TORQUE®

NEWS COMMUNITY BEGINNERS DEVELOPERS LEARN THEMES PLUGINS

SINCE 2013

"ALL THE WORD THAT'S FIT TO PRESS."

A WP ENGINE PUBLICATION

FEATURE LEARN

# YES, YOU REALLY CAN USE WORDPRESS TO BUILD APPS

JOSH POLLOCK JUNE 17, 2014 21 COMMENTS

On a recent episode of the WordPress podcast, The DradCast, WordPress co-founder Matt Mullenweg said that "when you think about it, we're kind of building a web operating system." This remark mirrored statements he made in last year's State of the Word address about making WordPress into a foundation for mobile apps.

Not only has WordPress itself evolved into a more suitable tool for app development, but there's also a slew of new plugins and services available that make it even easier to build web and mobile apps.

Thanks to these developments, WordPress is a relatively easy and affordable option for app development. In this article, I will explain the key questions and concerns you need to address before starting any WordPress app project. I will also introduce you to some options to

address these issues.

# Back-End: Database and Content Management

## Architecture

When planning how you're going to store and retrieve data, it's important to remember that you're not creating a completely new content management system from scratch. You're using WordPress so that you can benefit from the years of work that has gone into developing it.

WordPress's basic architecture of posts and pages can be extended to use as many different content types as you want. For example, an eCommerce plugin such as WooCommerce will add a custom post type (called "products") that's separate from posts and pages, and has custom fields that store everything from price, to shipping options, to SKU numbers.

Custom post types store their data the same way as the built-in post types (posts, pages, attachments, etc.)—in the post and post_meta tables of the database. Using custom post types allows for easy integration with the vast number of WordPress plugins that you may want to use in your project.

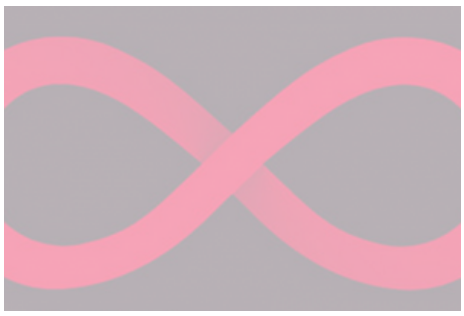
That said, the database structure of WordPress was not designed with large numbers of custom fields in mind. Not to mention it's difficult to integrate with other applications that are using the same database. For this reason, many plugin and app developers choose to add a completely separate database table for their service. But while doing so may increase performance and flexibility, it actually makes it harder to integrate the data in that table with WordPress plugins.

Whether you choose to use custom post types or a separate table before creating your own custom system for managing data, it's worth taking a look at a WordPress development framework plugins. These plugins, such as Custom Field Suite, Pods, Advanced Custom Fields, Easy Content Types, and Toolset, already have much of what you need to extend WordPress's database capabilities to fit your needs.

These types of frameworks, not only make advanced database management easier, but many of them also include tools for building user interfaces and addressing other front-end concerns. Each has a different set of tools, strengths, and weaknesses. However, if you can find the right framework (or frameworks) for managing custom content types and custom fields, you will save

Introducing Torque: Letter from the Editor


Wait! Don't Download That Theme Or Plugin



Taking a look at infinite scrolling: advantages and disadvantages

100 WordPress Influencers to Follow in 2015

countless hours of time developing your app.

### Content Management

One of WordPress's strengths is its admin, which is highly customizable—especially when using a tool like DP-Dashboard from DevPress. But, depending on your app, you might want to keep users out of the WordPress back-end entirely.

Making an app "feel like an app" is a difficult issue, but I think part of that process is keeping all user input as part of a consistent front-end experience.

One of the advantages to using WordPress to power your web app is that there are so many great solutions for generating front-end forms that automatically update the database. These options include Gravity Forms, Ninja Forms, and Caldera Forms.

TOP

## Creating Your App's API

### Do You Need an API?

An application protocol interface (API) is the tool that web sites and apps use to communicate with each other. If you want your app to integrate with another service, be it BaseCamp or healthcare.gov, your app will communicate with that service's API. Though you don't need to have your own API to integrate with other web services, it may make it easier. In the next section of this article, on front-end choices, I will discuss a few front-end options that exist outside of WordPress, but still connect with WordPress and the database via an API.

The real reason why you want to create an API for your app is so other services can integrate with it. There is no better way to make your offering an essential part of your user's workflow than to hook into what they are already doing. When you create something that works with a popular product, you have the opportunity to tap into their established user base.

### Options For Generating an API

The best part about creating your own API is that it's easy. There are two main options for generating your API: JetPack's JSON API and WP-API. Both are JSON RESTful APIs, which makes them very easy to work with and to integrate with other APIs that use the same very popular standard.

JetPack's JSON API allows you to set up a custom app that makes use of WordPress.com's oAuth2 service and REST API. Alternatively, WP-API, which is currently a plugin, but is on track to become a part of WordPress

core in version 4.1, offers a more decentralized approach.

The two APIs are fairly similar in functionality. While JetPack's API has been around longer, since reaching version 1.0 recently, WP-API is considered stable, and its developers have committed to ensuring that backwards-compatibility is maintained in future versions of the WP-API. The big difference between the two is that JetPack's API is hosted on WordPress.com's servers, while WP-API runs on your servers.

Off-loading some of your app's processing, by using the JetPack API, to a third-party may be a great way to increase performance and save money on server costs. Alternatively, using WP-API gives you more flexibility. This includes the ability to add custom endpoints and use the add-on plugins for WP-API. In addition, you can integrate with other WP-API add-on plugins that developers have already begun to develop in the weeks since the release of WP-API 1.0. Whether or not that flexibility is worth running the API on your server, when you could use WordPress.com's for free, is a decision that needs to be made on a per project basis.

# Front-End Considerations

## Beyond Traditional WordPress Themes

While WordPress's back-end content management and extensive ecosystem of plugins is unmatched, its front-end templating system wasn't built to meet the demands of modern web apps. WordPress has traditionally been centered around server-side processing via PHP, but it has been rapidly evolving to use more client-side (IE executed in the browser) JavaScript. This is especially true in the WordPress back-end, where most of the recent rewriting has used the Backbone and Underscores JavaScript libraries. Today, most apps, especially mobile apps, are client-side apps that minimize page loads.

Despite these advances, WordPress's theme system's core architecture is still based around the needs of a blog. Much of this has to do with the reliance on PHP for the templating system. One major disadvantage of WordPress themes is that they largely lack separation of concerns.

Separation of concerns is a principle in software development that states that different files should do different things. A typical WordPress template file—with its mix of PHP, HTML and JavaScript—violates this principle. In a design pattern such as Model View

Controller (MVC), a separate templating language is used so that HTML markup is separated from the code that prepares the data that the template displays. Timber is a good example of an implementation of using a templating language, in this case Twig, for WordPress themes.

There is no rule that says that a web app needs to be built around the MVC pattern, but there are many reasons why it's such a popular pattern in software design. There are several frameworks that replicate the MVC pattern using WordPress as the controller. These include ExoWP by New Clarity, and WPMVC. Using or replicating the MVC pattern in your app is not just a way to take advantage of this successful design pattern, it may also make it easier for developers that are used to working with Zend, Ruby on Rails or CakePHP to work on your WordPress-powered web app. This could potentially widen the available pool of developers that you can hire to work on your app.

Whether or not you choose to use a full MVC framework like the ones listed above, I can't overstate the importance of using a templating language to control your app's output. PHP is a very powerful language, but it is a notoriously terrible templating language. This is why templating languages like Twig and Handlebars are becoming more popular in frameworks for WordPress and other PHP platforms. With Twig or handlebars, you can stop opening and closing PHP tags in your templates, and create simple, reusable markup for your app.

**Separation Of Concerns – Before…**

```
$id = (int) $obj->ID();
$out = '<div class="group-view" id="group-view-'.$id.'">';
$out .= '<h3>'.holotree_link( $id, $type = 'post', $obj->field( 'post_title' ) ).'
if ( HT_DEV_MODE ) {
    $out .= '<span style="float:right">'.$id.'</span>';
}
$out .= $obj->field( 'group_description' );
$title = 'View '.$obj->field( 'post_title' );
$out .= '<br />'.holotree_link( $id, 'post', 'View', $title, true );
if ( holotree_group_class()->is_pending( get_current_user_id(), $id, $obj ) ) {
    $out .= $this->ui()->elements()->alert( __( 'Your Membership To This Group Is
}
elseif ( ! holotree_group_class()->is_member( $id, get_current_user_id(), $obj ) )

    $title = 'Click to Join '.$obj->field( 'post_title' );
    $append = array( 'action' => 'join-group', 'ID' => $id );
    $out .= ' '.holotree_link( site_url(), 'url', 'Join', $title,  true, false, fa
}

$out .= '</div>';
```

```
$id = $item[ 'id' ];
$out = '<div class="view-preview-'.$what.'-view" id="'.$what.'-view-' . $id . '"
$out .= '<h5>'.holotree_link( $id, 'post', $item[ 'post_title' ], $item[ 'post_t
$out .= '<div class="'.$what.'-status view-preview-status" id="'.$what.'' . $id
$out .= $item[ 'decision_status' ];
if ( HT_DEV_MODE ) {
    $out .= $id;
}
$out .= '</div><!--.view-preview-status-->';
if (  $item[ 'id' ]  !== $item[ 'change_to' ]['id']  ) {
    $out .= $this->change_to( $change_to = $item[ 'change_to' ] );
    if ( HT_DEV_MODE ) {
        $out .= $item[ 'change_to' ][ 'id' ] .'-' . $item[ 'id' ];
    }
}
$out .= '<div class="'.$what.'-description view-preview-description decision-des

$out .= $item[ 'decision_description' ];
$out .= '</div><!--.view-preview-description-->';
$out .= '<div class="'.$what.'-links view-preview-links">';
if ( is_singular( HT_DMS_DECISION_CPT_NAME  ) ) {
    $out .= $this->action_buttons( $what, $id );
}
else {
    $out .= holotree_link( $id, 'post', 'View', $item[ 'post_title' ], true );
}
$out .= '</div><!--.view-preview-links-->';
```

**... and After**

```
<div>
    <h3>{@post_title}</h3>
    <div class="details">
        <div>{@group_description}</div>
        <ul>
            <li>Organization: {@organiz
            <li>Members: {@members}</li
        </ul>
    </div>
    <div>
        [action buttons]
    </div>
</div>
```

```
<div class="decision">
    <h5>{@post_title}</h5>
    <div class="details">
        <div class="description">{@decision_desc
        <ul>
            <li>Status: {@decision_status}</li>
            <li>Manager: {@manager}</li>
            <li>Proposed By: {@proposed_by}</li>
        </ul>
    </div>
</div>
```

## Separate Front-End Clients

To address the shortcomings of the WordPress themes when used for app front-ends, there are a handful of the new themes and frameworks that take a different approach to interacting with WordPress, retrieving their data via the API. These tools use separate front-end clients that are powered by a JavaScript library such as Angular, Backbone or Node, and are better suited for modern web app development. In most cases, the theme retrieves its data from WordPress via a JSON API, such

as the JetPack API, or WP-API. For example, o2, the successor to the p2 theme that Automattic developed for its internal communication, is an example of a WordPress powered, client-side app built using backbone.js.

There are several strong choices to use as a starting point for creating a single page, client-side app. Roy Sivan has developed an anjular.js theme, reliant on WP-API, that he uses to power his site Coding Office Hours. Zack Tollman, has a backbone.js theme that he created to demonstrate how to build such a theme for his WP Sessions presentation on this topic. It is an excellent starting point for developing a backbone.js powered web app that integrates with core WordPress functionality.

For a complete example of how to run an entirely separate client-side app, linked to a WordPress backend running on a different server, see Bjoern Klose's angular-wordpress-pods. This system totally separates the front-end client from the backend, which should allow for easy scaling to multiple servers, with a central administration. The front-end client(s) are linked to WordPress using WP-API, and the Pods JSON API add-on for WP-API.

### UI Frameworks

Some of the resources I've listed in this section include a UI framework like Bootstrap or Foundation, while others will require you to build or include your own. UI frameworks (like Bootstrap, Foundation, Bourbon, and Gumby) provide useful, prebuilt CSS that's generated with a preprocessor (such as SASS or LESS), usually in combination with JavaScript plugins, to give you all of the UI tools you need.

Using a UI framework can save you a lot of time in development, especially in the early stages. However, here is a risk of ending up with too generic of an interface when using a popular UI framework. At the same time, there is also great virtue in being able to tap into these established design patterns that people commonly use when they interact with web and mobile apps. Creating a UI that is both unique and functional takes a talented designer, regardless if you start from scratch or a framework.

## Hosting & Deployment

In order for a web or mobile apps to be successful, it has to be fast. Budget hosting is not an option for these types of projects. However, there are plenty of hosting alternatives out there. When choosing a host, you will want to look for scalability, SSH access, ease of

deployments (including support for GIT or SVN,) and recent versions of PHP and MySQL.

Whether you want to choose a managed WordPress host like WP Engine or Page.ly, or use a cloud hosting service like Amazon Web Services, Digital Ocean, or Rackspace, depends on your needs and the skills of your team.

One challenge in developing, deploying, and testing an app is managing the dependencies of the 3rd-party libraries, plugins, and even WordPress that you use in your app. Keeping dependencies up to date manually, in your testing and production environments can quickly become unmanageable. Tools like Grunt, Composer, Capistrano, Puppet and Bower can save time and reduce errors when compared to manually managing third-party source code. They can also automate the process of building, optimizing, and deploying your app.

Bedrock, which is developed by Roots, is a great example of a complete WordPress stack incorporating dependency management with Composer, and Capistrano and virtual servers using Vagrant. You should also consider using an integrated source-code management and deployment service such as Beanstalk or codebase. These services offer version control, team and project management tools, and deployment tools.

# Going Mobile

## Native vs Non-native Apps

The programing languages that web developers typically use, such as PHP and JavaScript, are not compiled into machine code before they're deployed. That doesn't mean that they're not compiled before being executed. But rather that these types of languages are compiled, in real time, right before being executed by a server's processor.

Native apps, like most of the apps you use on your desktop, are written in languages that are compiled in advance. While native apps generally are much faster than non-native apps, a different version needs to be developed for each OS that they run on, which adds to their already high development cost.

Today, more and more mobile apps are being built as cross-platform HTML5 apps, relying on the same tools that you're used to using for creating web sites. Mozilla's FirefoxOS project based even does away with native apps entirely. Whether or not native apps or HTML 5 apps are the future is unclear. What is clear is that native

apps are expensive and time consuming to create. Furthermore Facebook, a web-app written largely in PHP, developed the open source "Just In Time" compiler HHVM to address the performance shortcomings of non-native web apps. When used with WordPress, HHVM can result in dramatic performance increases, further reducing the disadvantages of deploying your mobile app as an HTML5 app.

Building an app with WordPress and then using a service, such as Phone Gap, to turn it into an HTLM5 app is both faster and cheaper. Skipping the long and expensive process of native app development process, means that you can get your app out there without a huge capital investment. If it succeeds, you can decide if you want to use that success to raise the money to build a native app, knowing that your idea has been market-proven and what features your users actually want.

### Options for Creating Mobile Apps from WordPress Sites

There are multiple ways to turn your WordPress site or app into a mobile app that can be submitted to app stores. AppPresser is a toolset designed for optimizing a WordPress site or app before submitting it to Phone Gap. The tools AppPresser provides push notifications, touch gestures, and the ability to access the camera along with a theme designed especially for use as a mobile app. AppPresser also offers, a BuddyPress-powered social networking package as well as WooCommerce integration for creating eCommerce apps.

Alternatively, WPMUDEV's WP For iOS and WP For Android are based on WordPress mobile apps for iOS and Android. These services provide an alternate route into the iOS and Android app stores. Keep in mind that the WordPress mobile apps are all open source. A developer, with the right skills, could use them as starting points for developing your own custom mobile app, like WPMUDEV has done.

## Lots of Options, Endless Possibilities

The success of WordPress has given us plentiful options for creating web and mobile apps. These projects are not easy to pull off, but with planning, talented developers, and help from the right mix of tools like the ones I have listed in this article, they are achievable. Whether you are looking for a way to turn your great idea into the next "IT" app, or you are a developer looking for something special to offer your clients, it is exciting to know that the skills

you've learned developing WordPress sites can be transferred to web and mobile app development.

**Josh Pollock** started learning WordPress development when he was supposed to be working on his masters thesis, which ended up being about open source solutions for sustainable design and was presented in a WordPress site. You can learn more about him at JoshPress.net or follow him on twitter @Josh412

## Related

Mastering WordPress: 48 resources to go from newbie to pro

28 High-Quality Resources To Learn JavaScript For WordPress

Preparing Your WordPress Site to Power a Single Page Web App

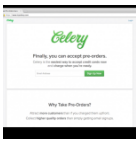
17 Of The Best Code Editors For WordPress Developers And Users

## Explore Torque

Daily Theme & Plugin Roundup - 6.19.2013

WordPress.org Rolls Out New Theme Tags

6 Tools That Make WordPress Migrations A Breeze




Pricing for Profit eBook via Chris Lema, James Dalman

Powered by

MORE WORDPRESS NEWS FROM TORQUE:

‹ How to Set up Disqus Comments on Your WordPress Blog

How to Introduce Social Proof to Your Site ›

## There are 21 comments

**Roy** · June 18, 2014 at 11:32 am
Thanks for the mention! If anyone doesn't want to hassle with my angular.js theme, but want to try out AngularJS… check out my plugin http://www.wordpress.org/plugins/angularjs-for-wp/ – it has angularJS directives and WP shortcodes to get you started
Reply

**Chuck Reynolds** · June 18, 2014 at 3:40 pm
nice post… I have an install script for HHVM and NGINX if it's helpful to add: https://github.com/chuckreynolds/hhnginx
Reply

chrisiufer · June 19, 2014 at 9:35 am
One especially difficult problem with WordPress is that it doesn't easily

facilitate a typical app dev > staging > production workflow. There are some plugins that help with migrating the database, and some fancy modification of wp-config can make this easier. Check out http://deliciousbrains.com/wp-migrate-db-pro/
Reply

**Mark Simchock** · April 2, 2015 at 5:25 am
See also Pantheon hosting.
Reply

**Bryan Robles** · June 19, 2014 at 9:54 pm
Good Article. Please note though that Node is not a front-end framework, it is used for server-side development, scripting, and even desktop development (brackets, atom).
Reply

Nikhil · September 24, 2014 at 8:45 pm
Good article. WordPress can be a powerful backend for mobile apps. Then there is also some services like http://www.mobapper.com which leverage it to automatically build mobile apps in the cloud.
Reply

Mark Suansing · September 29, 2014 at 7:29 pm
Yes this is really a good article. This mobile app feature is very important and reliable. It is very clear and excellent. http://www.triggerapp.com Mark Suansing
Reply

**Shawn** · October 14, 2014 at 6:11 am
I love using WordPress as an application framework. If you're going to do this, I highly recommend reading Building Web Apps with WordPress by Brian Messenlehner and Jason Coleman. Some great tips for doing exactly this.

http://www.amazon.ca/Building-Apps-WordPress-Brian-Messenlehner/dp/1449364071
Reply

**Adam Dartez** · November 6, 2014 at 5:03 pm
Great article bro! But as far as this statement is concerned

" Whether or not native apps or HTML 5 apps are the future is unclear"

I believe the writing is on the wall. Javascript is obviously nowhere near as powerful right now as say Java, which the WordPress Android app is built with https://github.com/wordpress-mobile/WordPress-Android, but the W3 has spoken, HTML5 (and essentially Javascript) are the future, the take over isn't if but when! Just look at how unbelievably easy Node.js is compared to writing a server in Java. I learned Javascript years ago and have only recently begun to write applications in Java exclusively for the sole reason of being able to develop native WordPress apps on top of the aforementioned project. But I tell you when I first started learning Java the compiling and build errors alone almost scared me away. Javascript has an unbelievable amount of catching up to do but it is so much easier than Java it's unreal. Anyway, thanks for the great article!
Reply

Veronica Wilson · January 29, 2015 at 2:27 am
The California-based tech giant has announced that the technology will also be available on the upcoming Apple Watch, meaning people will be able to manage their payments without even needing to take out their phone. But there are payment apps that can be very secure enough. These are now getting more popular.
Reply

**Mark Simchock** · April 2, 2015 at 5:31 am
For those who like to consider marketability when picking up new skills, I believe Drupal 8 – I know, I just used The D Word 🙂 – is moving to Twig for themes.
Reply

**Julien Renaux** · May 20, 2015 at 10:00 am
You can also create your own iOS/Android application for free with http://wphc.julienrenaux.fr/
Reply

**Julien Renaux** · May 29, 2015 at 10:06 pm
You can use http://wphc.julienrenaux.fr/ to build iOS and Android apps based on WordPress. It is Free and Open Source.

Reply

**Sean** · June 24, 2015 at 8:24 am
"Lots of Options, Endless Possibilities" indeed there are : ) Very useful info to get started with in this direction, thanks Josh!
Reply

**Kevin Koo** · September 13, 2015 at 9:51 am
I'm sorry to inform you that the WP for Android and WP for iOS apps are now retired. Good news: They are now free.
http://premium.wpmudev.org/blog/ios-android-retirement/
Reply

hamaricity · October 24, 2015 at 4:57 am
free
classified freeads

Hamaricity hamari
city classified free ads buy and sell
free advertisements your business company product website Buying and selling your item used and new
product find jobs post free your jobs and find jobs more Education Classes

Classified Free ads post buy and sell
Reply

Sunny Luthra · October 30, 2015 at 12:59 am
Recently used WordPress as a backend for my mobile and webapp. I wrote a custom rest api provider and added a redis layer for completely decoupling WordPress from the frontend. I also blogged about the process here http://snypd.com/wordpress-angular-redis-double-happiness/ [pretty new to blogging world] 🙂
Reply

Mehdi Abassi · December 14, 2015 at 6:15 am
dear sir i need ur help for a woocommerce plugin called SMT gateway WC_Mrova_SMT
Reply

Brad Klosterman · November 11, 2015 at 4:40 am
I would like to create an app that allows the user to login, fill in a form with their information and when it is complete the information is populated on different templates for them to print out, email and save. They can view their information on the different templates to find the graphic work that best fits their needs. Any suggestions on where to start?
Reply

**seojeek** · January 9, 2016 at 7:10 am
omg, that before code is awful. Just no…
Reply

Tinku · March 20, 2016 at 11:02 am
A great and unique post. This actually bridges modern web programming concepts with legendary web platform WordPress. This should help some one like me who is looking to dive deep in to WordPress with an experience built mostly around MVC/JavaScript frameworks.
Reply

Your email address will not be published. Required fields are marked *

**Comment**

**Name** * **Email** * **Website**

Continue the conversation via email!

Post Comment

TORQUE
A WP Engine Publication