCESS,
          transaction: Object.assign({},
            transaction, { order: result.order }),
          cart
        })
      })
    })
  }).catch(error => dispatch({
    type: types.ORDER_CONFIRM_FAIL,
    error: formatErrorMessage(error)
  })))
}
```

Code 6-17. ECommerce

Chapter 6 (cont'd)

Your first name

Your family name

Your country (for taxes & silly jokes 🇺🇸)

Email

Password

SHOW

☐ I don't even like cats. I'm a dog person!

🔒

Enter your credit card details

Card number

MM / YY

CVC

Cardholder name (as printed on your card)

[Ah, what is CVV again?](#)

🔒

Become a Member →

Figure 6-3. Smashing Magazine sign-up form

Chapter 6 (cont'd)

```
fetch('/.netlify/functions/membership', {
  headers: {Authorization: `Bearer ${token}`},
  method: 'POST',
  body: JSON.stringify({ plan, stripe_token: key })
})
```

Code 6-18. Membership and Subscriptions

```
function subscribeToPlan(params, token, identity) {
// Fetch the user identified by the JWT from the identity service
return fetchUser(token, identity).then(user => {
  console.log("Subscribing to plan for user: ", user);
  // Check if the user is already linked to a Stripe Customer
  // if not, create a new customer in Stripe
  const customer_id = user.app_metadata.customer
    ? user.app_metadata.customer.id : createCustomer(user);
  return Promise.resolve(customer_id)
    .then(customer_id => {
      // Check if the user has an existing subscription
      if (user.app_metadata.subscription) {
        // Update the existing Stripe subscription
        return updateSubscription(
          customer_id,
          user.app_metadata.subscription.id, params
        ).then(subscription => {
          // Add the user to the MailChimp list and the right Group
          addToList(user, params.plan);
          // Notifiy about the change of plan in Slack
          sendToSlack( 'The user ' + user.email + ' changed from ' +
            user.app_metadata.subscription.plan + ' to ' + params.plan
          );
          return subscription;
        });
      }
    });
}
```

Code 6-19. Membership and Subscriptions

Chapter 6 (cont'd)

```
// No existing subscription, create a new subscription in Stripe return
createSubscription(customer_id, params).then( subscription => {
  // Add the user to the MailChimp list and the right Group
  addToList(user, params.plan);
  // Notify about the new subscriber in Slack
  sendToSlack(
    'Smashing! The user ' + user.email + ' signed up for a ' +
params.plan + ' plan'
  );
  return subscription;
});
})
// In the end, update the user in the identity service.
// This will update app_metdata.customer and app_metadata.subscription
// for the user
.then(subscription => updateUser(user, subscription, identity)); });
}
```

Code 6-19 (cont'd). Membership and Subscriptions

```
"member_discounts": [
  {
    "claims": {"app_metadata.subscription.plan": "member"},
    "fixed": [
      {"amount": "10.00", "currency": "USD"},
      {"amount": "10.00", "currency": "EUR"}
    ],
    "product_types": ["Book"]
  }
]
```

Code 6-20. Tying it Together

Chapter 6 (cont'd)

```
{
  "email": "joe@example.com",
  "sub": "1234",
  "app_metadata": {"subscription": {"plan": "member"}}
}
```

Code 6-21. Tying it Together

```
function processJob(payload, job) {
  console.log("Processing job", job.meta);
  const content = `-\n${yaml.safeDump({
    title: job.meta.title || null,
    order_id: payload.order_id || null,
    date: payload.created_at || null,
    logo: job.meta.logo || null,
    commitment: job.meta.commitment || null,
    company_name: job.meta.company_name || null,
    company_url: job.meta.company_url || null,
    jobtype: job.meta.jobtype || null,
    location: job.meta.location || null,
    remote: job.meta.remote || null,
    application_url: job.meta.application_url || null,
    featured: job.meta.featured || false
  })}\n---\n\n${job.meta.description || ""}`;
```

Code 6-22. Job Board and Event Tickets

Chapter 6 (cont'd)

```
const path = `site/content/jobs/${format(payload.created_at, "YYYY-MM-DD")}-${payload.id}-${urlize(job.title)}.md`;
const branch = "master";
const message = `Create Job Post This is an automatic commit creating the Job Post: "${job.meta.title}"`;
return fetch( `https://api.github.com/repos/smashingmagazine/smashingmagazine/contents/${path}`,
  {
    method: "PUT",
    headers: {Authorization: `Bearer ${process.env. GITHUB_TOKEN}`}
  },
  body: JSON.stringify({
    message,
    branch,
    content: new Buffer(content).toString("base64")
  })
);
}
```

Code 6-22 (cont'd). Job Board and Event Tickets