

JAMSTACK

Aromal Anil

S7CS B

October 22, 2020

Guided By:

Jisy Raju

Department of Computer Science and Engineering

1. History
2. Introduction to Jamstack
3. How Jamstack Works
4. Traditional Web vs Jamstack
5. Why Jamstack
6. Jamstack Concepts
7. Tools for Jamstack
8. Live Demo
9. Reference

History

Problem with Legacy Web

- For each request HTML page is generated in server.
- Backend focused, generally runs on a web server all the times.
- Backend & Frontend are coupled together.
- Provides way too many opportunities for attack.
- Expensive to scale.

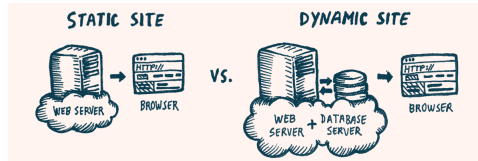
Static Websites vs Dynamic Websites

Static Websites

- It sends exactly the same response for every request.
- Will not contain dynamic data.
- Content changes only when someone publishes and updates the file.

Dynamic Websites

- It may generate different HTML for each of the request.
- Can contain dynamic data.
- Server can generate unique content for each request using SSR.



Static vs Dynamic Websites ¹³

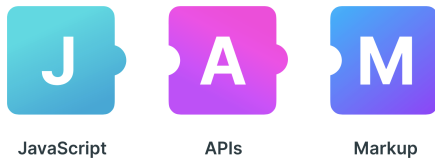
What lead to the need for a new approach.

- Static websites were not static anymore.
- Static website was able to show dynamic contents using JS & AJAX calls.
- Emergence of modern frontend frameworks.
- Availability of backend features like routing in frontend.
- Introduction of Node.js, allowed JavaScript to work on the server-side.
- Introduction of Serverless functions.
- These arose the need for a frontend focused stack.

Introduction to Jamstack

What is Jamstack

- Stands for JavaScript, API Markup.
- New approach to build fast and secure websites.
- Gives dynamic capabilities to static websites.



Components of Jamstack ¹



JavaScript ²

- Adds dynamic interactions to the website.
- Make client side rendering possible.
- No restriction on libraries/ frameworks you must use.



API ³

- Server side operations are abstracted into APIs.
- Accessed using JavaScript over HTTP/HTTPS.
- Each functions of sever can be split into separate APIs forming Microservices .
- Third party APIs could be used for adding functionality.



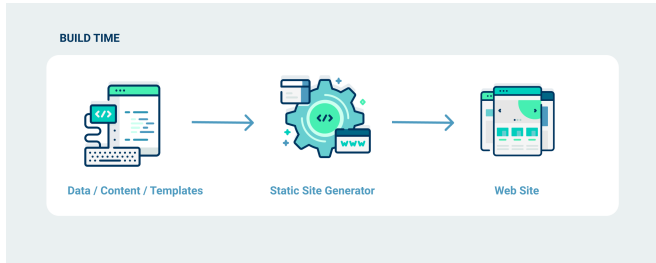
Hyper Text Markup Language (HTML) ⁴

- Any form of markup language like HTML, Markdown etc.
- During build process other forms of markups are converted into HTML pages.
- Static site generators can generate static HTML using data from database markup file.

How Jamstack Works

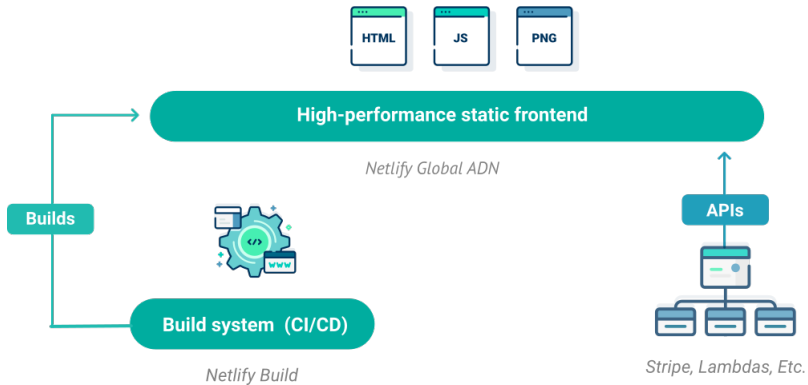
- Static files are generated during development or build process.
- Pre-built markup and assets are served to user on request.
- Browser executes JavaScript after HTML page is loaded.
- For any dynamic functionalities API call is done.
- The JavaScript updates the page according to the response from the API.

- Takes Markup/Template/Data & convert it into web pages during build process.
- Since web pages are generated during build time, it eliminates the need to render pages on request.



Working of Static Site Generators ¹²

Jamstack Architecture.



Jamstack Architecture ⁵

Traditional Web vs Jamstack

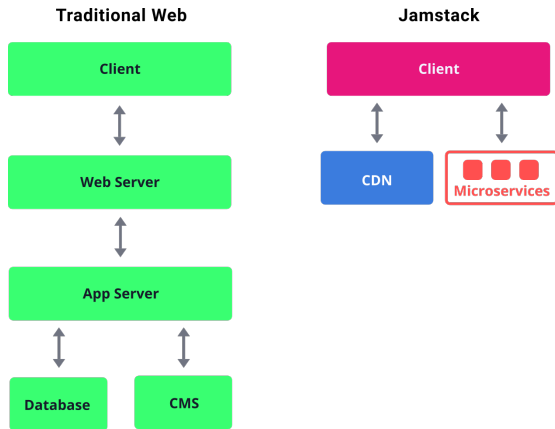
Traditional Web

- Frontend and Backend are *Coupled*.
- Server side rendering.
- Content updates are pushed through traditional CMS, like WordPress or Drupal.
- Core updates pushed to server using FTP.

Jamstack

- Frontend and Backend are *Decoupled*
- Webpages are pre-rendered.
- Content updates are pushed through Git or a headless CMS.
- Core updates are pushed through Git.

Traditional Web vs Jamstack



Traditional Web vs Jamstack ⁶

Why Jamstack ?

What Makes Jamstack Great?

Jamstack apps inherently satisfy most of the 5 pillars of the **AWS Well-Architected Framework**.



Characters of AWS well architected framework ⁷

- Assets can be served as static files from a CDN.
- Server don't need to build the page on each request.
- Static content be reproduced and stored across the globe.



CDN Distribution map ⁸

- Server-side logic is abstracted into APIs.
- User is served static HTML pages, without direct connection to the backend or database.
- Can use secure 3rd party APIs for specific needs like authentication.



Available API Services ⁹

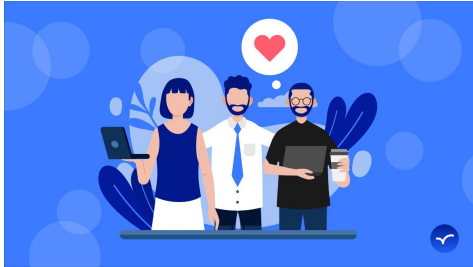
- Static Assets can be cached in a CDN, which are generally cheap.
- Since SSR is not used, server load is decreased.
- Use very few resources than alternatives.



Host Jamstack sites free on Netlify ¹⁰

Better Developer Experience

- No tight coupling between the application backend and frontend.
- Git based deployment makes deployment and development easy.
- Easy integration of third-party services like Algolia, Auth0, Cloudinary etc.



Happy Developers ¹¹

Jamstack Concepts

- **Decoupling**

- Separation of frontend pages and UI from the backend and databases.
- This separation allows frontend to be deployed separately

- **Prebuilding**

- Before deployment entire frontend is build into static pages and assets.
- Highly optimised.
- Example build step of a React app.

- **Git Based Workflow**

- Jamstack ties deployments closely to a Git-based workflows.
- Git brings the rigor and safety of version control to web projects.
- Allows support for numerous contributors with ease.

Tools for Jamstack

- **Static Site Generators**

- Generates static HTML and assets from Markdown or data fetched from database.
- Site Generation occurs during build time.

- **Headless CMS**

- Is a back-end only content management system.
- Makes content accessible via an API.

Live Demo

Reference

1. **Next.js + Netlify**
<https://medium.com/@azizhk/next-js-netlify-c246ea070ae2>
2. **JavaScript Logo**
<https://commons.wikimedia.org/wiki/File:JavaScript-logo.png>
3. **API Logo**
<https://www.pngegg.com/en/png-dqpxs>
4. **HTML logo**
<https://www.w3.org/html/logo/>
5. **Welcome to the Jamstack**
<https://www.netlify.com/jamstack/>
6. **Why Jamstack**
<https://jamstack.org/>
7. **What is the Jamstack and how do I get started?**
<https://www.freecodecamp.org/news/what-is-the-jamstack-and-how-do-i-host-my-website-on-it/>

8. **What is the Jamstack and how do I get started?**
<https://www.freecodecamp.org/news/what-is-the-jamstack-and-how-do-i-host-my-website-on-it/>
9. **New to Jamstack? Everything You Need to Know to Get Started**
<https://snipcart.com/blog/jamstack>
10. **Welcome to the Jamstack**
<https://www.netlify.com/jamstack/>
11. **Happy Developers**
<https://arc.dev/blog/happy-developers-7zr5xar1zs>
12. **What is a Static Site Generator? And 3 ways to find the best one**
<https://www.netlify.com/blog/2020/04/14/what-is-a-static-site-generator-and-3-ways-to-find-the-best-one/>
13. **Why static sites aren't as bad as you think**
<https://tiago.codes/why-static-sites-arent-as-bad-as-you-think>

Thank You
