Rise of the JAMstack

Matt Biilmann, 2017
Matt Biilmann
CEO, Netlify
Publishing for the modern web
Building thousands of websites

7 years in Spain, 2 CMSs, 10k+ websites
Building a fully hosted CMS

Moved to the Bay Area, built a managed CMS for the legacy stack
Trends: Git, MicroServices, JS

Git vs FTP

Microservices

Modern Javascript (ES6)
The Web Needs a Paradigm Shift

For the last 15 years the web has been built on the fly, causing ever increasing issue with:

**Security**
7% of all legacy sites hacked

**Performance**
Single origin, builds for every visitor, low conversion

**Scalability**
Infrastructure needs scaling up front
The Evolution of the Web

- **Unix Model** (~1970-1997)
  - CLIENT
  - SERVER

- **Legacy Web** (~1997-Today)
  - CLIENT
  - WEB SERVER
  - APP SERVER
  - DATABASE

- **Modern Web** (Now)
  - CLIENT
  - CDN
  - SERVICES
A New Stack

Decoupled Architecture + Git Centric Workflow = A Better Web

10x Faster
No more malware
Infinitely scalable
**JavaScript**

Any dynamic programming during the request/response cycle is handled by JavaScript, running entirely on the client. This could be any frontend framework, library, or even vanilla JavaScript.

**APIs**

All server-side processes or database actions are abstracted into reusable APIs, accessed over HTTP with JavaScript. These can be custom-built or leverage third-party services.

**Markup**

Templated markup should be prebuilt at deploy time, usually using a site generator for content sites, or a build tool for web apps.
Don't Miss These Articles on Smashing

Vitaly Friedman wrote — ABOUT 1 MONTH AGO
A Little Surprise Is Waiting For You Here — Meet The Next Smashing Magazine ▪ 101 comments

Anastasia Shevchuk wrote — ABOUT 2 MONTHS AGO
The Art Of Calligraphy: Getting Started And Lessons Learned ▪ 9 comments

# Inspiration 609  # Typography 171  # Fonts 95

# Smashing Magazine 1  # Redesign 6
Building an Open-Source Eco-System

NetlifyCMS
Netlify CMS is the first CMS for the new age of Git-centric web development. It is completely build tool agnostic and works with storing structured content in Git. A CMS for site generators. Netlify-CMS gives non-technical users a simple way to edit and add content to any site built with a site generator.
https://www.netlifycms.org
Contribute on GitHub ➔

GoTell
GoTell is an API and build tool for handling large amounts of comments.
https://github.com/netlify/gotell
Contribute on GitHub ➔

GoTrue
GoTrue is a small open-source API written in golang that can act as a self-standing API service for handling user registration and authentication for projects. It’s based on OAuth2 and JWT and will handle user signup, authentication and custom user data.
https://www.gotrueapi.org
Contribute on GitHub ➔

GoJoin
GoJoin is a tiny API that acts as a wrapper for Stripe’s subscription API. It exposes simple call methods for single page apps and sites.
https://github.com/netlify/gojoin
Contribute on GitHub ➔

GoCommerce
GoCommerce is a small Go based API for e-commerce sites that handles orders and payments. It integrates with Stripe for payments and will support international pricing and VAT verification.
https://www.gocommerceapi.org
Contribute on GitHub ➔

Gotiator
A tiny API Gateway based on JSON Web Tokens. Gotiator can handle simple API proxying with signing for single page apps that already use JWTs for authentication.
https://github.com/netlify/gotiator
Contribute on GitHub ➔
An open-source CMS for your Git workflow

- Content management in a single-page app
- Built in React.js with extensible components
- Integrate with any build tool
- Plug in to any static site generator

Find out more →
Try it out →
Make it better →
Decoupled E-commerce
A headless e-commerce API with client-side JavaScript integration using the GoCommerce JS library.

- Handles orders and payments
- Integrates with Stripe or PayPal
- Supports international pricing and VAT verification
- Built in Go, accessible via client-side JavaScript
Handle 200,000 comments!

Vitaly Friedman wrote — About 1 month ago

A Little Surprise Is Waiting For You Here — Meet The Next Smashing Magazine

101 comments

# Smashing Magazine # Redesign
Building an Open-Source Eco-System

**NetlifyCMS**
Netlify CMS is the first CMS for the new age of Git-centric web development. It is completely build tool agnostic and works with storing structured content in Git. A CMS for site generators. Netlify-CMS gives non-technical users a simple way to edit and add content to any site built with a site generator.
https://www.netlifycms.org

Contribute on GitHub →

**GoTell**
GoTell is an API and build tool for handling large amounts of comments.
https://github.com/netlify/gotell

Contribute on GitHub →

**GoTrue**
GoTrue is a small open-source API written in golang that can act as a self-standing API service for handling user registration and authentication for projects. It's based on OAuth2 and JWT and will handle user signup, authentication and custom user data.
https://www.gotrueapi.org

Contribute on GitHub →

**GoJoin**
GoJoin is a tiny API that acts as a wrapper for Stripe's subscription API. It exposes simple call methods for single page apps and sites.
https://github.com/netlify/gojoin

Contribute on GitHub →

**GoCommerce**
GoCommerce is a small Go based API for ecommerce sites that handles orders and payments. It integrates with Stripe for payments and will support international pricing and VAT verification.
https://www.gocommerceapi.org

Contribute on GitHub →

**Gotiator**
A tiny API Gateway based on JSON Web Tokens. Gotiator can handle simple API proxying with signing for single page apps that already use JWTs for authentication.
https://github.com/netlify/gotiator

Contribute on GitHub →
JavaScript
Any dynamic programming during the request/response cycle is handled by JavaScript, running entirely on the client. This could be any frontend framework, library, or even vanilla JavaScript.

APIs
All server-side processes or database actions are abstracted into reusable APIs, accessed over HTTP with JavaScript. These can be custom-built or leverage third-party services.

Markup
Templated markup should be prebuilt at deploy time, usually using a site generator for content sites, or a build tool for web apps.
Best Practises

- Entire Project on a CDN
- Automated Builds
- Everything Lives in Git
- Atomic Deploys
- Modern Build Tools
- Instant Cache Invalidation
Let’s Continue The Talk!

#JAMstack
jamstack.org
gitter.im/jamstack/community

@biilmann