

Bodiless Drupal

Think before you migrate

Paul Day | Quotient, Inc.

Published: July 31, 2017

Introduction

For Federal agencies and other organizations that have invested significant resources in managing domain content, migrating that content to a content management system (CMS) can be an extremely disruptive and expensive endeavor. What if a website built based on a CMS could provide standard CMS features without the expense and disruption resulting from migrating large volumes of distributed domain content? This paper answers that question (spoiler alert: yes), and describes how Drupal can be used to solve this problem, providing several examples to highlight scenarios where Quotient has successfully implemented this approach.

With Drupal's robust Content Management Framework (CMF) architecture, migrating large volumes of data is not required. Instead Drupal's core, including security framework and theme layer, can be used to tap into external data stores to present a unified Drupal experience experience to the users.



What is Bodiless Drupal?

Bodiless Drupal is a Drupal application that makes use of data stored outside of Drupal's internal database. Organizations can leverage Drupal's features to serve domain content from *external* data stores. Drupal does not care where the domain content is stored, nor the format in which it is stored, because custom application components – Drupal modules and theme elements (alters, templates, render callbacks) – will serve as the bridge between the website and the domain content (see Figure 1). External data stores can include, but are not limited to, external databases, SharePoint, Alfresco, JSON or XML API data, or static HTML.

It is important to note that while Drupal installs its own database, a Bodiless Drupal solution only uses this database for standard Drupal functionality such as user accounts, roles and permissions, system settings, and routing. Bodiless Drupal may store temporary copies of pre-processed domain content in the Drupal database to improve application performance, but the actual domain data resides outside of Drupal. In other words, any domain content stored within Drupal is disposable because domain content is not migrated.

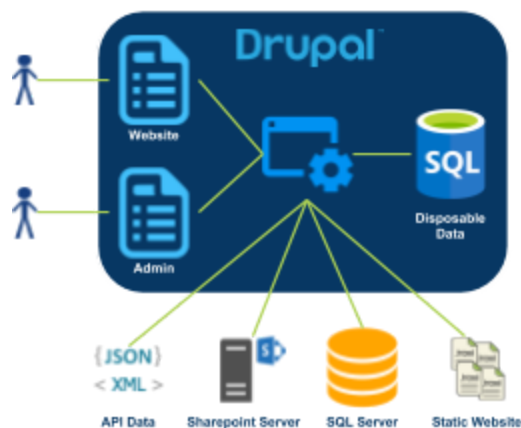


Figure 1: Bodiless Drupal Application

Benefits of Bodiless Drupal

Drupal is an extremely powerful and flexible CMF. Drupal core provides robust security, content editing and content translation, to name a few features. Contributed modules enhance core with many more capabilities.

Bodiless Drupal allows organizations to build dynamic applications using this robust CMF, leveraging their existing content where it lives today. Bodiless Drupal can be used in a number of scenarios: to explore Drupal without fully committing to a migration, to continue to benefit from legacy systems as they sunset, to provide data for proofs-of-concept, or as a long-term solution when non-Drupal systems will continue to be supported in tandem with a Drupal application.

More Power With Drupal 8

With its transition to the Symfony PHP framework, Drupal 8 allows designers and developers to focus solely on site customization through module, application programming interface (API), and theme development. The support for YAML Ain't Markup Language (YAML) promotes separation between application configuration and code, making the application easier to maintain and configure across evolving environments.

Drupal 8 was written with componentization in mind. Componentization provides stakeholders and developers the opportunity to pick and choose the CMF features that make sense for a particular project. These capabilities alone make Drupal 8 a leading choice for organizations that need more than a website delivering static content.

Bodiless and Headless

This description brings some interesting visuals to mind, but there are cases where it makes sense to build a system that is both Bodiless and Headless. In this scenario, Drupal's framework is used for things like user roles, permissions and workflow, but domain data is stored elsewhere and non-Drupal interfaces such as kiosks or mobile platforms are used to display or otherwise use the data. Drupal may be serving the domain data via routes, or the non-Drupal front-ends may access domain data from the non-Drupal data sources in some other fashion, such as by calling API endpoints.

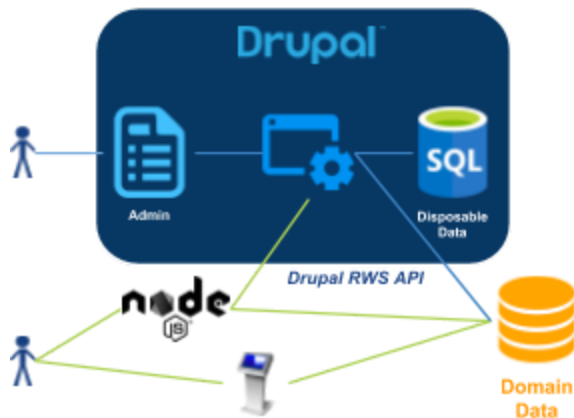


Figure 2: Bodiless and Headless Drupal Application

Headless Drupal, also called *Decoupled Drupal*, is a solution that marries a robust CMF for managing data with the ability to use non-Drupal interfaces to render that data. Headless Drupal leverages the strengths of Drupal while allowing designers to create engaging user experiences through the use of development technologies such as Angular, Ember, React, Node.js, digital kiosks, and other smart devices.

Bodiless Drupal can be used to maintain content management workflows within the Drupal framework, displaying and managing data from external data sources. While a Bodiless Drupal implementation does not store or manage domain content within Drupal's database, Drupal can be used to create, read, update and delete content, and can be used to create relationships between data stored in disparate repositories.

Bodiless and Headless Drupal combines these two concepts, essentially using Drupal as the glue, bridge, or insert-your-favorite-connection-metaphor, between data stored outside of Drupal and one or more user interfaces that are built using a separate application than Drupal.

	Bodiless Solution	Headless Solution	Bodiless & Headless
Drupal content migration required	No	Yes	No
Leverage Drupal's theme and render APIs	Yes	No	No
Leverage Drupal's security and other features	Yes	Yes	Yes
Use Drupal to view and/or manage domain content	Yes	Yes	Yes

Table 1: Bodiless and Headless Drupal Comparison

Bodiless Drupal by Example

Bodiless and Headless

Working with a Federal customer, Quotient's team applied the Bodiless and Headless approach to provide interactive kiosks that display data from an existing large non-Drupal content repository. Drupal 7's Form API was used to access and manage the domain content, and API endpoints were developed to render the content from the external source for display on the interactive kiosks. In conjunction with LDAP integration, Drupal roles and permissions were used to distribute content management responsibilities.

Bodiless With Drupal 8

An organizational unit of an educational institution needed to connect their content with related content from other units of the same institution. Quotient's team leveraged Drupal 8 to provide the user interface for managing the unit's content, storing their content in a central, shared repository, including links to related content owned by other units.

Bodiless With Drupal 7 Too

A Federal program for adult education needed to provide on-going access to the archives of a recently deprecated Mailman discussion list. Hundreds of thousands of HTML files that had been written using Pipedream needed to be

accessible from the client's Drupal 7 website. Quotient's team wrote a controller that would fetch the correct HTML file from disk when archives were requested using their original, un-altered URLs. This allowed the existing HTML files to remain as independent flat files on disk. The controller leveraged DOM-manipulation routines to ensure the final pages met accessibility requirements, and the results were rendered using Drupal's Theme API ensuring the archives matched the look and feel of the rest of the website.

Yes, Even Drupal 6

A Federal client had a legacy ColdFusion system that needed to be slowly deprecated. Data needed to be migrated to Drupal, but the transition had to happen slowly as workflows were reviewed and either moved to Drupal or retired. Quotient's team wrote code that synchronized the ColdFusion data stored in Microsoft SQL Server to nodes stored in Drupal's database. Apache Solr was used to index Drupal's content. The client initially maintained the two systems in tandem, continuing to maintain their data using ColdFusion. The custom code continued to update Drupal's nodes when SQL Server data was updated. Once the client was satisfied that Drupal was a viable long-term solution, the ColdFusion system was deprecated and Drupal's UI was used to maintain the nodes. This bodiless approach facilitated a slow, progressive migration of thousands of nodes.

Conclusion

Bodiless Drupal allows organizations to consider an alternative to a large-scale content migration. Utilizing Drupal's framework without requiring all domain content to be migrated to Drupal's internal database provides flexibility to maintain domain content across various groups while rendering that content in a consistent manner.

More Information

Drupal 8 Features

<https://www.drupal.org/8/platform>

Migrating to Drupal

<https://www.drupal.org/docs/8/migrating-to-drupal>

Ain't No Body: Not Your Mama's Headless Drupal

Presented by Paul Day at Baltimore DrupalCon 4/26/2017

[DrupalCon Session Page](#)

[Also on YouTube](#)

About Quotient, Inc.

Quotient is a woman-owned small business specializing in content management, web application design and development, and IT operational support. As a small business, Quotient has been able to grow and adapt to a rapidly evolving Federal Government, embracing Drupal and other open-source software implementations. Founded in 1999, Quotient has supported Federal agencies in a variety of capacities from providing System Administration and Help Desk support, to implementing and managing Content Management Systems, to complete application design and development services.