When does it make sense to go Headless?

What is a Headless Web Application and when you should consider using it.

MARZEE AND ANA ROSSETTO
APR 17, 2024

♡ 1 Share ···

At Marzee, we always do our best to select the right technology stack for each project we deliver. This is why we have experience with a range of technologies and approaches. For us, it is not always about using the latest shiny new technology or keeping true to solid and old technologies. For over 12 years, we have learned that each project, client, and field has its own needs and particularities. We can't simply apply the same approach and use the same technology stack.

We have started mainly working with Drupal without ever stopping to create and deliver static websites. Not every project needs the complexity and robustness that Drupal brings; in cases like this, we need alternatives.

With the rise of Static Site Generators and the JAMstack infrastructure concept, we have given dynamic to simple static sites, allowing editors to easily add and edit content without having to change a line of code. We gave them speed and high performance, while delivering a new website in less time and cost.

As technology advanced, we could build more complex web platforms without relying on monolithic infrastructures, giving the front end more flexibility and possibilities for out-of-the-box designs.

In this scenario, some of the demands we received from our clients proved to be perfect opportunities to separate the frontend from the backend (and content management system) and build what the experts in the field call Headless Web Web Application or a Decoupled Infrastructure.

Photo by Nik on Unsplash

This approach brings a series of benefits to a web platform or website, but it is not something you should apply to every project. It is important to understand the needs and conditions of that specific project and determine if this is a good opportunity to separate the front and back ends.

Before we discuss the aspects and factors that influence decision-making, let’s first understand headless web development and its associated benefits.

In this very intuitive and simple article, Catherine Dee gives a very simple explanation of Headless Web Development:

> "In the simplest terms, headless means "decoupling" the front end and back end, implementing a best-in-class API architecture. You're separating where the content is stored — the content repository — from the presentation layer on which it's displayed."

Algolia

An infrastructure like this can, for instance, make content endlessly reusable, eliminating manual processes like copy and pasting and making editing way easier — change the copy or image in one place and that change applies everywhere the content is located. As simply put in this Storyblok article:

> *"A headless CMS is a backend-only content management system that's built from the ground up as a content repository. The content is then accessible via a RESTful API or GraphQL API for display on any device. This separates your data (the "body") from how it's presented (the "head"), hence the term "headless"*

Storyblok

Separating the head from the body (data) can be beneficial in some projects. Suppose you have an eCommerce store that needs to be updated across different websites and marketplaces. You don't need to replicate this change manually if you have one source coming from a single Content Warehouse.

The same could happen if you have different sources of data to be presented on the same website. You do not always need to put everything under the same infrastructure and have the data come from different sources and be served to the front end.

For instance, we recently built a Live-Streaming platform where the recordings and live videos came from one place and the copy with information from each session came from a Content Management System. Instead of putting everything within a CMS and serving it in the front end, we built an architecture where the videos are served directly to the front end.

One important thing to have in mind is that a project does not need to be just monolithic or headless. You can have a web application with some parts that are headless or decoupled and be monolithic at its core.

For many years, developers have built websites and platforms on top of monolithic Content Management Systems like Drupal or WordPress. As more people used those technologies, their core and plugins evolved to accommodate various needs, always enabling customisations. The idea of one system doing it all.

Using Drupal or WordPress can get as complex as it can be, and there are some projects that don't need the complexity of those technologies, either for reduced scope, budget, or timeline.

For simple marketing and campaign websites with a beautiful and complete custom design, going for simplicity is the best choice. In cases like this, a headless CMS and a Static Site Generator can be the best option to get the job done.

The question remains: When should we consider using Headless?

This is what we usually try to cover when we ask ourselves this question:

- Must have the same information across multiple channels and devices, such as website, App, sub-sites, e-mail, etc.
- Provide personalised user interfaces and outof the box experiences.
- Performance will impact the web application's service.
- Peak traffic websites and applications that need to be easily scalable.
- Website that does not need to have many layout options for editors
- Real-time Notifications and Information delivered across different channels.
- Reduced Budget or/and reduced timeline.
- Must rely on Third-party service providers for specific parts of the website.

We have delivered quite a few Headless applications already, and we share them here so we can maybe help you see in practice when it makes sense to go Headless.

Live-Streaming Website For International Organisartion

- Live-stream website with peak traffic, so performance and scalability were key to them.
- Using Drupal just as the content hub, we trigger builds of the frontend only when content changes.
- Notification system for real-time live stream.
- Fully static site optimized for SEO and fast content delivery.
- Integration with Kaltura JS Player library without having to pass through Drupal.

Multilingual Content Hub for TV Channel

- Improve the creation and organisation of many articles from the various ARTE TV programmes.
- Distribute the same content across different websites and channels.
- Standardised how the 60+ editors publish articles (in different languages) on different pages.
- Use Drupal as a content hub for articles and events that must be displayed across many websites.

eCommerce Platform for NFTs with Football Videos

- The back end developed using Blockchain with data and content that needed to be displayed on an engaging website.
- Out-of-the box design.
- Peak traffic moments with the NFTs package sales.

Campaign Website For Non-profit

- Limited time for design and development - 4 weeks.
- Reduced Budget.
- Simple one-page website.
- No need for layout changes.

Data-driven Web App to Address Racial Inequity in Schools

- Rely on Third-party services to manage the survey process instead of building it from scratch.
- Automatically manage and present mass survey data.
- Auto-scale using AWS products.

Are you ready to consider a Headless Architecture?

Thanks for reading Marzee's Newsletter! Subscribe for free to receive new posts and support my work.

← Previous

A guest post by

Ana Rossetto

A Brazilian living in Portugal and working as a Business Developer at a Web Dev agency.

Comments

Write a comment...