Company Website

Software Development Documentation

Appwebtech

Developed and Designed by Joseph M Mwania for Appwebtech

Table of Contents

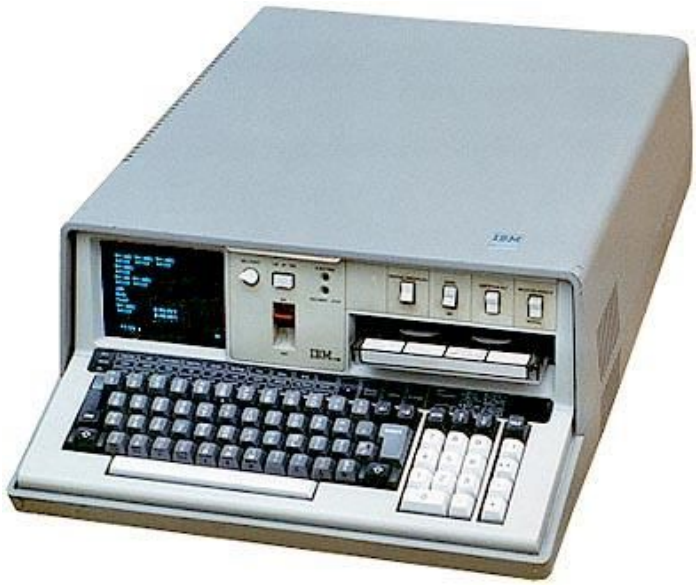
Introduction

Wordpress is a CMS that uses plugins, themes etc to enhance the functionality and usability within a website. It runs PHP in the BackEnd and HTML5 and CSS (no severside scrypting like SASS). Many bloggers do use wordpress due to it's versatility and UX/UI customizability.

Why did I write this documentation

In the past I used to code and push to GitHub without documentation, especially when I was creating Ruby on Rails apps. But then when people started writing and asking me if they could use it in their applications, etc, then I thought I should write some documentation.

Hopefully 20, 50, or even 80 years from now, people will read this and see how backward we were? (*Sorry for hitting reality on it's head*) but that is what kiddos think of the old tech like the 70's and 80's. Take a look at this IBM's revolutionary computer, the **IBM 5110**. It was the smallest and first portable computer weighing **28 kg's** and costed **$ 10,975**. It boasted a **64 kb of RAM** while I'm currently complaining my iMacs **32 GB of RAM** to be *somewhat* insufficient. See the IBM 5110 below.



Source: IBM

Who is my target audience

I figured out that the people who were asking if they could use my software were the people who needed help to know how the code worked. The rule of the thumb is; *if it's on GitHub, then it's open source and freely to use under the MIT license.* But after getting stuck, unseasoned developers are prompted to write to software owners and request help/opinion on how a piece of script would fit well in their projects.

So this documentation is targeted to developers and users who are curious to know the behind the scenes stuff on what made it possible for the application to come to be.

What does the documentation entail

The Scope of this documentation is to show in the **basic** way possible (without jargon) the steps I took in building this application. This is NOT a **tutorial**. It won't teach your anything about CMS's, MEAN Stack, DB's or what have you. Just to let you know before you get started (I value people's time :-).

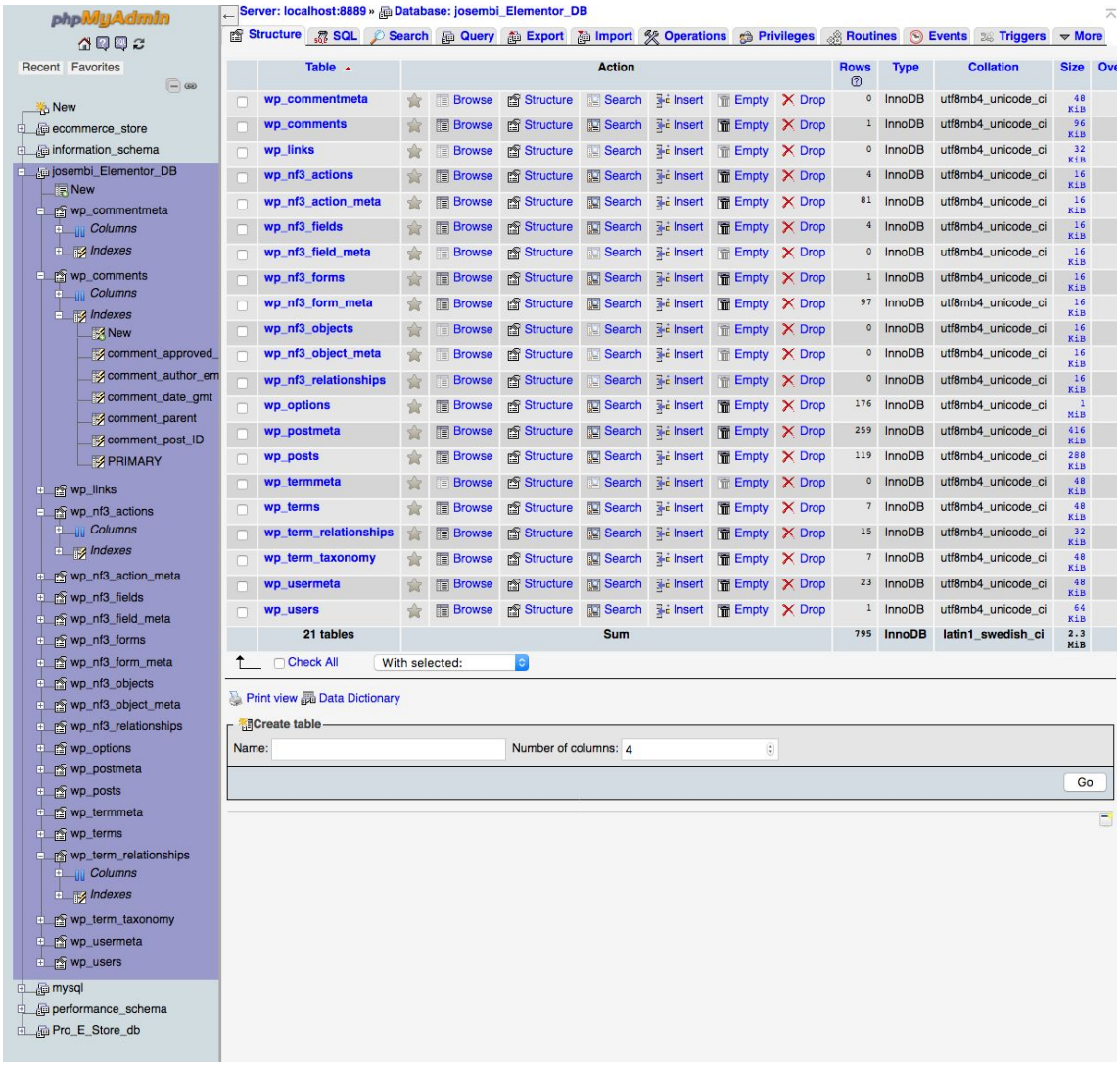
Wordpress Release

Wordpress was released in 2003 and since then it has evolved tremendously and now has become one of the best web technologies in building websites. At the time of writing, Wordpress powers close to 75 million websites. So what is Wordpress anyway?

Wordpress is an open-source Content Management System (CMS) that uses Hypertext Preprocessor (PHP) for its BackEnd (server-side) system and MySQL to manage it's databases (DB). PHP is a server-side scripting language which is amazing in web development; this is due to it's flexibility to be embedded in HTML, more or less the way Ruby is embedded in Rails. PHP is very versatile and compatible across many browsers, web servers and operating systems. The preference of using it for server-side systems has grown and will continue due to it's de-facto pro-bono policy. Same applies to MySQL.

Wordpress uses MySQL to manage it's DB's and PHP renders data from the server to the frontend. MySQL is an RDBMS and most of it's functionalities are open-source. It's a main component of LAMP open-source web application software stack.

Below is a snapshot of the DB I used. It can be queried, searched and exported to other servers. That is the main reason I used SoC to have a reputable external source to process payments. Actually, WooCommerce will not process payments but will direct them to be processed by Paypal.



There are a myriad of reason as to why I love using open-source software in design and development. Compelling advatanges gravitate towards the following factors which are quite intuitive;

- Security
- Customizability
- Freedom
- Auditability
- Ready Support (Bug fixes are timely)
- Try and discard if you dont like.
- Quality etc etc

Why Wordpress

Wordpress has somewhat become trendy in building CMS's for web publishing especially if the main goal is to have a website with pages, media, links and posts. A skilled developer can easily venture deep in the Wordpress ecosystem and use themes and plugins to suit his/her **needs**. Talking of needs, a bus cannot float on water unless it's well customised to function like a boat, like a case I saw in Budapest.

That out of the way, it's important to underscore the fact that complex applications will need complex technologies eg banks, hospitals, corporate companies, etc.

When it comes to themes, a developer can design one and sell to whomever wants to buy, eg themeforest, etc. Themes vary from free downloadables to a cost that would vary from $2, see here up to a cost way beyond $999, like this. It all depends really on what your website entails, your target audience, security features, design, etc.

Been an open-source technology, bugs within the CMS are fixed in a timely manner and any developer can contribute via a pull request and later submitting in GitHub.

Wordpress, Joomla and Drupal are currently the trending CMS systems in the planet at the time of writing. Their ability to deliver projects within a given time frame coupled with their versatility to use functionalities which reside in plugins have led to their growth and support globally.

Technologies used

Elementor

Elementor

Contributors: pojo.me, KingYes, ariel.k, jzaltzberg, yehudah, mati1000, pojosh
Tags: page builder, editor, elementor, builder, visual editor, front-end editor, design, drag and drop builder, front-end builder, landing page, page builder plugin, site builder, template builder, visual builder, website builder, wysiwyg
Requires at least: 4.4
Tested up to: 4.7
Stable tag: 1.0.10
License: GPLv3
License URI: https://www.gnu.org/licenses/gpl-3.0.html

Faster. Much faster.

Instant reaction was a key feature we wanted to achieve. Instant drag & drop, instant live edit, instant page load. The speed of Elementor is not matched by any other page builder, free or paid. This makes the interface fun and easy to work with, as well as reduces the time it takes to design.

Live design. Truly live.

Never again work on the backend and guess what the frontend will look like. With Elementor, you edit the page and simultaneously see exactly how it looks like. Elementor features live editing that is truly live, with no need to press update or go to preview mode.

Surprises you won't see anywhere else.

Create pages that have "the designer touch", by including unique features like box shadows, background overlays, hover and entrance animations, advanced buttons and more. We worked hard to create the perfect balance between full design capabilities and an intuitive and clean interface. You no longer need to use code, HTML, CSS or shortcode.

NEW! Template Library.

We've added **Library**, which includes a collection of 20+ beautiful templates, made for you by our top notch designers. You can also save your own pages and sections, and reuse them on different pages, or export them to whole different websites.

NEW! Mobile Editing.

Elementor comes with an exclusive toolset, that lets you create truly a responsive website in a whole new and visual way. From different font size per device, to reverse column ordering, this is the most powerful solution for creating perfect mobile pages.

28 widgets and counting

We packed 28 of the most useful widgets into Elementor. True, that's way more than we had to offer, but we wanted to spare no widget from you, so you can reach the top of your design capabilities.

WPGlobus for WPBakery Visual Composer

Contributors: alexgff, tivnetinc, tivnet
Donate link: http://www.wpglobus.com/
Tags: globalization, i18n, international, l10n, visual composer, localization, multilanguage, multilingual, translate, WPGlobus
Requires at least: 4.0
Tested up to: 4.2.3 **Stable tag:** trunk
License: GPLv2 **License URI:**
https://github.com/WPGlobus/wpglobus-for-wpbakery-visual-composer/blob/master/LICENSE

Description

WPGlobus for WPBakery Visual Composer is an extension to the WPGlobus plugin.

NOTE: You need to install and activate the WPGlobus Multilingual Plugin version 1.0.14 or later before installing the WPGlobus for WPBakery Visual Composer extension.

More info

- WPGlobus for WPBakery Visual Composer home page.
- GitHub code repository.

Installation

You can install this plugin directly from your WordPress dashboard:

1. Go to the *Plugins* menu and click *Add New*.
2. Search for *WPGlobus for WPBakery Visual Composer*.
3. Click *Install Now* next to the *WPGlobus for WPBakery Visual Composer* plugin.
4. Activate the plugin.

Alternatively, see the guide to Manually Installing Plugins.

Frequently Asked Questions

How do I contribute to WPGlobus for WPBakery Visual Composer?

We appreciate all contributions, ideas, critique, and help.

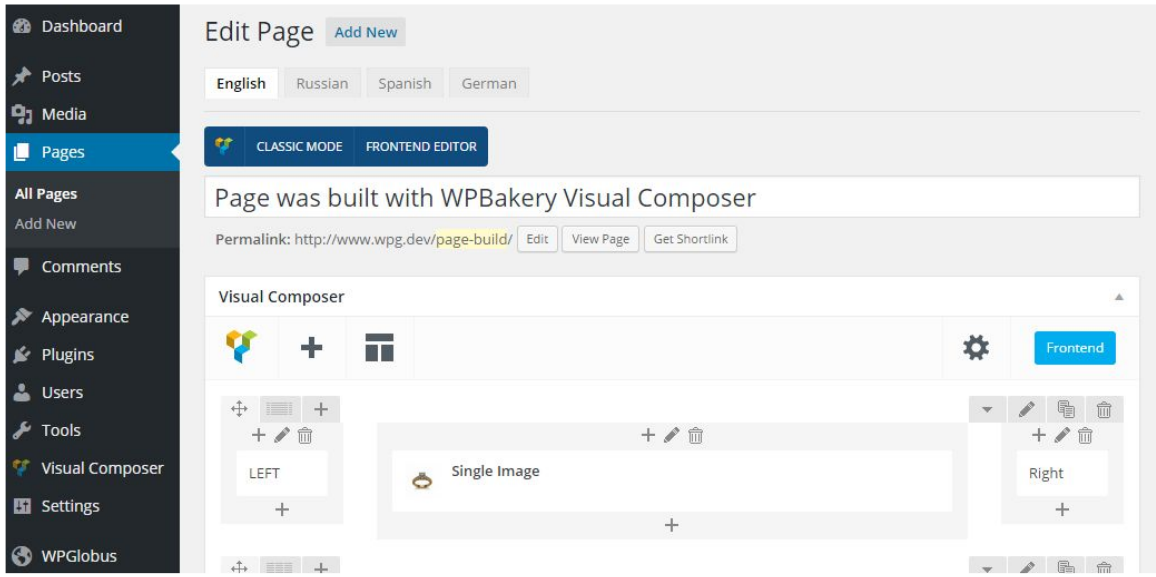
- To speed up our development, please report bugs, with reproduction steps on WPGlobus for WPBakery Visual Composer GitHub.
- Plugin and theme authors: please try WPGlobus for WPBakery Visual Composer and let us know if you find any compatibility problems.
- Contact us directly on WPGlobus.com.

More info?

Please check out the WPGlobus Website for additional information.

Screenshots

Plugin admin interface.



Changelog

1.0.0

- Initial release

Google Maps Widget

Description

Bring the world into your application, with the Google Maps widget!

Typical usage scenario

Giving a visual representation of an address. Showing an overview where all your branches/members/clients/orders/etc. are.

Features and limitations

Tap directly into all the power of Google Maps. Easy to implement. No route planning or advanced features yet.

Dependencies

Mendix 5.x Environment

Configuration

One of the most important things to keep in mind when implementing this widget is the Google Maps terms of use. These can be found here: Terms of Use. Chapter 9.1 is especially important.

```
    Free, Public Accessibility to Your Maps API Implementation. Your Maps API
Implementation must be generally accessible to users without charge. You may
require users to log in to your Maps API Implementation if you do not require
users to pay a fee. U nless you have entered into a separate written agreement
with Google or obtained Google's written permission , your Maps API Implementation
must not:      (a) require a fee-based subscription or other fee-based restricted
access; or     (b) operate only behind a firewall or only on an internal network
(except during the development and testing phase).
```

There are 3 use-cases for which this widget can be used.

```
    Outside a dataview: Will just retrieve the objects specified and show them on
the map.     Inside a dataview not matching the Objects property: Will show the
objects specified, can use '[%CurrentObject%]' in XPath Constraint.     Inside a
dataview matching the Objects property: Will show the objects specified, can NOT
use '[%CurrentObject%]'. Can set up the dataview to listen to a matching datagrid.
If 'Pan to context' is set to true, it will focus on the marker of the object that
is selected in the datagrid.
```

To finish up, just enter the correct values into the widget. For more information on the different input properties, read below.

Maps API Access Key

The Google Maps Javascript API v3 does not require an API key to function properly. However, I have loaded the Maps API using an APIs Console key which will allow the webmaster / site administrator to monitor the application's Maps API usage.

Authentications

Google JavaScript API do need authentications. I have a premium plan and thus enjoy two authentication options.

- **API key** set up in the Google Maps APIs
- **Client ID** instead of an API key. I always encourage clients to get a private API instead of using APIS's from a developer for many reasons which are beyond the scope of this documentation.

Many developers also turn out as site admins after delivering projects enjoying some special perks for website maintenance. Having a private API key, you can log in and see what's happening instead of relying on what someone else is saying. (Remember the old adage? **Trust but Verify.**)

Simple Light Box

Touch-friendly image lightbox for mobile and desktop with jQuery

Features

- responsive
- touchfriendly
- swipe gestures for next/previous image
- easy to install, easy to use
- minimalistic
- Only some css is included. You can change the style like you want!
- lots of options
- preloading next and previous image
- Android, iOs and Windows phone support
- CSS3 Transitions with fallback for older browsers
- Works in every modern Browser, even in IE 9+
- Can use jQuery 1.x,2.x and 3.x
- Keyboard support

Installation

```
//Bower bower install simplelightbox  //NPM npm install simplelightbox
```

Usage

Simple include simplelightbox.css and simple-lightbox.js to your page

```
var lightbox = $('.gallery a').simpleLightbox(options);
```

JavaScript Options

Property	Default	Type	Description
overlay	true	bool	show an overlay or not
spinner	true	bool	show spinner or not
nav	true	bool	show arrow-navigation or not
navText	['←','→']	array	text or html for the navigation arrows
captions	true	bool	show captions if availabled or not
captionSelector	'img'	string	set the element where the caption is. Set it to "self" for the A-Tag itself
captionType	'attr'	string	how to get the caption. You can choose between attr, data or text
captionsData	title	string	get the caption from given attribute
captionPosition	'bottom'	string	the position of the caption. Options are top, bottom or outside (note that outside can be outside the visible viewport!)
captionDelay	0	int	adds a delay before the caption shows (in ms)
close	true	bool	show the close button or not
closeText	'×'	string	text or html for the close button
swipeClose	true	bool	swipe up or down to close gallery
showCounter	true	bool	show current image index or not
fileExt	'png|jpg|jpeg|gif'	regexp or false	list of fileextensions the plugin works with or false for disable the check
animationSpeed	250	int	how long takes the slide animation
animationSlide	true	bool	weather to slide in new photos or not, disable to fade
preloading	true	bool	allows preloading next und previous images
enableKeyboard	true	bool	allow keyboard arrow navigation and close with ESC key
loop	true	bool	enables looping through images
rel	false	mixed	group images by rel attribute of link with same selector.

docClose	true	bool	closes the lightbox when clicking outside
swipeTolerance	50	int	how much pixel you have to swipe, until next or previous image
className:	'simple-lightbox'	string	adds a class to the wrapper of the lightbox
widthRatio:	0.8	float	Ratio of image width to screen width
heightRatio:	0.9	float	Ratio of image height to screen height
disableRightClick	false	bool	disable rightclick on image or not
disableScroll	true	bool	stop scrolling page if lightbox is opened
alertError	true	bool	show an alert, if image was not found. If false error will be ignored
alertErrorMessage	'Image not found, next image will be loaded'	string	the message displayed if image was not found
additionalHtml	false	string	Additional HTML showing inside every image. Usefull for watermark etc. If false nothing is added
history	true	bool	enable history back closes lightbox instead of reloading the page

Events

Name	Description
show.simplelightbox	this event fires before the lightbox opens
shown.simplelightbox	this event fires after the lightbox was opened
close.simplelightbox	this event fires before the lightbox closes
closed.simplelightbox	this event fires after the lightbox was closed
change.simplelightbox	this event fires before image changes
changed.simplelightbox	this event fires after image was changed
next.simplelightbox	this event fires before next image arrives
nextDone.simplelightbox	this event fires after next image was arrived
prev.simplelightbox	this event fires before previous image arrives
prevDone.simplelightbox	this event fires after previous image was arrived
nextImageLoaded.simplelightbox	this event fires after next image was loaded (if preload activated)
prevImageLoaded.simplelightbox	this event fires after previous image was loaded (if preload activated)
error.simplelightbox	this event fires on image load error

Example

```
$('.gallery a').on('open.simplelightbox', function () {   // do something… });
$('.gallery a').on('error.simplelightbox', function (e) {   console.log(e); //
some usefull information });
```

Public Methods

Name	Description
open	Opens the lightbox with an given jQuery Element
close	Closes current openend Lightbox
next	Go to next image
prev	Go to previous image
destroy	Destroys the instance of the lightbox
refresh	Destroys and reinitilized the lightbox, needed for eg. Ajax Calls, or after dom manipulations

Example

```
var gallery = $('.gallery a').simpleLightbox();  gallery.next(); // Next Image
```

Multiple Lightboxes on one page

You can have multiple lightboxes on one page, if you give them different selectors. Here is a small example:

```
var lightbox1 = $('.lighbox-1 a').simpleLightbox(); var lightbox2 = $('.lighbox-2 a').simpleLightbox();
```

PHP

This website uses PHP as it's backend system for it's database and server client functionalties. The beauty of PHP is it's ability to be embedded with HTML. The server in this website uses CGI protocol to process PHP code and render the website functionalities.

jQuery

jQuery is a Javascript library that enhances script composition from the designer/developer coding environment. It's super fast and amazing at event handling, animation, document traversal, Ajax and if you write API's constantly, then jQuery can come in handy instead of using vanilla javascript.

It's DOM element selection has lead to insights in creating other javascript frameworks tailored at specific needs. It has many contributors with over 6000 commits on it's GitHub page at the time of writing, and big multinationals like Nokia and Microsoft bundle it on their platforms eg ASP.NET, AJAX, Visual Studio etc.

Using jQuery to code has been very rewarding as it solves many a problem with less pain saving a lot of time and thus enhancing productivity in the long run.

Photoshop

It's no doubt that Adobe Photoshop remains the best raster graphics editor on the planet. It's ability to use RGB, CMYK, duotone, etc coupled with the great variety of image extensions makes it the best option to edit images professionally.

I have extensive experience in PSD design even prior to my web development career. I used it in this project for image sizing and for enhancing vivid display accross platforms from smart tv's to smart phones. I have targeted large images for big displays like desktops whilst small displays have images of reduced dimensions in pixels.

I tweak images to ensure that pixels are not lost when switching viewports and that the image extensions like **.jpeg** and **.png** are well saved to curb image degradation.

Deployment instructions

Deploy to a server which supports Wordpress like Godaddy, HostGator etc. HostGator is more speedy in loading pages imo than other websites.

In most cases I code locally and later upload to my cPanel. I will try andupload this website and fight the urge of pulling it down. (For some reason many people buy demo websites and request that they get pulled down. Let's cross fingures to thwart that.)

Developer

Joseph M Mwania

Contacts

http://www.theappwebtech.com/

https://github.com/appwebtech

https://twitter.com/appwebtech

https://www.facebook.com/theappwebtech/

https://it.pinterest.com/appwebtech/

License