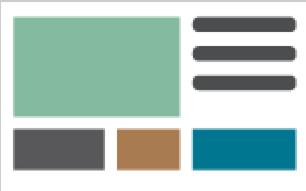
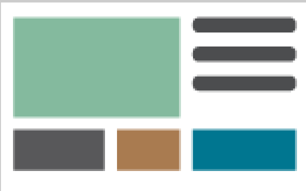
DANG designs Docs

My Active Projects

Gadgets can only be viewed in browsers



Gadgets can only be viewed in browsers



Meeting Notes & Brainstorm

- July 10, 2020 Meeting Notes
- 7/31/20 Meeting Agenda & Notes
- July 13, 2020 Meeting Notes

General Documentation

- Crypto/Blockchain FAQ
- GitHub Documentation
    - GitHub Dang Workflow Documentation
    - How to Use .gitignore
    - Version Control Process
- Edge-Caching - Cost & Implementation Analysis
- Event & Website Server Time Zone Locations
- Migrating Fishbowl Server to Windows Server
- New Hire Onboarding Software Introduction
- RICS Software - API Info Documentation
- RICS Software (WordPress) VS. RT (Magento 2.0)
- Social Media Image Sizes
- Sunflora - Affiliate Groups & Credits
- SunFlora PO Box Update
- Windows Server GCP VM Business Plan
- WHMCS (Web Host Manager Complete Solution)
    - WHMCS Support Ticket Setup
- Yootheme ZOO-AMP Pages

Standard Operating Procedures

- 8OVE Product Store Approval (DRAFT)
- AVAST Business Portfolio Instructional Documents & Video
- Adding A New Customer Service Affiliate (cs.sunflora.org)
- Adding a New Location in Reputation Manager (DRAFT)
- Checking Out a Cart Using the Dashboard Form (DRAFT)
- Clearing A Full Cart From The Dashboard (Draft)
- Creating a Short-Print URL with Redirection (DRAFT)
- Confirming a Store is Open When No Response is Given
- Creating a Store Signature for GMAIL (DRAFT)
- Creating a LinkedIn Page for Your CBD Store Locations (DRAFT)
- Database Backup Process (SOP)
- DANG designs Project Management (JIRA Tasks)
- DANG Client Website Off-boarding
- DANG-designs Antivirus Client Install
- Establishing a Product in Magento for Testing & Live Sites
- Formatting Rules For Addresses
- Google Email Data Migration (SOP)
- Google Takeout Process (DRAFT)
- Google Username & Password Security Update Procedure (SOP) -- QUASI-DRAFT
- Processing Sunflora Credits (DRAFT)
- Responding to Reviews
- Reviewing Fraud Transactions in the Admin Dashboard (DRAFT)
- SEO Friendly Web Design Guidelines
- Setting up a BrightLocal Location (DRAFT)

Navigation Guides

- Java Selenium Automation Projects
    - Create A List of Write-A-Review Links For Stores
    - List All Website Expiration Dates
    - Setting Up GTMetrix Website Testing Automation
    - Setting Up Pingdom Website Testing Automation
    - Setting Up Screenshots of All Web Pages
    - Setting Up Web Dev Website Testing Automation
    - Test All Subdomains For Issues Or Non-matching References
    - Website A HREF List Testing
- PHP Projects
    - 360twin - XML URL Creation For Google Search Sitemap
    - Billing.dang-designs - Call Logs Widget
    - Billing.dang-designs - ForceEveryoneToLogin
    - Billing.dang-designs - GoogleCalendarToCRM
    - Billing.dang-designs - CRMToGoogleCalendar
    - Billing.dang-designs - _Self Opening Link Hook
    - Billing.dang-designs - Send Monthly Report Emails
    - Billing.dang-designs - Update CRM with RingCentral Call Logs/Missed Calls
    - Capitalcityshoes - Add Customization To Item Description (Ariba)
    - Capitalcityshoes - Cookies
    - Capitalcityshoes - DMS Users Update Store ID
    - Capitalcityshoes - Import Users View & Credit Update
    - Capitalcityshoes - Insert Inventory Missing Rows
    - Capitalcityshoes - Send Activation Email Automation
    - CBDRX4U - Create Folders and .htaccess Files With List
    - CBDRX4U - Update All HTML Files To Replace TM with R
    - CBDRX4U - Update Zoo Item Links In Database
    - Mywatersidesouth - Forgot Username
    - Mywatersidesouth - Forgot Password
    - Mywatersidesouth - Reset Password
    - SunFlora - Setting Up Cloud Storage Bucket From PHP Script
        - Operating System Specifics
    - SunFlora - 3PL API Import
- JavaScript Projects
    - Set Store Image Via Cookie (Delayed & NOT In Use)
    - How To Create A Custom Input Field Through JavaScript
- WordPress Projects
    - 360twin - Universal Checkbox Confirmation

- Fishbowl
    - Fishbowl SQL Errors & Solutions
    - How To Restore A Database Backup On Fishbowl Server
- How To Block Malicious Bots From Servers
- How To Connect A Fishbowl Client Remotely - DANG designs Users
- How To Connect To API End Point Using Insomnia
- How To Correct Anacron Job Email Issues AND Other Linux Issues
- How To Correct Web.dev Listed Issues

- How To Create An Instance From A Snapshot (GCP)
- How to Create Social Media Open Graphs For Any Website Using Joomla.
- How To Perform Better Google Searches
- How To Set Up Enhanced Ecommerce Conversion Tracking (GA + GTM ONLY)
- How To Set Up New Jira Projects
- How To Set Up SSH Through FileZilla
- Setting Up .gitignore File For Repository
- How To Set Up Magento 2 Site
- How To Migrate Data From Magento 1 To Magento 2
- How To Set Up Source Report for Data Studio (360twin)
- Removing Foreign Characters Within Website Text
- Injecting An Admin User In Joomla Via phpMyAdmin
- Injecting An Admin User in WordPress Via phpMyAdmin
- Setting Up Ubuntu Server on Google Cloud VM
    - Download MySQL Database & Workbench In Ubuntu
- Setting Up Load Balancing Between Instances
- Setting Up Log File Containing Every Update To GitHub
    - Getting WordPress (SimpleHistory) Logs
    - Getting Joomla Logs
    - Getting Magento Logs
    - Getting Code Logs
- Setting Up Local MAMP Environment
    - Setting Up WordPress Site
    - Setting Up Joomla Site
- Setting Up A Batch File To Transfer Backups And Delete Them
- Setting Up A Cron Job
- Setting Up A CDN For Joomla Site
- Setting Up WordPress/Joomla Site On Alternative Domain
- Setup PayPal Auto Return for Website Payments
- Setting Up Windows Server 2019
    - Setting Up Fishbowl/MySQL Workbench on Windows Server
- Website @media Responsive Queries
- Where To Update IP Addresses On Change

Testing Guides

- 360 Twin Testing Guide | Wordpress Ecommerce
- Overview for Joomla and WordPress
- SUNFLORA.ORG Website Testing Protocol
- Test Orders On Capitalcityshoes Sandbox

DANG Dev Team Process Development

- Client On Boarding Process
- Build Out
    - Web Development Checklist
    - Wordpress Site Quality List
    - Joomla Website Quality List
- Confluence Content Ideas
    - Employee Handbook Template
    - Decision Tree Template

Information Required To Quote App

- Google Script Apps

General Documentation

This location will store all general documentation information that doesn’t necessarily meet a specific category.

- Crypto/Blockchain FAQ
- GitHub Documentation

- Edge-Caching - Cost & Implementation Analysis
- Event & Website Server Time Zone Locations
- Migrating Fishbowl Server to Windows Server
- New Hire Onboarding Software Introduction
- RICS Software - API Info Documentation
- RICS Software (WordPress) VS. RT (Magento 2.0)
- Social Media Image Sizes
- Sunflora - Affiliate Groups & Credits
- SunFlora PO Box Update
- Windows Server GCP VM Business Plan
- WHMCS (Web Host Manager Complete Solution)
- Yootheme ZOO-AMP Pages

Crypto/Blockchain FAQ

Blockchain

How blockchains work

A blockchain is a shared database that is managed by a global network of computers. Information held in the database is distributed and continually reconciled by the computers in the network. The computers are often referred to as *nodes, miners,* or *peers.* Whatever they're called, the computers are creating and maintaining their blockchain by validating and transmitting entries. And *entries* are the data that is published by the users of the network.

Often this data represents the movement of cryptocurrency from one user of the network to another user of the network.

For example, when you send some Bitcoin to your friend Tom, you're creating and publishing an entry into the Bitcoin network. In this example, the entry you created has some restrictions. The computers in the Bitcoin network will check to make sure that you haven't already sent the data representing the cryptocurrency to another person earlier. When you send Tom the Bitcoins, his account is credited, and your account is debited.

The computers in the network are preventing you from "double-spending." In the case of the Bitcoin network, the problem is solved by requiring every computer in the network to keep a complete record of the history of all the entries made within the network. The entire history gives the balance of every account, including yours.

Most blockchains are not controlled by any single entity and do not have a single point of failure. All the entries are viewable by the whole network. When data is entered into a blockchain, it can't be removed. It's there forever.

The innovation of blockchains that makes them different from a normal database is that they achieve agreement on the history by sharing and restricting the entries without a central server or authority.

Useful Links:
The Blockchain explained
Blockchain.com

Cryptocurrencies

What is cryptocurrency?

You know how your every day, government-based currency is reserved in banks?
And that you need an ATM or a connection to a bank to get more of it or transfer it to other people? Well, with cryptocurrencies, you may be able to get rid of banks and other centralized middlemen altogether. That's because cryptocurrencies rely on a technology called *blockchain,* which is *decentralized* (meaning no single entity is in charge of it). Instead, every computer in the network confirms the transactions.

The definition of money

Before getting into the nitty-gritty of cryptocurrencies, you need to understand the definition of money itself. The philosophy behind money is a bit like the whole "which came first: the chicken or the egg?" thing. In order for money to be valuable, it must have a number of characteristics, such as the following:

- Enough people must have it.
- Merchants must accept it as a form of payment.
- Society must trust that it's valuable and that it will remain valuable in the future.

Of course, in the old days, when you traded your chicken for shoes, the values of the exchanged materials were inherent to their nature. But when coins, cash, and credit cards came into play, the definition of money and, more importantly, the trust model of money changed.

Another key change in money has been its ease of transaction. The hassle of carrying a ton of gold bars from one country to another was one of the main reasons cash was invented. Then, when people got even lazier, credit cards were invented. But credit cards carry the money that your government controls. As the world becomes more interconnected and more concerned about authorities who may or may not have people's best interests in mind, cryptocurrencies may offer a valuable alternative.

Here's a fun fact: Your normal, government-backed currency, such as the U.S. dollar, must go by its fancy name, *fiat currency,* now that cryptocurrencies are around. Fiat is described as a legal tender like coins and banknotes that have value only because the government says so.

Simply stated, *cryptocurrency* is a new form of digital money. You can transfer your traditional, non-crypto currency money like the U.S. dollar digitally, but that's not quite the same as how cryptocurrencies work. When cryptocurrencies become mainstream, you may be able to use them to pay for stuff electronically, just like you do with traditional currencies.

However, what sets cryptocurrencies apart is the technology behind them. You may say, "Who cares about the technology behind my money? I only care about how much of it there is in my wallet!" The issue is that the world's current money systems have a bunch of problems. Here are some examples:

- Payment systems such as credit cards and wire transfers are outdated.
- In most cases, a bunch of middlemen like banks and a broker takes a cut in the process, making transactions expensive and slow.
- Financial inequality is growing around the globe.
- Around 3 billion unbanked or underbanked people can't access financial services. That's approximately half the population on the planet!

Cryptocurrencies aim to solve some of these problems, if not more.

Useful Links:
Bitcoin Explained
Cryptocurrency For Beginners

Staking

What is crypto staking?

Crypto staking is similar to depositing money in a bank, in that an investor locks up their assets, and in exchange, earns rewards, or "interest."

Staking is a term used to refer to the delegating of a certain number of tokens to the governance model of the blockchain and thus locking them out of circulation for a specified length of time.

A particular network's protocol locks up an investor's holdings — similar to depositing money in a bank and agreeing not to withdraw it for a set time period, which benefits the network in a couple of ways.

First, this can increase the value of a token by limiting the supply. Second, the tokens can be used to govern the blockchain if the network uses a proof-of-stake (PoS) system. A PoS system — as opposed to a proof-of-work (PoW) one, which incorporates "mining" — can be fairly complicated, especially for crypto newcomers.

In PoS systems, coins are staked to forge new blocks in the blockchain, for which participants are rewarded. Winners are selected through randomization, ensuring no single entity will gain a monopoly over forging,

The process is simplified for crypto exchange users. Staking is as easy as "going to the staking page [on the user's interface], specifying the amount you want to stake, and hitting submit."

Setting up a staking system on your own can be quite difficult. You need to maintain and run a node yourself. And you need to know the crypto's infrastructure, which may require background knowledge many investors won't have.

Depending on how much of their total holdings are being staked, and the length that they're being staked for, a staker can earn a proportional reward by forging. Stakers can also pool their holdings to meet any required minimums, too, into a "staking pool." It's also possible to "cold stake" on some networks, which involves staking coins or tokens that are held in a "cold" wallet, or one that is kept offline.

While not every cryptocurrency can be staked, most can. For instance, seven of the ten most popular current coins can be staked. Here are some examples:

- **Cardano:** Investors can also delegate Ada — the Cardano network's cryptocurrency — to staking pools to earn rewards. Cardano users can even set up their own staking pools, too, assuming they have the technical know-how to create and administer one.
- **Ethereum:** Previously employed a PoW system, Ethereum is now moving to PoS. To stake Ethereum on your own, you'll need a minimum of 32 ETH to become a validator, and you'll then "be responsible for storing data, processing transactions, and adding new blocks to the blockchain.
- **Solana**: Solana, or SOL, can likewise be staked or delegated to a staking pool, assuming an investor uses a digital wallet that supports it. From there, it's a matter of selecting a validator and deciding how much you'd like to stake.

Useful Link:
https://en.wikipedia.org/wiki/Proof_of_stake

NFT's

What is an NFT? What does NFT stand for?

Non-fungible token.

That doesn't make it any clearer.

"Non-fungible" more or less means that it's unique and can't be replaced with something else. For example, a bitcoin is fungible — trade one for another bitcoin, and you'll have exactly the same thing. A one-of-a-kind trading card, however, is non-fungible. If you traded it for a different card, you'd have something completely different. You gave up a Squirtle, and got a 1909 T206 Honus Wagner, which *StadiumTalk* calls "the Mona Lisa of baseball cards."

How do NFTs work?

At a very high level, most NFTs are part of the Ethereum blockchain. Ethereum is a cryptocurrency, like bitcoin or dogecoin, but its blockchain also supports these NFTs, which store extra information that makes them work differently from, say, an ETH coin. It is worth noting that other blockchains can implement their own versions of NFTs. (Some already have.)

What's worth picking up at the NFT supermarket?

NFTs can really be anything digital (such as drawings, music, your brain downloaded and turned into an AI), but a lot of the current excitement is around using the tech to sell digital art.

You mean, like, people buying my good tweets?

I don't think anyone can stop you, but that's not really what I meant. A lot of the conversation is about NFTs as an evolution of fine art collecting, only with digital art.

The founder of Twitter sold one for just under $3 million

Do people really think this will become like art collecting?

I'm sure some people really hope so — like whoever paid almost $390,000 for a 50-second video by Grimes or the person who paid $6.6 million for a video by Beeple. Actually, one of Beeple's pieces was auctioned at Christie's.

Sorry, I was busy right-clicking on that Beeple video and downloading the same file the person paid millions of dollars for.

Wow, rude. But yeah, that's where it gets a bit awkward. You can copy a digital file as many times as you want, including the art that's included with an NFT.

But NFTs are designed to give you something that can't be copied: ownership of the work (though the artist can still retain the copyright and reproduction rights, just like with physical artwork). To put it in terms of physical art collecting: anyone can buy a Monet print. But only one person can own the original.

No shade to Beeple, but the video isn't really a Monet.

What do you think of the $3,600 Gucci Ghost? Also, you didn't let me finish earlier. That image that Beeple was auctioning off at Christie's ended up selling for $69 million, which, by the way, is $15 million more than Monet's painting *Nymphéas* sold for in 2014.

I think I remember hearing that NFTs are already over. Didn't the boom go bust?

But surely you've heard of penguin communities?

Penguin communities?

Right, so... people have long built communities based on things they own, and now it's happening with NFTs. One community that's been exceedingly popular revolves around a collection of NFTs called Pudgy Penguins, but it's not the only community built up around the tokens. It could be argued that one of the earliest NFT projects, CryptoPunks, has a community around it, and there are other animal-themed projects like the Bored Ape Yacht Club that have their own clique.

Of course, the communal activities depend on the community. For Pudgy Penguin or Bored Ape owners, it seems to involve vibing and sharing memes on Discord, or complimenting each other on their Pudgy Penguin Twitter avatars.

What's the point of NFTs?

That really depends on whether you're an artist or a buyer.

I'm an artist.

First off: I'm proud of you. Way to go. You might be interested in NFTs because it gives you a way to sell work that there otherwise might not be much of a market for. If you come up with a really cool digital sticker idea, what are you going to do? Sell it on the iMessage App Store? No way.

Also, NFTs have a feature that you can enable that will pay you a percentage every time the NFT is sold or changes hands, making sure that if your work gets super popular and balloons in value, you'll see some of that benefit.

I'm a buyer.

One of the obvious benefits of buying art is it lets you financially support artists you like, and that's true with NFTs (which are way trendier than, like, Telegram stickers). Buying an NFT also usually gets you some basic usage rights, like being able to post the image online or set it as your profile picture. Plus, of course, there are bragging rights that you own the art, with a blockchain entry to back it up.

No, I meant I'm a *collector*.

Ah, okay, yes. NFTs can work like any other speculative asset, where you buy it and hope that the value of it goes up one day, so you can sell it for a profit. I feel kind of dirty for talking about that, though.

So every NFT is unique?

In the boring, technical sense that every NFT is a unique token on the blockchain. But while it could be like a van Gogh, where there's only one definitive actual version, it could also be like a trading card, where there's 50 or hundreds of numbered copies of the same artwork.

Who would pay hundreds of thousands of dollars for what basically amounts to a trading card?

Well, that's part of what makes NFTs so messy. Some people treat them like they're the future of fine art collecting (read: as a playground for the mega-rich), and some people treat them like Pokémon cards (where they're accessible to normal people but also a playground for the mega-rich). Speaking of Pokémon cards, Logan Paul just sold some NFTs relating to a million-dollar box of the—

Yeah, he sold NFT video clips, which are just clips from a video you can watch on YouTube anytime you want, for up to $20,000. He also sold NFTs of a Logan Paul Pokémon card.

Are NFT's mainstream now?

It depends on what you mean. If you're asking if, say, my mom owns one, the answer is no.

But we have seen big brands and celebrities like Marvel and Wayne Gretzky launch their own NFTs, which seem to be aimed at more traditional collectors, rather than crypto-enthusiasts. While I don't think I'd call NFTs "mainstream" in the way that smartphones are mainstream, or *Star Wars* is mainstream, they do seem to have, at least to some extent, shown some staying power even outside of the cryptosphere.

But what does The Youth think of them?

Ah yes, excellent question. Stay interested in what the next generation is doing, and it certainly does seem like some of them have been experimenting with NFTs. An 18-year-old who goes by the name FEWOCiOUS says that his NFT drops have netted over $17 million — obviously, most haven't had the same success. *The New York Times* talked to a few teens in the NFC space, and some said they used NFTs as a way to get used to working on a project with a team or to just earn some spending money.

Can I buy this article as an NFT?

No, but technically anything digital could be sold as an NFT (including articles from *Quartz* and *The New York Times*, provided you have anywhere from $1,800 to $560,000). Deadmau5 has sold digitally animated stickers. William Shatner has sold Shatner-themed trading cards (one of which was apparently an X-ray of his teeth).

Gross. Actually, could I buy someone's teeth as an NFT?

There have been some attempts at connecting NFTs to real-world objects, often as a sort of verification method. Nike has patented a method to verify sneakers' authenticity using an NFT system, which it calls CryptoKicks. But so far, I haven't found any teeth, no. I'm scared to look.

Look? Where?

There are several marketplaces that have popped up around NFTs, which allow people to buy and sell. These include OpenSea, Rarible, and Grimes' choice, Nifty Gateway, but there are plenty of others.

I've heard there were kittens involved. Tell me about the kittens.

NFTs really became technically possible when the Ethereum blockchain added support for them as part of a new standard. Of course, one of the first uses was a game called CryptoKitties that allowed users to trade and sell virtual kittens. Thank you, internet.

At least it's not digital pet rocks... right?

In fact, there are people who are spending tens or hundreds of thousands of dollars on NFT pet rocks (the website for which says that the rocks serve no purpose other than being tradable and limited).

Could I pull off a museum heist to steal NFTs?

That depends. Part of the allure of blockchain is that it stores a record of each time a transaction takes place, making it harder to steal and flip than, say, a painting hanging in a museum. That said, cryptocurrencies have been stolen before, so it really would depend on how the NFT is being stored and how much work a potential victim would be willing to put in to get their stuff back.

Note: Please don't steal.

Should I be worried about digital art being around in 500 years?

Probably. Bit rot is a real thing: image quality deteriorates, file formats can't be opened anymore, websites go down, people forget the password to their wallets. But physical art in museums is also shockingly fragile.

I want to maximize my blockchain use. Can I buy NFTs with cryptocurrencies?

Yes. Probably. A lot of the marketplaces accept Ethereum. But technically, anyone can sell an NFT, and they could ask for whatever currency they want.

Will trading my Logan Paul NFTs contribute to global warming and melt Greenland?

It's definitely something to look out for. Since NFTs use the same blockchain technology as some energy-hungry cryptocurrencies, they also end up using a lot of electricity. There are people working on mitigating this issue, but so far, most NFTs are still tied to cryptocurrencies that generate a lot of greenhouse gas emissions. There have been a few cases where artists have decided to not sell NFTs or to cancel future drops after hearing about the effects they could have on climate change. Thankfully, one of my colleagues has really dug into it, so you can read this piece to get a fuller picture.

Can I build an underground art cave / bunker to store my NFTs?

Well, like cryptocurrencies, NFTs are stored in digital wallets (though it is worth noting that the wallet does specifically have to be NFT-compatible). You could always put the wallet on a computer in an underground bunker, though.

What if I wanted to watch a TV show that's somehow related to NFTs?

Believe it or not, you have options! Steve Aoki is working on a show based on a character from a previous NFT drop, called *Dominion X*. The show's site says that it'll be an episodic series launched on the blockchain (the first short video is on OpenSea), and there are hundreds of NFTs already associated with the show.

There's also a show called *Stoner Cats* (yes, it's about cats that get high, and yes it stars Mila Kunis, Chris Rock, and Jane Fonda), which uses NFTs as a sort of ticket system. Currently, there's only one episode available, but a *Stoner Cat* NFT (which, of course, is called a TOKEn) is required to watch it.

Useful Link:
https://en.wikipedia.org/wiki/Non-fungible_token

Mining

What Is Crypto Mining?

Most people think of crypto mining simply as a way of creating new coins. Crypto mining, however, also involves validating cryptocurrency transactions on a blockchain network and adding them to a distributed ledger. Most importantly, crypto mining prevents the double-spending of digital currency on a distributed network.

Like physical currencies, when one member spends cryptocurrency, the digital ledger must be updated by debiting one account and crediting the other. However, the challenge of a digital currency is that digital platforms are easily manipulated. Bitcoin's distributed ledger, therefore, only allows verified miners to update transactions on the digital ledger. This gives miners the extra responsibility of securing the network from double-spending.

Meanwhile, new coins are generated to reward miners for their work in securing the network. Since distributed ledgers lack a centralized authority, the mining process is crucial for validating transactions. Miners are, therefore, incentivized to secure the network by participating in the transaction validation process that increases their chances of winning newly minted coins.

In order to ensure that only verified crypto miners can mine and validate transactions, a proof-of-work (PoW) consensus protocol has been put into place. PoW also secures the network from any external attacks.

Proof-of-Work

Crypto mining is somewhat similar to mining precious metals. While miners of precious metals will unearth gold, silver, or diamonds, crypto miners will trigger the release of new coins into circulation. For miners to be rewarded with new coins, they need to deploy machines that solve complex mathematical equations in the form of cryptographic hashes. A hash is a truncated digital signature of a chunk of data. Hashes are generated to secure data transferred on a public network. Miners compete with their peers to zero in on a hash value generated by a crypto coin transaction, and the first miner to crack the code gets to add the block to the ledger and receive the reward.

Each block uses a hash function to refer to the previous block, forming an unbroken chain of blocks that leads back to the first block. For this reason, peers on the network can easily verify whether certain blocks are valid and whether the miners who validated each block properly solved the hash to receive the reward.

Over time, as miners deploy more advanced machines to solve PoW, the difficulty of equations on the network increases. At the same time, competition among miners rises, increasing the scarcity of cryptocurrency as a result.

Useful Link:
WhatToMine

Documentaries

Introduction to Bitcoin (Very Informative)
Inside The Cryptocurrency Revolution
Banking on Bitcoin
The Rise and Rise of Bitcoin
The End of Money As We Know It

Misc Info

Bitcoin Wiki
Cryptocurrency Prices
Crypto Lending

GitHub Documentation

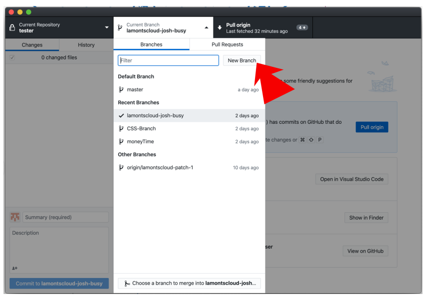
This will serve as the documentation destination for all GitHub related information.

GitHub Dang Workflow Documentation

Table Of Content

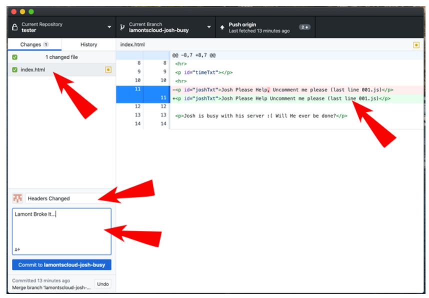
1) What New & Changing: 3 Steps…

Download From GitHub



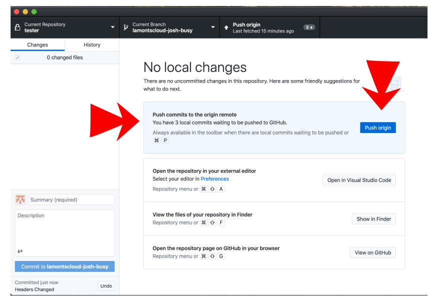
Creating A Branch & Download It

Edit Locally & Save



Make Your Changes, Save Your Files, Commit in Github App

Upload Change To GitHub



Push Changes To Github

2) Github Overview (Why)

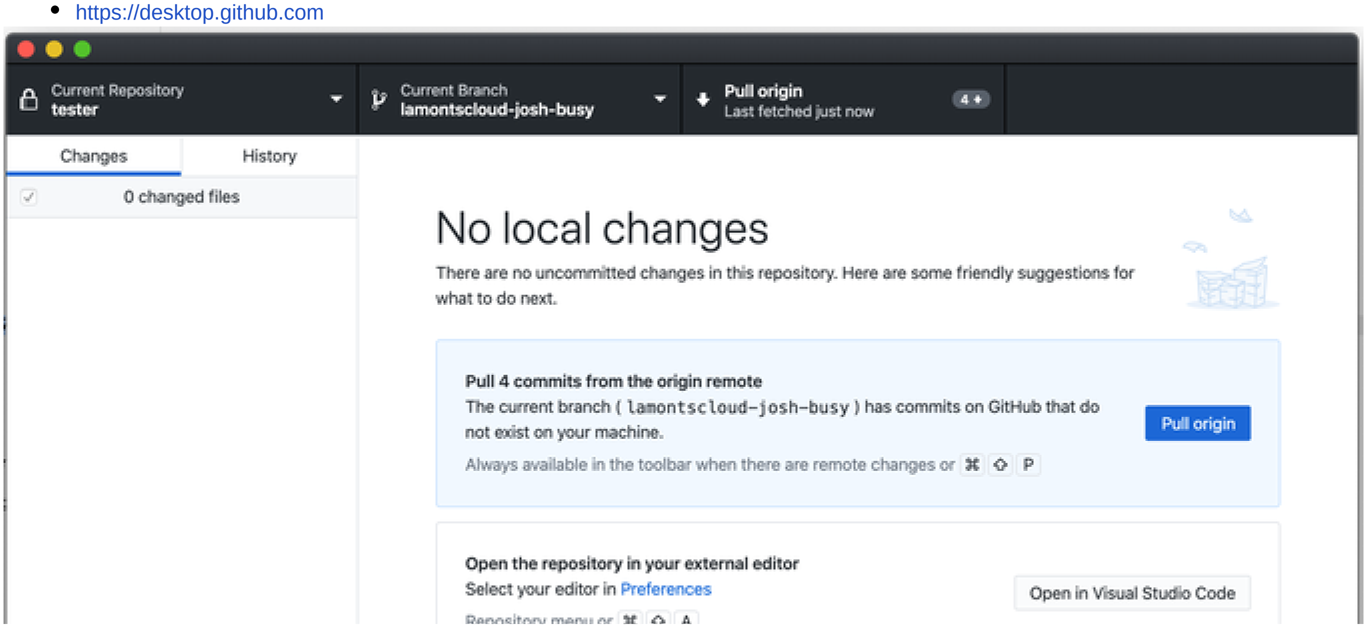
GitHub Is a Repository That Allows You To Collaborate and Track Changes in Your Code Across Versions. In short we will dramatically reduce the amount of code or code changes that we lose. No more having to redo work or fix things that break with no means of tracking down the break.

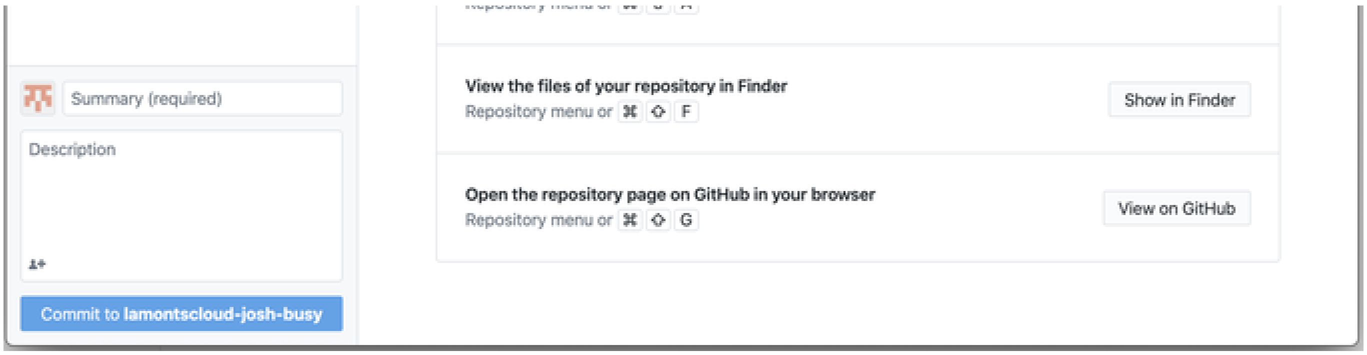
A **repository** is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs.

3) Software Needed

GitHub Desktop:

- https://desktop.github.com



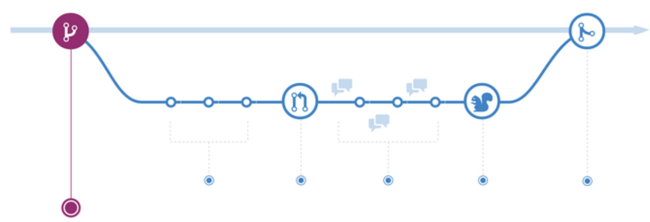


Select A Code Editor

- Brackets: http://brackets.io
- VS Code: https://code.visualstudio.com

4) The Github Flow

Your Branch (Master Should Always Remain Deployable To Live)



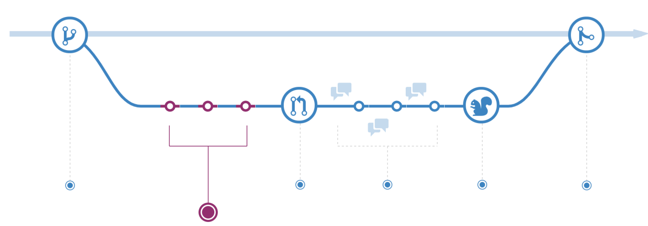
Branching is the way to work on different versions of a repository at one time. By default your repository has one branch named master which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to master.

ProTip

Branching is a core concept in Git, and the entire GitHub flow is based upon it. There's only one rule: anything in the master branch is always deployable.

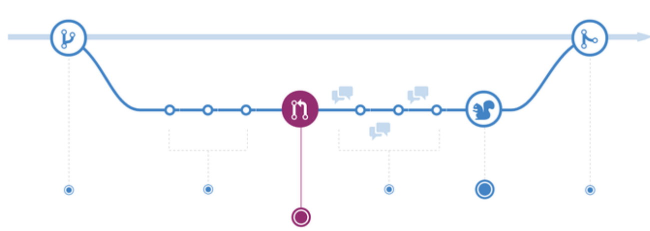
Because of this, it's extremely important that your new branch is created off of master when working on a feature or a fix. Your branch name should be descriptive (e.g., refactor-authentication, user-content-cache-key, make-retina-avatars), so that others can see what is being worked on.

Make & Commit Changes



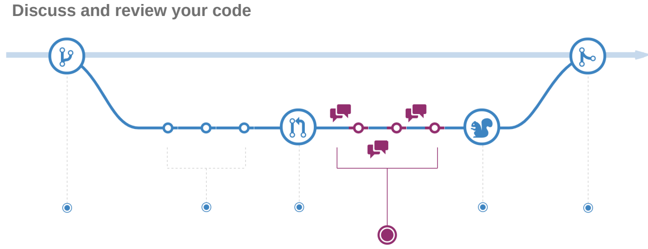
saved changes are called *commits*. Each commit has an associated *c ommit message*, which is a description explaining why a particular change was made. Commit messages capture the history of your changes, so other contributors can understand what you've done and why.

Pull Request



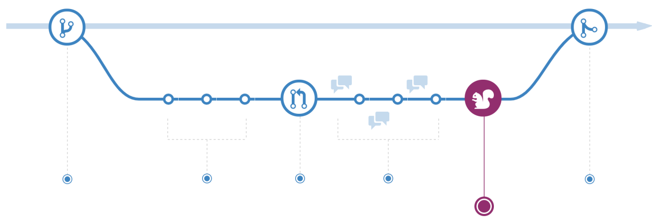
Pull Requests are the heart of collaboration on GitHub. When you open a *pull request*, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch.

Discuss and review your code



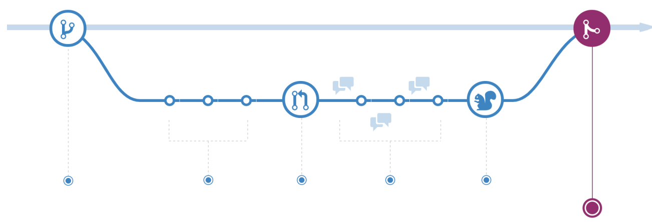
Once a Pull Request has been opened, the person or team reviewing your changes may have questions or comments. Perhaps the coding style doesn't match project guidelines, the change is missing unit tests, or maybe everything looks great and props are in order. Pull Requests are designed to encourage and capture this type of conversation.

Deploy Your Code



With GitHub, you can deploy from a branch for final testing in production before merging to master.

Merge With Master (Bring all the changes together).



ow that your changes have been verified in production, it is time to merge your code into the master branch.

Once merged, Pull Requests preserve a record of the historical changes to your code. Because they're searchable, they let anyone go back in time to understand why and how a decision was made.

Source: https://guides.githubv.com/introduction/flow/

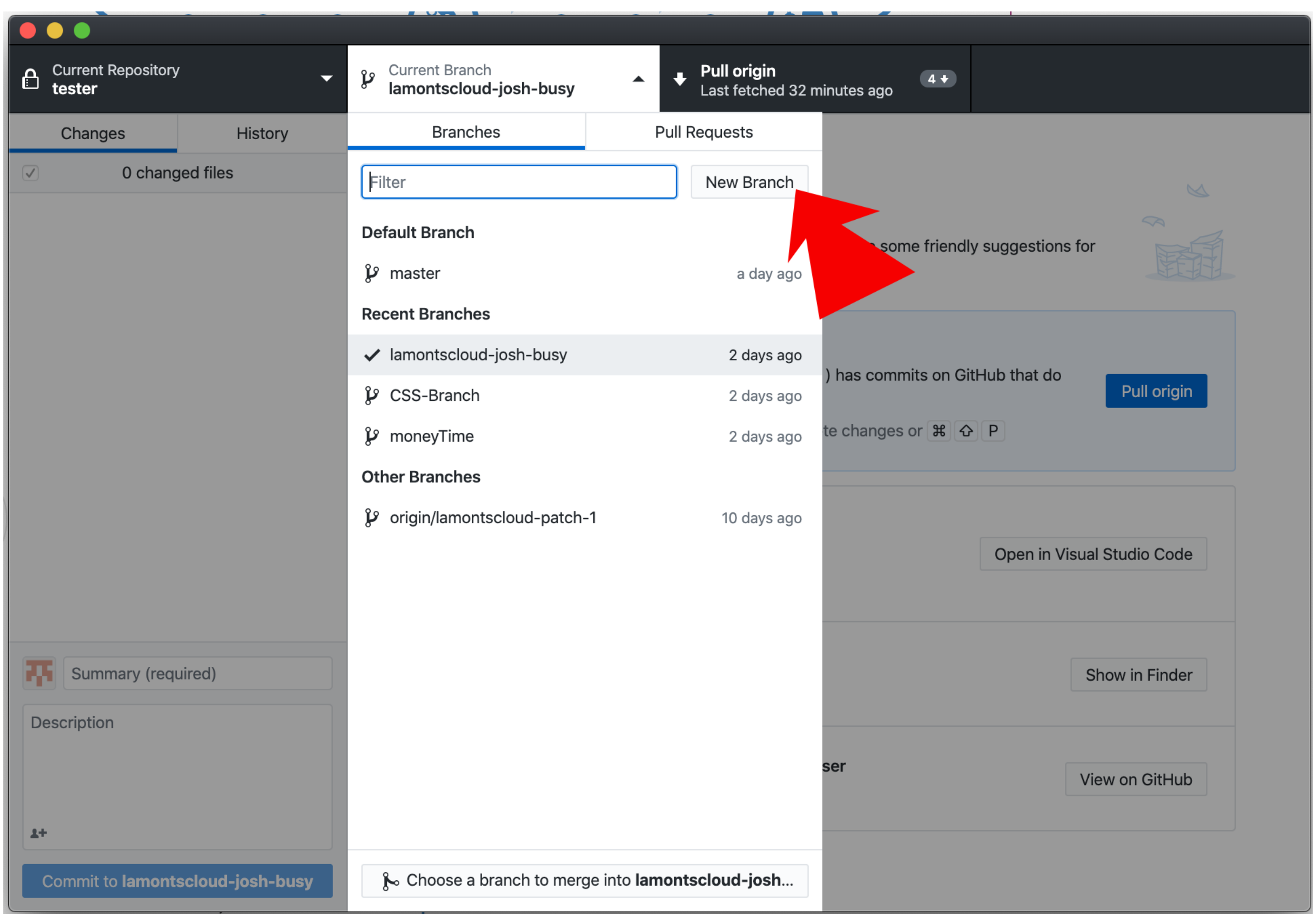
5) Our Development Process

a) Create A Branch & Download It To Your Local Machine

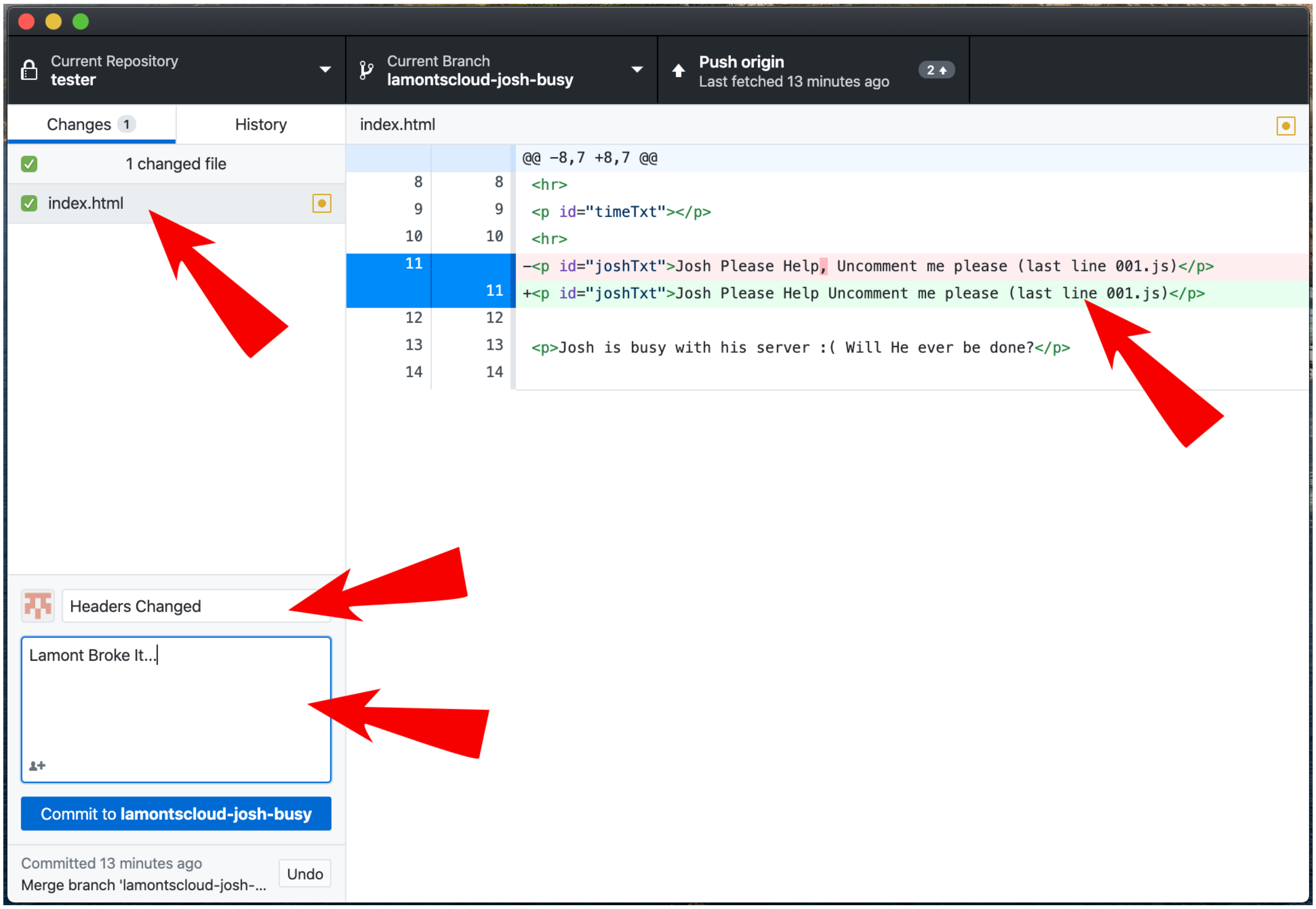
b) Make Changes, Edits & Revisions Locally & Commit To Github

c) Push Changes To Github

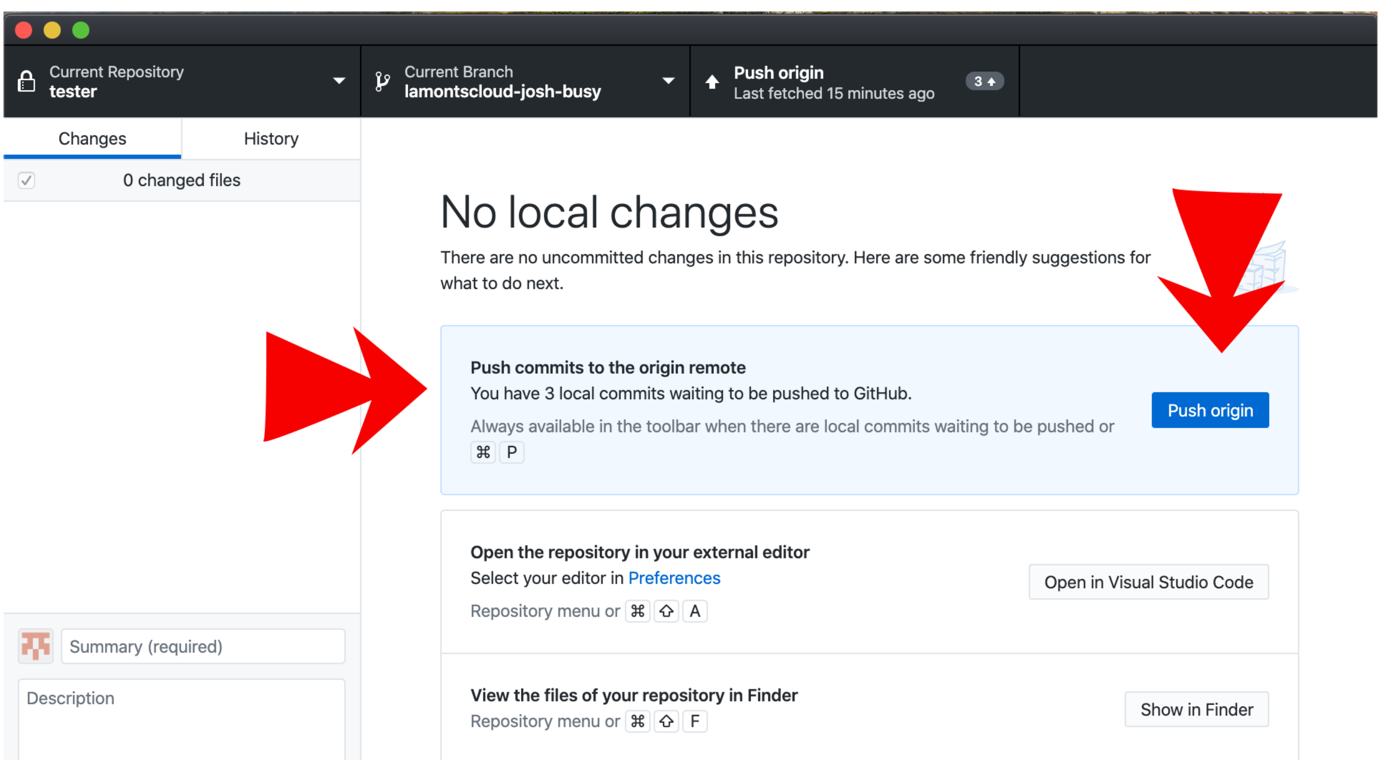
Creating A Branch & Download It

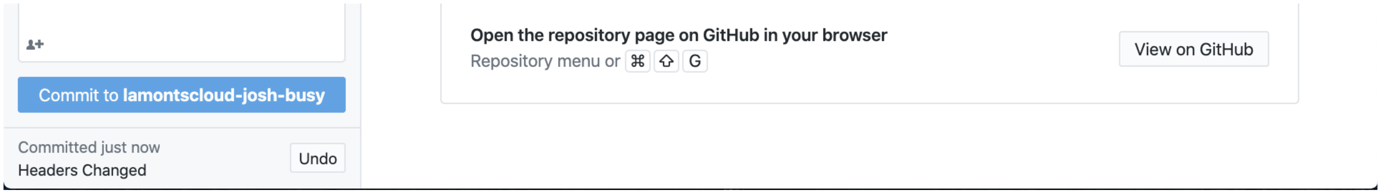


Make Your Changes, Save Your Files, Commit in Github App

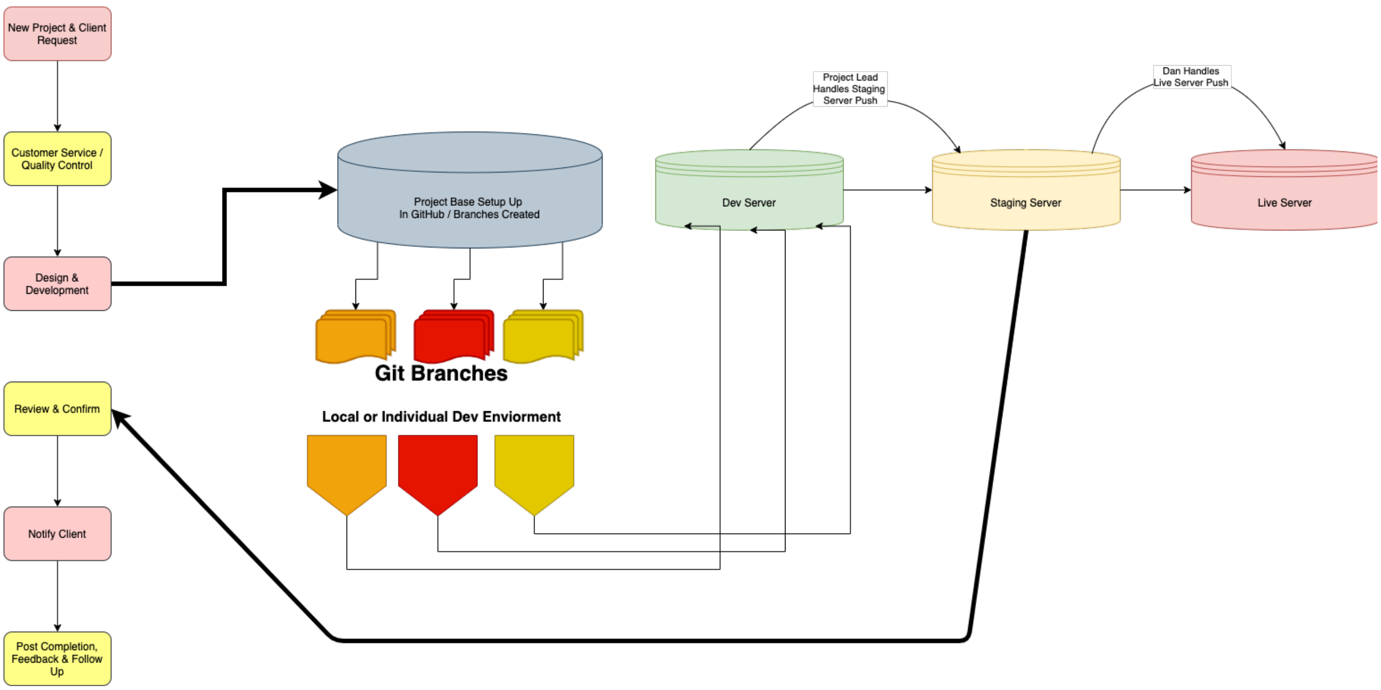


Push Changes To Github





6) DANG Workflow



(more to come on this)
Changes from our current process that will need to but put in place

1. Local Development Server
2. Staging Server
3. Project Manager For Relative Project

How to Use .gitignore

This document will serve as guidance for how to use .gitignore and how to make updates to the file to ensure that specific information is not sent over to the GitHub repository. For any actual step-by-step directions on how to perform a specific process, please refer to the Navigation Guides directory. Additional links to such files will be listed within this documentation directory to ensure that users can find it as well.

What Is It?

The .gitignore file is a text file that tells Git which files or folders to not push to the repository. In other words, this file is where all items that you don't want to commit to a repository should be located (i.e. logs, configuration files, password files, etc.)

How Does It Work?

A .gitignore file is a plain text file where each line contains a pattern for file/directories to ignore generally stored in the root folder of the repository (to ensure it can access all documents). Inside of this file, all rows started with a ('#') will be read as comments; everything else will be read as either a directory or file.

All patterns in the files are relative to the location of the .gitignore file.

How To Set Up A .gitignore File

Setting Up .gitignore File For Repository

Add File

Files can be added by simply typing the filename into the .gitignore file under the comment section. **NOTE:** If the file is under a specified directory, make sure to use the directory to reach the file prior to the file itself.

```
# Comment line here [STANDARD FILE EXAMPLE]
filename.txt

# Comment line here [FILE INSIDE OF DIRECTORY]
folder/filename.txt
```

Add Directory

Entire directories can be added to be ignored as well by simply placing a '/' (forward slash) after the directory name. NOTE: If a file is required to push to the repository inside of the directory you ignore, that file will not be uploaded.

```
# Comment line here [STANDARD DIRECTORY EXAMPLE]
logs/
```

Special Character Actions			
Character Name	**Symbol**	**Example**	**Notes**
Wildcard	*	*.log	The wildcard matches 0 or more characters (except /). In the example used, all files that end with .log will be ignored.
Exclamation Point (Exception)	!	!file.log	Using the example above, all log files would be ignored except for file.log. **NOTE:** If a full directory is ignored, the file inside of the directory cannot be excluded.
Double Wildcard	**	**/logs OR **/logs/*.log OR logs/**	Double wildcards can be used to match any number of directories. The examples listed in order do the following: • Matches all files or directories named logs • Matches all files ending with .log in a log directory • Matches all files inside of the logs directory

** Additional documentation can be found here at the link below **

https://git-scm.com/docs/gitignore

References
https://www.pluralsight.com/guides/how-to-use-gitignore-file
https://git-scm.com/docs/gitignore

Version Control Process

This page will serve to provide clarity into the version control methods used for branches on each website. Version control is essential to understanding the changes that a project has undergone to get to its respective finished product.

Semantics

1. A normal version number must take the form of x.y.z where x, y, and z are non-negative integers that DO NOT contain leading zeros.
    a. **X** is the major version
        i. **MAJOR** version when you make incompatible API changes
    b. **Y** is the minor version
        i. **MINOR** version when you add functionality in a backwards-compatible manner
    c. **Z** is the patch version
        i. **PATCH** version when you make backwards-compatible bug fixes
2. Software in production should be listed as **1.0.0** by default
3. Any changes made with a new update should be communicated regularly and transparently. Specifically users should be aware of the following:
    a. **What's new?**
    b. **What's been fixed?**

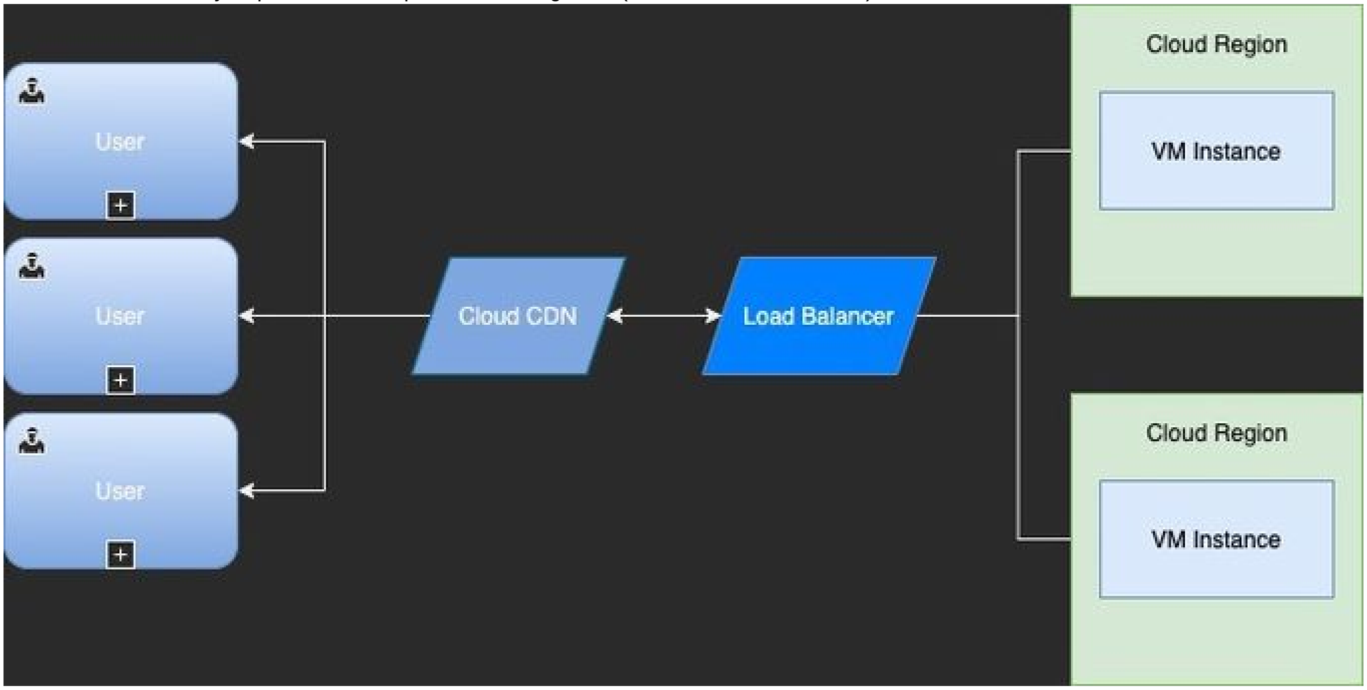
c. **What’s been improved?**
d. **What’s been deprecated?**

Examples			
Previous Version	**Update**	**New Version**	**Comments**
Version 1.0.1	A string of text was updated on the website from ‘taht’ to ‘that’	**Version 1.0.2**	**Reasoning:** No major changes were made that affect any compatibility with existing processes/products.
Version 1.0.1	A new plugin was added called ‘simplehistory’	**Version 1.1.0** OR **Version 2.0.0**	**Reasoning:** Depending on the plugin, there could be compatibility issues that cause other plugins to not work; however, in the instance where that isn’t the case, the version update would be to the former rather than the latter (1.1.1 instead of 2.0.0).
Version 1.0.1	A database was updated to include additional columns, rows, fields, etc.	**Version 2.0.0**	**Reasoning:** The change made will not be communicated within the GitHub repository and should be updated on the live database before committing the changes to the master.
Version 1.0.1	A new script was added to the website to enhance a plugin	**Version 1.1.0**	**Reasoning:** This would be considered a minor change but worthy of being noted as more than a simple update.

References
https://rollout.io/blog/best-practices-when-versioning-a-release/#:~:text=Given%20a%20version%20number%20MAJOR,make%20backwards%2Dcompatible%20bug%20fixes.
https://semver.org/

Edge-Caching - Cost & Implementation Analysis

This page will break down the options for edge-caching and what each price point will be. Further, the implementation will be broken down as well since each site may require a different process for integration (WordPress, Joomla!, etc.).



Edge-Caching Explanation

CDNs work by reducing the distance between your website and your target audience by caching content locally to the client. When the client attempts to reach the content, instead of pulling from the web server directly, they pull from a server near them for a faster access point.

As an example, imagine if you were throwing a ball from across a football field every time you wanted to play catch with a friend (*ignore the feasibility*). The ball would have to travel a large distance every single time it was being returned. However, if I have a friend (*a nearby server in this metaphor*) who is closer to who I am throwing the ball to, I can throw the ball to them and have them throw the ball back and forth with my friend for me.

Plugins for CDN

To properly cache items for each website using any vendor, the respective plugin will need to be included along with the service to ensure that the data is being routed correctly. If this is not done properly, there is no added benefit to edge-caching.

For WordPress sites, the implementation is done through the W3-Total-Cache plugin (link in the references).

- MaxCDN (StackPath) **CONFIRMED**
- Google Edge-Caching **CONFIRMED**
- **In order to use W3-Total-Cache, we will have to remove the WP-Rocket**
- Free version supports CDN for MaxCDN (StackPath)

**WordPress also has a plugin for through WP Rocket for CDN services. However, from WP-Rocket's site, "The subscription is unique to your domain - so you cannot use the same RocketCDN subscription on multiple domains."*

**"W3 Total Cache Pro licenses are valid on a per-install basis, meaning you need one for every WordPress instance you run. If you have three separate websites, you will need three licenses in total—one for each site. If you have five sites in a WordPress multisite instance, you need just one license for all five sites."*

For Joomla sites, the implementation is done through the use of Pro Version of CDN For Joomla! plugin (link in the references). *The free version of the CDN For Joomla plugin does not support HTTPS. If you need HTTPS support, you need to buy the paid version.*

- MaxCDN (StackPath) **CONFIRMED**
- Amazon CloudFront **CONFIRMED**
- Google Edge-Caching **NEGATIVE**
- Free version and Pro version support multiple website usage

For Magento sites, the implementation is done through the use of domains through admin panel and the MaxCDN website.

- MaxCDN (StackPath) **CONFIRMED**
- Google Edge-Caching **CONFIRMED**
- Both of these will work since Magento requires the user to update the fields via domains submitted through the admin panel and the main location where those would be set.

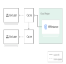
CDN For Joomla! Plugin (Pro Version)	
Features	**Comments**
Handle HTTPS urls	Option to also convert SSL urls (https) and optionally keep the https in the cdn urls
Multiple CDN roots	You have the ability to set up 5 different CDN urls, each with their of settings, like file types, https handling, etc.
File versioning	Enable versioning of the file URLs to force the CDN to pull a new version of the file if it gets updated.
Pro downloads & updates	Get access to the Pro downloads and updates for a full year.
Install on multiple domains	**You can install the extension on any domain you want. No limitations. No domain checks**
30 Day money back guarantee	Not happy with the extension? Or simply have no use for it after trying out the Pro version? Simply ask your money back!
Open Source code	The code is open source (GPL), meaning there is no form of code encryption. No use of Zend Guard or ionCube.

W3-Total-Cache plugin	
Benefits	**Features**

Improvements in search engine result page rankings, especially for mobile-friendly websites and sites that use SSL	Compatible with shared hosting, virtual private / dedicated servers and dedicated servers / clusters
Improved conversion rates and “site performance” which affect your site’s rank on Google.com	Transparent content delivery network (CDN) management with Media Library, theme files and WordPress itself
“Instant” repeat page views: browser caching	**Secure Socket Layer (SSL) support**
Reduced page load time: increased visitor time on site; visitors view more pages	Defer offscreen images using Lazy Load to improve the user experience
Up to 80% bandwidth savings when you minify HTML, minify CSS and minify JS files.	Extension framework for customization or extensibility for Cloudflare, WPML and much more

---

Costs



Data transfer from a cache to a client is called cache egress. Data transfer to a cache is called cache fill. As illustrated in the following figure, cache fill can originate from another Cloud CDN cache or from the origin server. On cache hits, you incur costs for cache egress bandwidth. On cache misses—including misses that resulted in cache-to-cache fill—you also incur costs for cache fill bandwidth.

Google Cloud Edge Caching Estimates		
Based on data from: 7/4/2020 - 8/4/2020		
Incoming	58.44 GB	$2.34
Outgoing	2.756 TB	$225.77
Plugin Support	N/A	$3.85
VM Instance	N/A	$295.06
Total:		$526.99

- backups may account for size of outgoing.
- Local testing by pulling all files on website down may also play a significant factor.

Google Cloud Edge Caching Estimates		
Based on 100GB ingress and egress		
Incoming	100.00 GB	$4.00
Outgoing	100.00 GB	$8.00
Plugin Support	N/A	$3.85
VM Instance	N/A	$295.06
Total:		$310.91

MaxCDN (StackPath) provides users with one TB per month bandwidth; all extra bandwidth is charged .04 per GB after the one TB has been reached. Unlike GCP, however, MaxCDN has support for WordPress AND Joomla! plugins while also being supported by Magento Plugins as well.

MaxCDN (StackPath) Edge Caching Estimates		
Based on 1.360 TB Bandwidth Reported by WHM		
Incoming	1.360 TB	$14.40
Outgoing		
StackPath	N/A	$20.00
Plugin Support (Joomla ONLY)	N/A	$3.85
VM Instance	N/A	$295.06

Total:		$333.31

Amazon CloudFront Edge Caching Estimates		
Based on 1.360 TB Bandwidth Reported by WHM		
Incoming	1.360 TB	$115.60
Outgoing		
Plugin Support (Joomla ONLY)	N/A	$3.85
VM Instance	N/A	$295.06
Total:		$414.51

Technical Needs

Google Cloud Specifics

Since we will be using an instance that has the WHM information stored, the instance type required will be an unmanaged instance group. The unmanaged instance group must be created to be placed into the load balancer; this allows the CDN to be utilized from the load balancer settings. Additionally, this means that there must be an external static IP set up for the new load balancer, and each instance to be placed into the unmanaged instance group.

Managed or Unmanaged Instance Groups		
Support_Type	**Managed Instance Group**	**Unmanaged Instance Group**
Load Balancing	POSITIVE	POSITIVE
Autoscaling	POSITIVE	NEGATIVE
Autohealing	POSITIVE	NEGATIVE
Rolling Update Support	POSITIVE	NEGATIVE
Multi-zone Support	POSITIVE	NEGATIVE

The current instance is bringing attention to its configuration being overutilized and, thus, would need to be increased for the new instance to match the requirements suggested (10 CPUs, 51.5 GB). NOTE: 51.5 is not an accurate number that can be used for a custom instance; it must be an integer. Therefore, 52 would work.

Alternative is using a Google Cloud Bucket to reroute the traffic coming in towards the bucket. This process removes the necessary steps of changing the instance altogether.

Purging

Caching all files means that the content will need to updated on the remove server when updates occur. This is done through purging the content from the CDN source, and having the CDN reattempt to pull the content from the origin. The breadth of this requirement is different depending on the CMS type.

Alternative Example: Imagine you have a card in your hand, and your hand can only hold one card. Every time I look at your hand, you hold the same card. However, changes have occurred and the card needs to be updated. Therefore, you must drop the card in order to hold the new one; from this awful analogy, it can be understood that the content must be purged (dropped) to have the new content grabbed.

1. **WordPress:** *W3-Cache Plugin* > Empty All Caches Button
2. **Joomla:** *CDN for Joomla!* > System > Clear Cache
3. **Magento:** *Admin Panel* > Purge Cache or Clear Cache per subdomain

---

Implementation

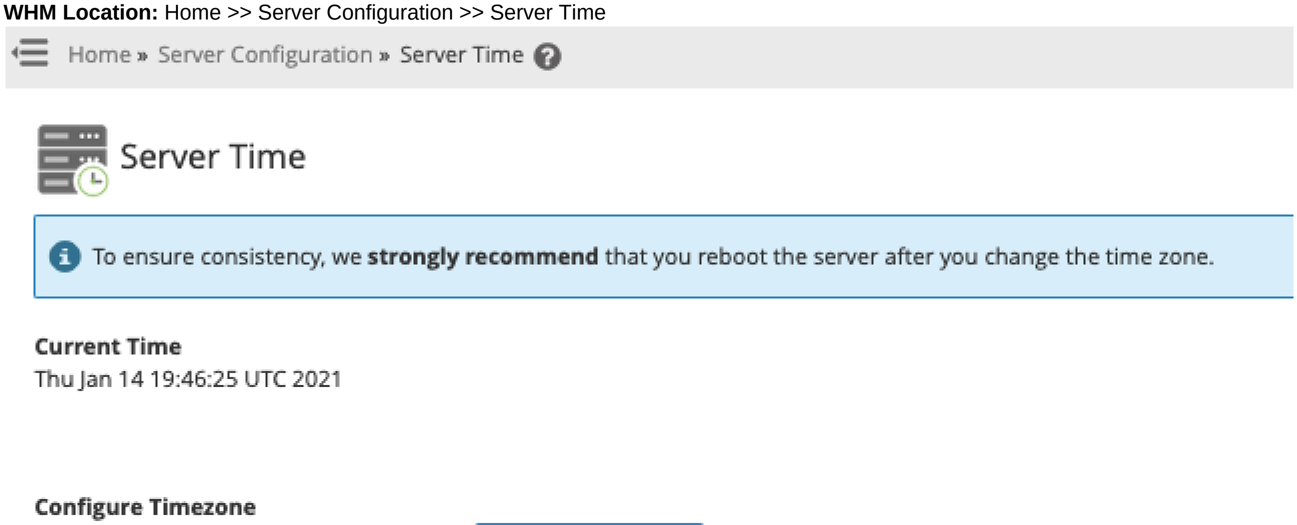
SlackPath Videos		
CMS	**Youtube Link**	**Comments**

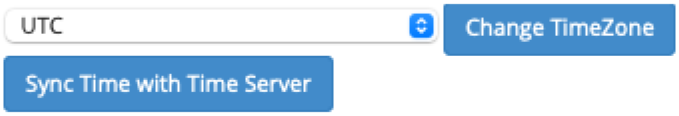
WordPress	https://www.youtube.com/watch?v=-i5a9gLiPLE	• Create site in StackPath • Authorize W3-Total-Cache by generating ID from StackPath • test linkage • Enable ‘Cache Pages’, ‘Minify’, ‘Object Caching’, CDN, browser caching, and update CDN to use canonical headers
Joomla	https://www.youtube.com/watch?v=FVcXWkrJ04E	• Create site in StackPath • Upload extension file • Go to Basic Settings and enter CDN domain • Select all applicable checkboxes and save
Magento	https://www.youtube.com/watch?v=E9fAd2LFd_Q&t=223s	• Create site in StackPath • Add zone host for CDN • Update web configuration in Magento Admin panel

References
https://cloud.google.com/cdn/?utm_source=google&utm_medium=cpc&utm_campaign=na-US-all-en-dr-bkws-all-all-trial-b-dr-1009135&utm_content=text-ad-none-any-DEV_c-CRE_227055718745-ADGP_Hybrid+%7C+AW+SEM+%7C+SKWS+%7C+US+%7C+en+%7C+Multi+~+Cdn-KWID_43700026980903140-kwd-299230872223&utm_term=KW_%2Bcdn-ST_%2Bcdn&gclid=EAIaIQobChMIoMP9ueP_6gIVCZ2zCh00mgrrEAAYASAAEgLoUPD_BwE
https://cloud.google.com/blog/products/networking/google-cloud-networking-in-depth-cloud-cdn
https://www.codeinwp.com/blog/maxcdn-vs-cloudflare-vs-cloudfront-vs-akamai-edge-vs-fastly/
https://docs.wp-rocket.me/article/1307-rocketcdn
https://www.regularlabs.com/extensions/cdnforjoomla
https://support.stackpath.com/hc/en-us/articles/360001086403-Joomla-StackPath-CDN-Setup-Guide
https://docs.cpanel.net/knowledge-base/web-services/how-to-install-cpanel-on-google-compute-engine-through-the-google-marketplace/
https://cloud.google.com/compute/docs/instance-groups
https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-unmanaged-instances
https://cloud.google.com/products/calculator#id=
https://www.youtube.com/watch?v=PH_ZABYaYqs
https://vlemon.com/blog/google-cloud-platform/compute-engine-create-centos-7-vm-instance-and-install-c-panel/
https://deliciousbrains.com/wp-offload-media/doc/how-to-set-up-a-custom-domain-cdn-for-google-cloud-storage/
https://wordpress.org/plugins/w3-total-cache/
https://www.youtube.com/watch?v=akX-soLyjw4

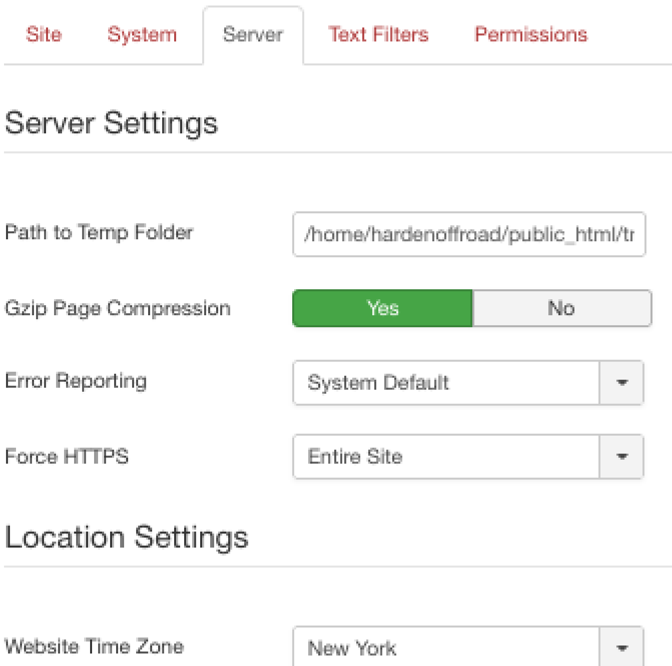
Event & Website Server Time Zone Locations

1. **WHM Location:** Home >> Server Configuration >> Server Time

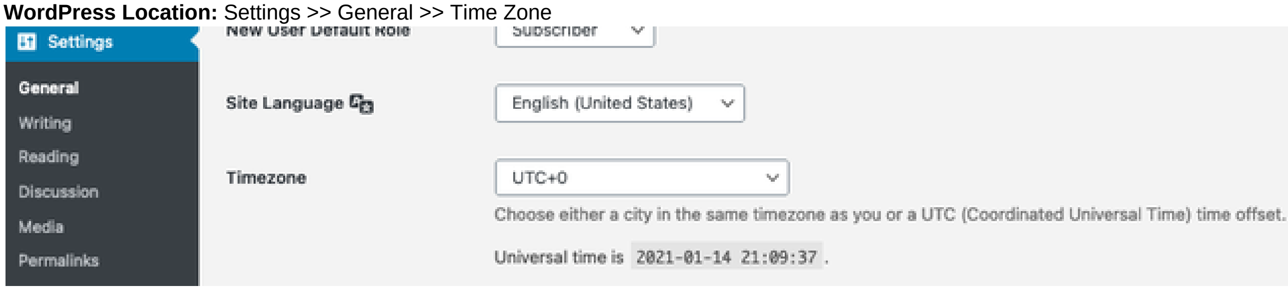




2. **Joomla Location:** Global Configuration >> Server Tab >> Location Settings Section



3. **WordPress Location:** Settings >> General >> Time Zone



NOTE: You may also need to check Event, Membership, or eShop plugins as well.

Migrating Fishbowl Server to Windows Server

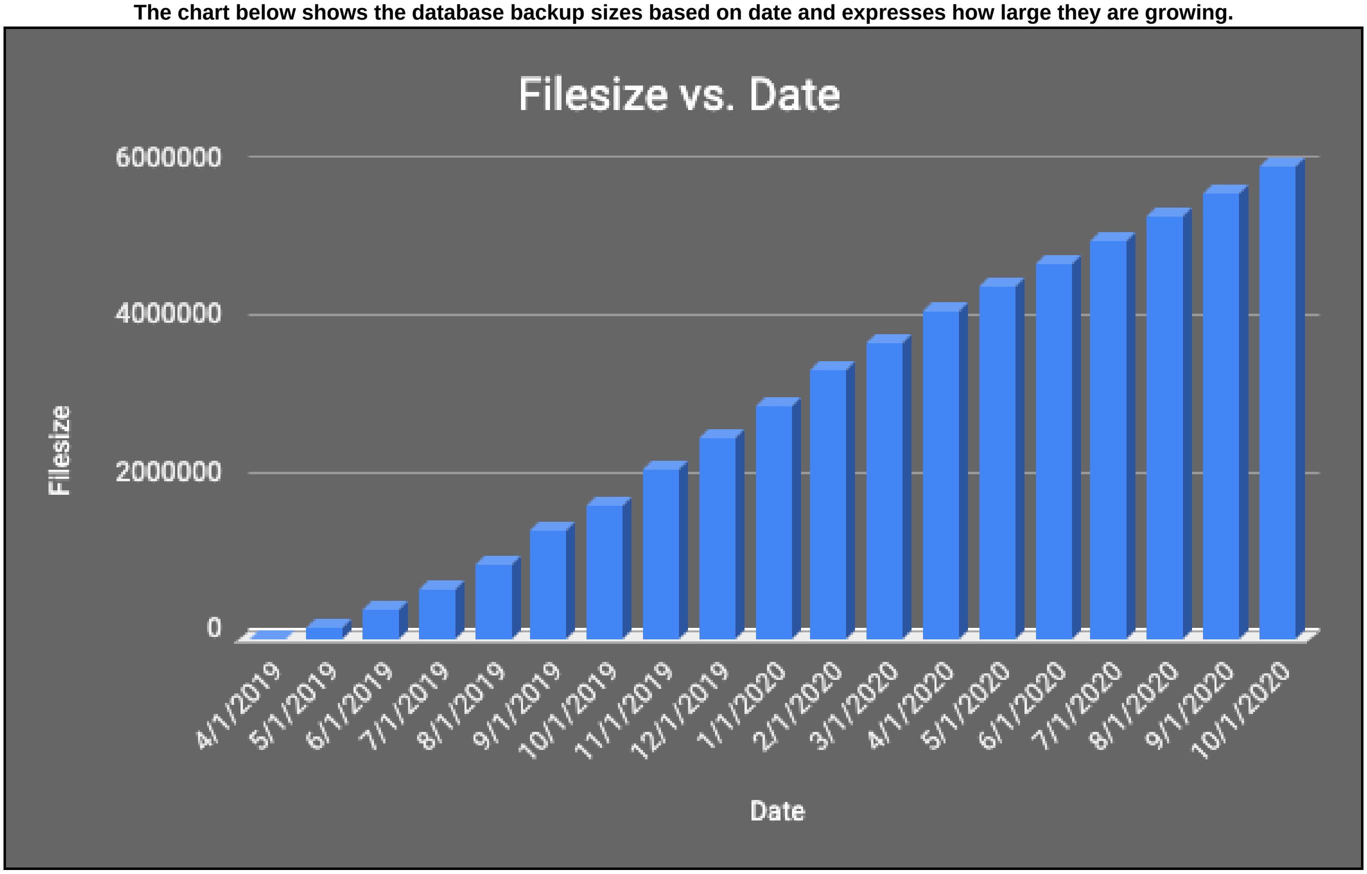
This page will cover the steps and information required to make the migration from our current Fishbowl virtual machine to the new Windows Server Datacenter virtual machine.

First we will cover all the requirements that are needed to be in place prior to the transition.
Next, we will discuss the security details required to make on the VM to ensure that it meets the same standards held for the current Fishbowl (and what else should be set in place).
Finally, we will detail the additional precautions that can be taken to ensure success.

Requirements		
Need	**Reason**	**Comments**
Fishbowl Server	**The server is required to allow connections, whether local or remote.**	This has already been set up and configured on the current Windows server. *(see security requirements for additional information about ports and firewall settings)*

Fishbowl Client	**The client is required on the machine to have a GUI available for users to continue their processes as normal.**	This has already been set up and configured on the Windows server.
Jar Files	**There are specific jar files needed for plugins on the Fishbowl client to run. Inside of the Fishbowl files, these jars are placed in the plugins folder. Also worth noting, the plugin for advanced shipstation must be downloaded within the client in order for the plugin to function accurately *(after placed in plugin folder)*** • advanced-shipstation-19.11.6.jar • fishbowl-go-18.12.01.jar • magento-cart-19.11.3.jar	These files have already been included in the Windows server installation but are also located in **01-DANG-Work > 00-Stock-&-Software-related > JAR Files** in the Google Drive directory.
Adequate Space	**The backup files are set to be stored within the Fishbowl > Data > Backups and are getting larger with each passing day. Since VM instances have a larger associated cost with storage internally, it would be wise to either push the backups to a Google Cloud bucket (see** SunFlora - Setting Up Cloud Storage Bucket From PHP Script **) or download/delete these backups to an alternative location daily.**	The script required to make this push is an automated solution that would need to be set up to run daily to make that push. However, at the rate of file size increase *(refer to column chart below)* this is not the most optimal solution. We should sincerely consider removing tables or data from the database that isn't necessary since its exponential increase could overwhelm our storage solutions long term.

The chart below shows the database backup sizes based on date and expresses how large they are growing.



Security Requirements

Firewall Rules Required

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Logs
block-outgoing	Egress	Apply to all	IP ranges: 222.186.175.217, 2	all	Deny	1000	Off

whm-and-outgoing-email	Egress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22;tcp:465;tcp:587;tcp:25;tcp:26;tcp:2525	Allow	1000	Off
blocked-ips-vm-sunflora	Ingress	Apply to all	IP ranges: 172.105.89.161, 10	all	Deny	0	Off
vnc-port	Ingress	Apply to all	IP ranges: 47.205.3.38	tcp:5901	Allow	1	On
dang-office	Ingress	Apply to all	IP ranges: 47.205.3.38	all	Allow	10	Off
ip-blocks-mod-sec	Ingress	Apply to all	IP ranges: 54.36.148.0/24, 54	all	Deny	1000	Off
astra-sec-ip	Ingress	Apply to all	IP ranges: 18.209.148.132, 12	all	Allow	1000	Off
default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	Off
default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	Off
mysql-fishbowl	Ingress	Apply to all	IP ranges: 47.205.9.144	tcp:3306	Allow	1000	Off
ssh-dang-office	Ingress	Apply to all	IP ranges: 47.205.13.182	tcp:22	Allow	1000	Off
vnc-dang-ip-whitelist	Ingress	Apply to all	IP ranges: 47.205.3.38	all	Allow	1000	On
whm-cpanel-rules	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:21;tcp:80;tcp:443;tcp:2082;tcp:2083;tcp:2089;tcp:2095;tcp:2096;tcp:2087;tcp:2086	Allow	1000	Off
zapier	Ingress	Apply to all	IP ranges: 54.86.9.50	all	Allow	1000	Off
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	Off
default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	Off
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	Off
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	Off

Steps For Transition

1. **Ensure the backup on the new Windows Server is up-to-date with the most recent backup**
    a. Copy a zipped file through the Google Drive and download to the Windows Server
    b. Stop the server (if active)
    c. Go to File > Options
    d. A new panel will open, go to the Restore tab
    e. Select the new database SQL file downloaded
    f. Ensure the port is set to 3306 and update the password to ‘password’
    g. Press ‘Restore’
        i. *The file may take up to an hour to fully restore due to the sheer size of it
2. **Once the restore has completed, mark all tasks as inactive to ensure they don’t have the wrong file path and run incorrectly**
    a. Open up the Fishbowl Client and sign in
    b. Navigate to Setup > Schedule
    c. If tasks have started to queue, cancel them by pressing the 'X' next to the running task
    d. Do this for each and every task in the list:
        i. Select a task in the list and select the icon on the right-hand side that looks like a pencil
        ii. De-select the checkbox that says ‘Active’
        iii. Press ‘Next’, ‘Next’, ‘Finish’
    e. In order to see these tasks for the next step, check the checkbox on the Schedule view called ‘Show Inactive’
3. **Correct the file paths for all tasks to point to the proper location for when they will run**
    a. The file path for the following plugins will need to be reviewed to ensure they are still accurate and have the same destination:
        i. Magento Cart 1 OrderImport Split Payments
            1. Should point to C:\Data\Payment2Import
        ii. Database Back Up
            1. Should point to C:\Program Files\Fishbowl\data\backups

4. **Ensure that the connections are correct on the new machine for Magento Cart**
    a. Select the plugin from Integrations
    b. Press the 'Test Connection' button and ensure that the message says 'Connection Successful'
5. **Correct all custom processes are pointing to the new location, as opposed to the previous one**
    a. ? (These processes need clarity as Ron has them set up mostly through API connection points)
    b. IP address references will need to point to this machine or the load balancer to send the data correctly over.

References	
Link	**Comments**

New Hire Onboarding Software Introduction

DANG Specific Docs

- Web Development Checklist - https://dangdesigns.atlassian.net/l/c/3oTWsKTr
- Client On-Boarding Process - https://dangdesigns.atlassian.net/l/c/0p9137KU
- Joomla and WordPress Testing - https://dangdesigns.atlassian.net/l/c/M5fEHLr1
- Custom Website Code Pieces - https://dangdesigns.atlassian.net/l/c/Y0JcSz8R
- GitHub Dang Workflow Documentation - https://dangdesigns.atlassian.net/l/c/iQryWSHs
- Dev Team Processes - https://dangdesigns.atlassian.net/l/c/eP4ZNB20

Jira (Project and issue tracking)

- Official Website - https://www.atlassian.com/software/jira
- Video Tutorial - How to use Jira for Beginners

Confluence (Document collaboration)

- Official Website - https://www.atlassian.com/software/confluence
- Video Tutorial - https://youtu.be/IZJmdvoXeQs

Slack (Team conversations and channels)

- Official Website - https://slack.com/
- Video Tutorial - https://youtu.be/EYqxQGmQkVw

Google Workspace (Email, Docs, Sheets, Drive, Meet, Keep)

- Official Website - https://workspace.google.com
- Video Tutorial - https://youtu.be/bE31y5HbukA

Joomla (Content Management System - Website Builder)

- Official Website - https://www.joomla.org/
- Admin User Setup - https://dangdesigns.atlassian.net/l/c/6knCa6Js

WordPress (Content Management System - Website Builder)

- Official Website - https://wordpress.org/
- Admin User Setup - https://dangdesigns.atlassian.net/l/c/j4caK1hZ

Yootheme (Page Builder)

- Official Website - https://yootheme.com
- Youtube Channel - https://www.youtube.com/channel/UCScfGdEgRCOh9YJdpGu82eQ

Slickplan (Website Sitemap Planning Tool)

- Official Website - https://slickplan.com/
- Youtube Channel - https://www.youtube.com/channel/UCIR2f27R8JaUEQ2SyhhU4MA

Magento 1 (Sunflora, Capital City Shoes, Shoebox)

- Official Website - https://magento.com/
- User Guide - https://drive.google.com/file/d/19g1VjiO2i28Aq3veeyskY0ukAOBgRTWW/view?usp=sharing

Magento Commerce 2

- Official Website - https://magento.com/products/magento-commerce/page-builder
- Setting Up Magento 2 - https://dangdesigns.atlassian.net/l/c/1NrR9qFJ

RICS Software - API Info Documentation

Due to the difficulty in finding the location for RICS software API documentation, this page will host the link for where these resources can be found to be used in Magento 2 integration.

https://enterprise.ricssoftware.com/Api/help

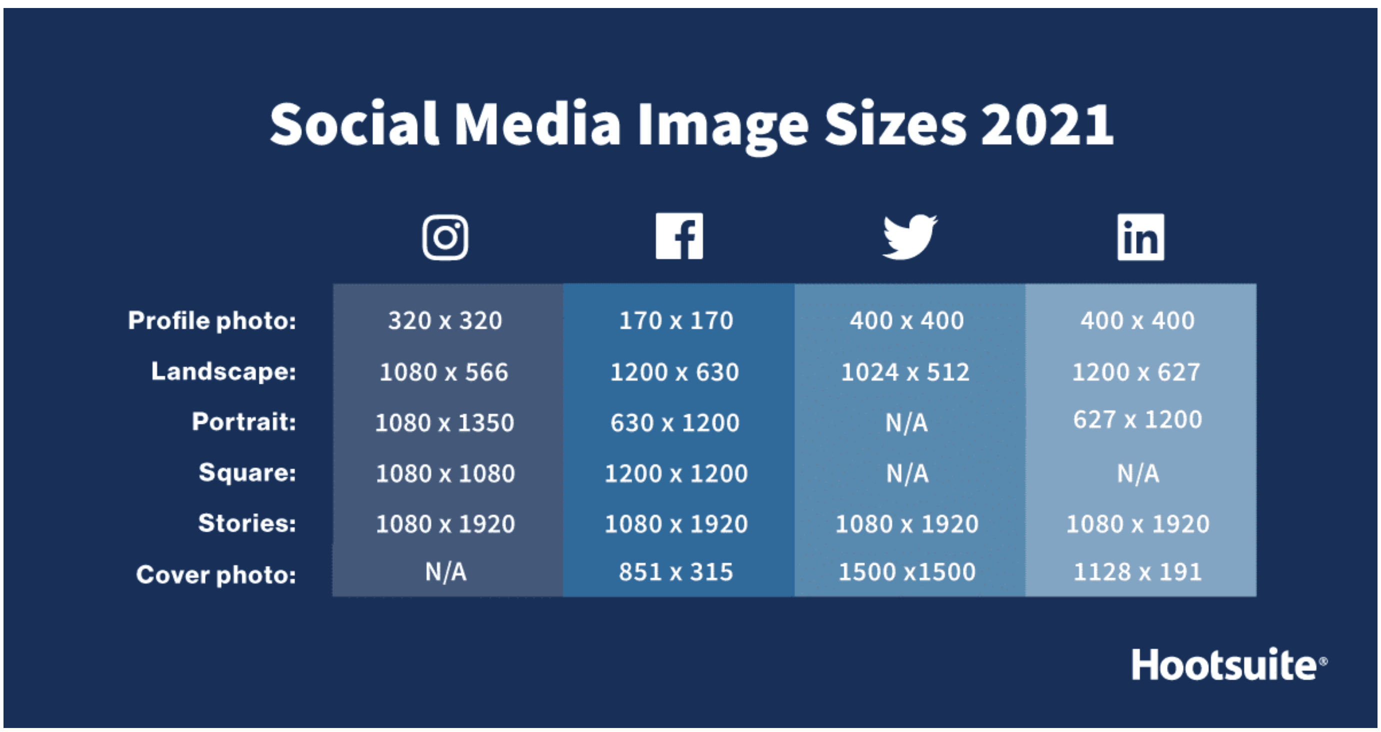
RICS Software (WordPress) VS. RT (Magento 2.0)

This page will cover the advantages and disadvantages between using RICS software (using Woocommerce on WordPress) and RT (using another plugin on Magento 2.0).

References	
Link	**Notes**
https://support.ricssoftware.com/hc/en-us/articles/360033409652-WooCommerce-Setup	Covers Woocommerce setup on WordPress for RICS software

Social Media Image Sizes

Cheat Sheet.


Social Media Image Sizes 2021

Profile photo:	320 x 320	170 x 170	400 x 400	400 x 400
Landscape:	1080 x 566	1200 x 630	1024 x 512	1200 x 627
Portrait:	1080 x 1350	630 x 1200	N/A	627 x 1200
Square:	1080 x 1080	1200 x 1200	N/A	N/A
Stories:	1080 x 1920	1080 x 1920	1080 x 1920	1080 x 1920
Cover photo:	N/A	851 x 315	1500 x1500	1128 x 191

Hootsuite®

Detailed size specifications for different social media networks, including Instagram, Twitter, Facebook, LinkedIn, and more.

https://blog.hootsuite.com/social-media-image-sizes-guide

Sunflora - Affiliate Groups & Credits

The page will provide an overview of the Affiliate Groups (found in Magento backend > Affiliate Pro > Manage Affiliate Groups) and commissions for those stores based on their respective ruleset. With each new rule that is needed, a new group must be created, and

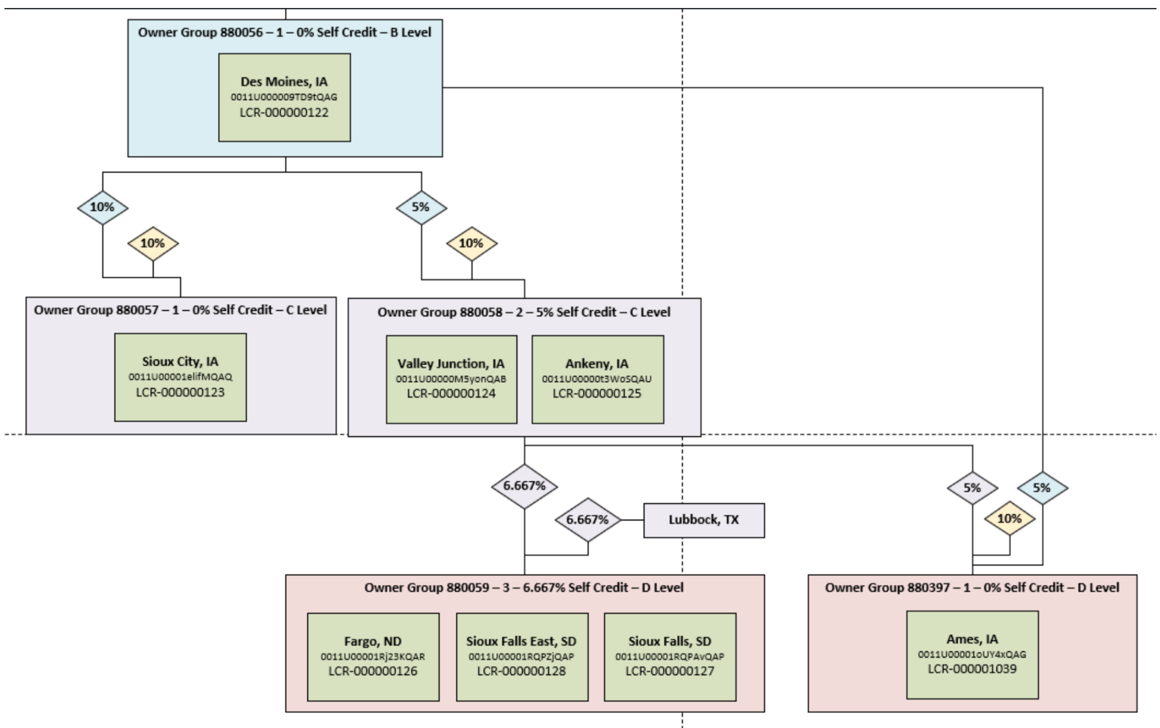
the group must be added to two separate files to function properly (commissions.php and commissions-manual.php found in the dashboard VM under the Utilities folder).

Number	Name	Definition	Visual
1	Corporate and Test Accounts		
2	Founding Store		
3	B Level		
4	C Level		
9	Founding Stores with Subs		
11	B level with Subs		
12	C Level with Subs		
13	Founding Stores with Subs Straight 20		
14	B Level Special 10 - Augusta / Cortese Stores		
15	D Level		
16	A Special Woodstock		
17	B- Split between Davenport and Augusta		
18	D split betwe		

	en special 5 and Nola		
19	Special 5		
20	Wholesale A		
22	D with Subs		
23	C - split between Davenport, Boynton, and Augusta		
24	CE		
25	CE-B1		
26	E Level		
27	D split between special 5 and Nola with Subs		
28	B level - mountainbrook special split northport and Birmingham		
29	E Level Special 5		
30	Singer Special 10		
31	CE-B with Subs		
32	CE-C1		
33	CE-C with subs		

34	Closed or Never Open		
35	Closed		
36	F Level		
37	00 Credit Hold		
38	Sold or Transferred to New Owner		
39	D Level special splt 6.7% each to Navin and Smyrson		
40	C level special split each to Acworth and Chattanooga get issue #220		
41	E Level with Subs		
42	Single Pay 10 to Loudonville (B) - Ithaca (B) from Guilderland 361 (C)		
43	Wholesale Split between naugatuck and bristol		
45	Converted		

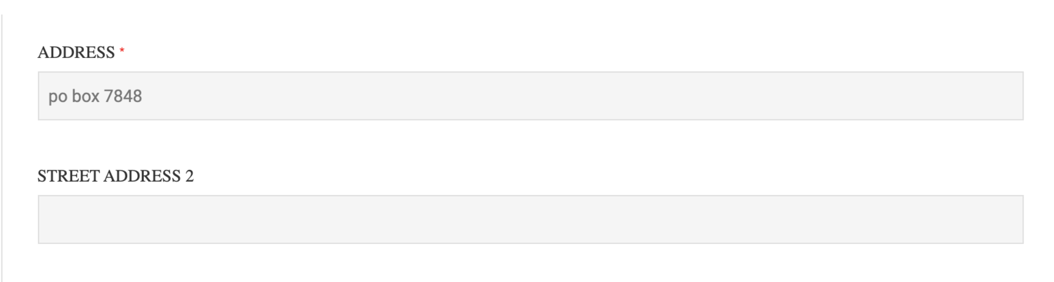
	to Franchise		
46	whole sale split between Palm Harbor and Oldsmar		
47	Referral Parent		
48	Referrer		
49	UK split 5% each to special 5 and Nola		
50	B - Special Split Between Thousand Oaks and Aiea		
51	D - With Subs Special 5 Fairfax		
52	D Level split with Tempe and Gilbert		
53	Temp Closed		
54	E Level Special 5 With Subs		
55	Temporarily Closed		
56	C Level Special Split between Thousand		

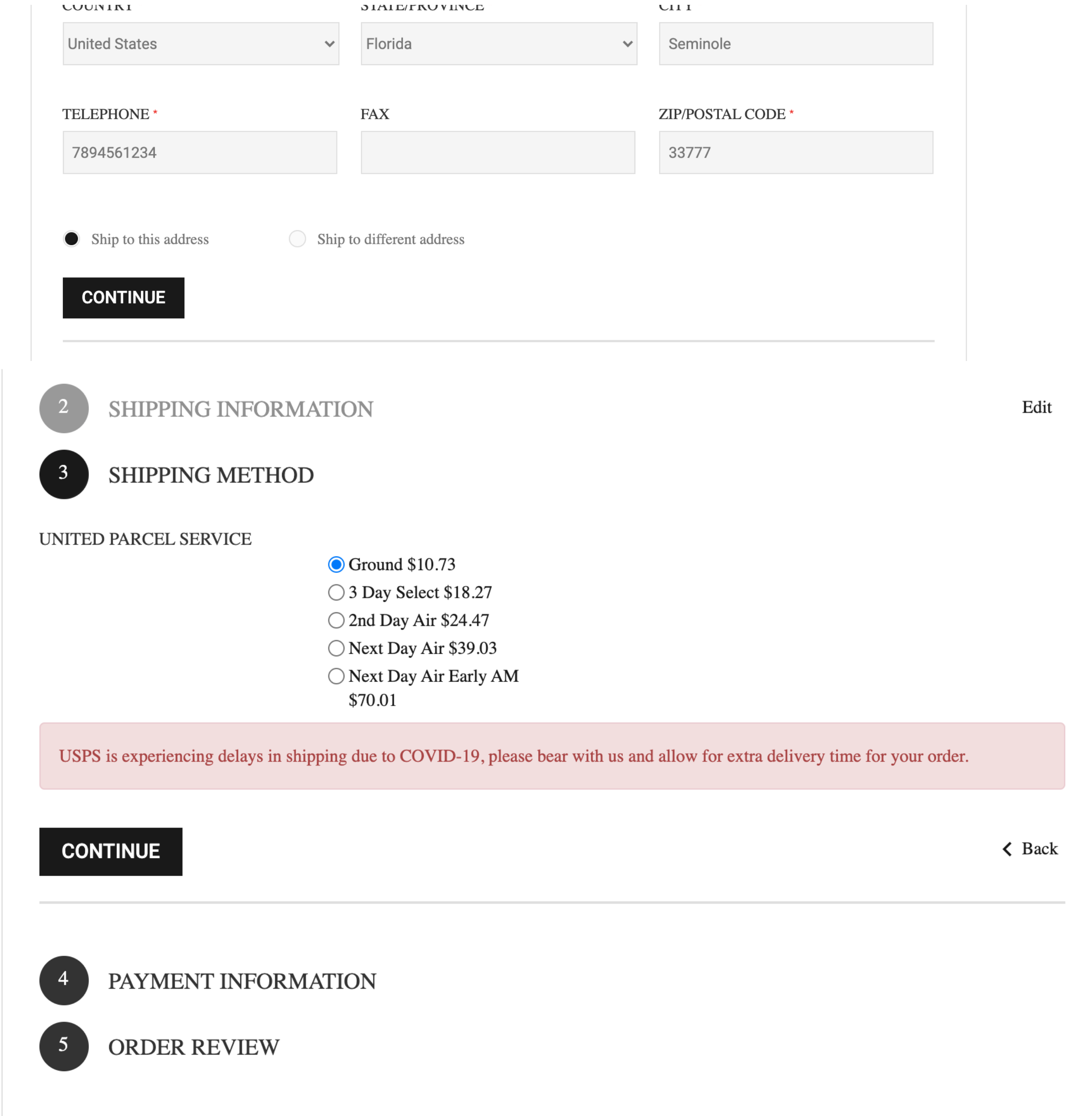
	Oaks and Aiea		
57	UK 20 - Nola 5 - Special 5 = 5		
58	UK B level split Nola - 5 Special 5 - 5		
59	F Level Special 5		
60	D Level Ames split Valley Junction and Des Moines	**Non-standard:** **Orders from the Ames, IA store result in the following credits:** Davenport store gets 10% Des Moines store gets 5% Ankeny store gets 5%	

SunFlora PO Box Update

This page will cover the update to the shipping/billing information implemented to correct orders from having a shipping address as a PO box. Currently, the SunFlora site has no handling for this entry and allows the user to enter the po box either directly in the shipping information, or when using the “Use Billing Address” checkbox.

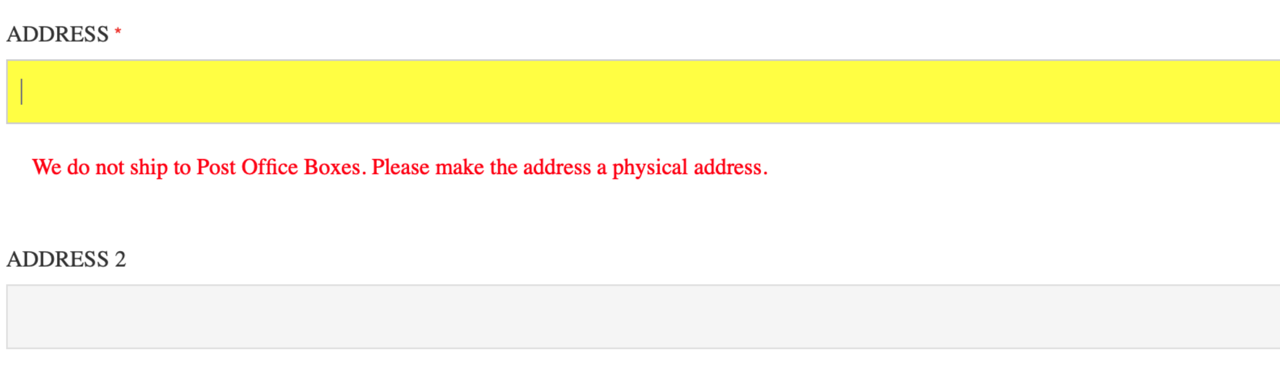
SunFlora Site Without PO Check





After the implemented change, the regex used alerts the user when they attempt to use any variation of PO box, P.O box, p.o. box, etc. that "we do not ship to Post Office Boxes". This is performed when attempting to use the checkbox "Use Billing Address" or when entering the PO box manually into the address field.

SunFlora Site With PO Box Check



CITY * ZIP/POSTAL CODE * TELEPHONE *

Seminole | 33777 | 7273125456

STATE/PROVINCE * COUNTRY * FAX

Florida | United States |

Use Billing Address

CONTINUE

Back

ADDRESS *

We do not ship to Post Office Boxes. Please make the address a physical address.

ADDRESS 2

CITY * ZIP/POSTAL CODE * TELEPHONE *

Seminole | 33777 | 7273125456

STATE/PROVINCE * COUNTRY * FAX

Florida | United States |

Use Billing Address

CONTINUE

Back

You have selected a Post Office Box address. We do not ship to PO Boxes. Please select a different address or select New Address from the drop down list.

OK

1 BILLING INFORMATION Edit

2 SHIPPING INFORMATION

SELECT A SHIPPING ADDRESS FROM YOUR ADDRESS BOOK OR ENTER A NEW ADDRESS.

Grace Test, po box 7878, Seminole, Florida 33777, United States

Use Billing Address

CONTINUE

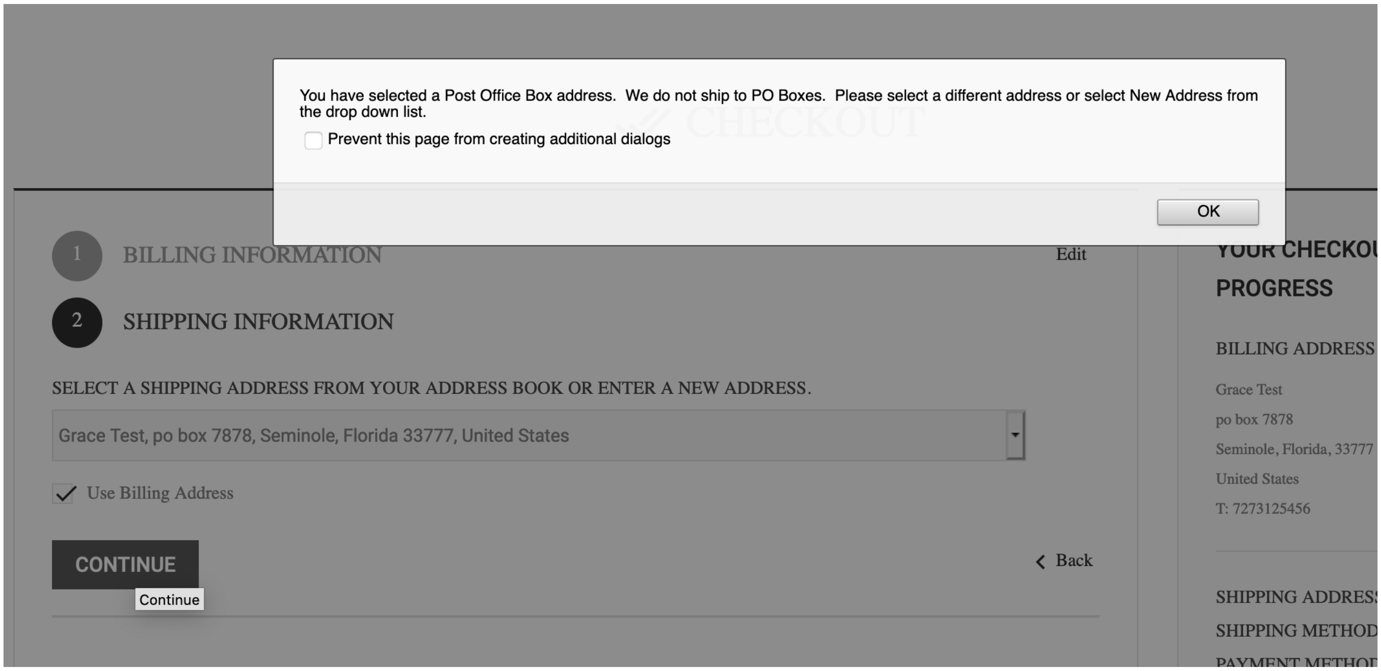
Back

YOUR CHECKOUT PROGRESS

BILLING ADDRESS

Grace Test
po box 7878
Seminole, Florida, 33777
United States
T: 7273125456

SHIPPING ADDRESS



Windows Server GCP VM Business Plan

This document serves as a fundamental reference for cross-comparing different options within the Google Cloud Platform for Windows Servers to determine the optimal cost, hardware, and software. Additional information regarding the data supported can be found under the References location towards the bottom of the document. Each item that requires more information to fully understand will be referenced with a superscript number to identify within the references what it means.

**** Any superscript links can be utilized by opening the link in a new browser tab; anchors were stripped from confluence and, therefore, offer that as the only solution.****

Current Server Requirements

- 1 Dual-core Processor
- 16 GB RAM
- 1 TB Storage
- 20 GB per month

Windows Server Specifications

Available Windows Server Options[1]				
Products	**Details**			
Windows Server 2012 R2	Updates to PowerShell	Enhanced functionality for storage	Ability to server software to mobile devices	R2 version was first to allow integration with Azure
Windows Server 2016	Inclusion of Nano Server[2]	Network Controller[3]	Support for using containers[4]	Aforementioned inclusions to 2012 on this one also
Windows Server 2019 **RECOMMENDED**	Windows Admin Center[5]	Hyper-converged infrastructure[6]	Advanced Threat Protection	Includes all features from Windows Server 2016 and prior versions

Windows Server 2019 Versions			
Version	**Details**		
Datacenter	Supports Shielded VM Features	Provides Host Guardian Service	Supports GUI
Datacenter Core	Supports Shielded VM Features	Supports unlimited number of virtual machines like Datacenter version	No GUI support
Datacenter for Containers	Supports Shielded VM Features	Comes with Docker EE preinstalled[7]	No GUI support
Datacenter Core for Containers	Supports Shielded VM Features	Comes with Docker EE preinstalled	No GUI support

Google Cloud Machine Information

Google Machine Types			
E2 **General purpose** **RECOMMENDED**	**N2, N2D, N1** **General purpose**	**M2, M1** **Memory-Optimized**	**C2** **Compute-Optimized**
Day-to-day computing at a lower cost	Balanced price/performance across a wide range of VM shapes	Ultra high-memory workloads	Ultra high performance for compute-intensive workloads
• Web serving • App serving • Back office applications • Small-medium databases • Microservices • Virtual desktops • Development environments	1. Web serving 2. App serving 3. Back office applications 4. Medium-large databases 5. Cache 6. Media/streaming	1. Large in-memory databases like SAP HANA 2. In-memory analytics	1. HPC 2. Electronic Design Automation (EDA) 3. Gaming 4. Single-threaded application

E2 Standard Machine-Types						
Machine name	**vCPUs**	**Memory (GB)**	**Max number of persistent disks (PDs)**	**Max total PD size (TB)**	**Local SSD**	**Network egress bandwidth (Gbps)**
`e2-standard-2`	2	8	128	257	No	4
`e2-standard-4`	4	16	128	257	No	8
`e2-standard-8`	8	32	128	257	No	16
`e2-standard-16`	16	64	128	257	No	16

Financial Evaluations

Machine Costs With Windows[8]

e2-standard-2 (vCPUs: 2, RAM: 8GB)

730 total hours per month
Total RAM: 16GB
Paid OS Cost: $67.16
GCE Instance Cost: $48.92
Total Estimated Monthly Cost:

USD $116.08

e2-standard-4 (vCPUs: 4, RAM: 16GB)

730 total hours per month
Total RAM: 64GB
Paid OS Cost: $134.32
GCE Instance Cost: $97.84
Total Estimated Monthly Cost:

USD $232.16

e2-standard-16 (vCPUs: 16, RAM: 64GB)

730 total hours per month
Total RAM: 1024GB
Paid OS Cost: $537.28
GCE Instance Cost: $391.35
Total Estimated Monthly Cost:

USD $928.63

Machine Costs With Linux

e2-standard-2 (vCPUs: 2,RAM: 8GB)

730 total hours per month
Total RAM: 16GB
GCE Instance Cost: $48.92
Total Estimated Monthly Cost:

USD $48.92

e2-standard-4 (vCPUs: 4, RAM: 16GB)

730 total hours per month
Total RAM: 64GB
GCE Instance Cost: $97.84
Total Estimated Monthly Cost:

USD $97.84

e2-standard-16 (vCPUs: 16, RAM: 64GB)

730 total hours per month
Total RAM: 64GB
GCE Instance Cost: $391.35
Total Estimated Monthly Cost:

USD $391.35

Remote Cloud Storage[9]

Cloud Storage for Backups

Total Amount of Storage: 1,024 GiB
Ingress Cost: Free
Egress Cost: $0.01/GB (US-EAST1 to NORTHAMERICA-NORTHEAST1,US-EAST1 to US, etc**)**
Total Estimated Monthly Cost:

USD $20.48

Windows Server Options

Lowest Cost Monthly
RECOMMENDED

Machine: e2-standard-2 (vCPUs: 2, RAM: 8GB)
Storage: Remote Cloud Storage

Cost_Type	Cost
Paid OS Cost:	$67.16
GCE Instance Cost:	$48.92
Storage Cost:	$20.48
Total Cost: $136.56	

Medium Cost Monthly

Machine: e2-standard-4 (vCPUs: 4, RAM: 16GB)
Storage: Remote Cloud Storage

Cost_Type	Cost
Paid OS Cost:	$134.32
GCE Instance Cost:	$97.84
Storage Cost:	$20.48
Total Cost: $252.64	

Highest Cost Monthly

Machine: e2-standard-16 (vCPUs: 16, RAM: 64GB)
Storage: Remote Cloud Storage

Cost_Type	Cost
Paid OS Cost:	$537.28
GCE Instance Cost:	$391.35
Storage Cost:	$20.48
Total Cost: $949.11	

Linux Server Options

Lowest Cost Monthly
RECOMMENDED

Machine: e2-standard-2 (vCPUs: 2, RAM: 8GB)
Storage: Remote Cloud Storage

Cost_Type	Cost
GCE Instance Cost:	$48.92
Storage Cost:	$20.48
Total Cost: $69.40	

Medium Cost Monthly

Machine: e2-standard-4 (vCPUs: 4, RAM: 16GB)
Storage: Remote Cloud Storage

Cost_Type	Cost
GCE Instance Cost:	$97.84
Storage Cost:	$20.48
Total Cost: $118.32	

Highest Cost Monthly

Machine: e2-standard-16 (vCPUs: 16, RAM: 64GB)
Storage: Remote Cloud Storage

Cost_Type	Cost
GCE Instance Cost:	$391.35
Storage Cost:	$20.48
Total Cost: $411.83	

Footnotes

[1]*The ones listed are the ones that include a GUI for simplified usage and navigation.*
[2]*Nano Server is a stripped-down version of Windows Server for running cloud applications. This removes the graphical interface, 32-bit support, and server core components to improve security and dramatically reduce resource consumption*
[3]*Allows all devices — physical or virtual — from a singular location*
[4]*Enhances support for Hyper-V by interacting with Docker for additional encryption*
[5]*Provides tools for central management such as performance monitoring, configuration management, and management of services on different servers; this tool is also able to support servers with different operating systems along with being backwards compatible for 2012 and 2016 Windows server editions.*
[6]*Allows network administrators to manage virtualized systems without taking down instances, automated VM adjustments, and supports software-defined data centers and their customers.*
[7]*Docker EE can be installed within the Windows Server 2019 Datacenter*
[8]*All Windows servers are the same cost and, therefore, is not listed*
[9]*Storing data on the machine instance increases the cost by approximately $250.00 per machine and has been factored out as an option due to feasibility*

References
https://www.dnsstuff.com/windows-server-versions-guide
https://cloud.google.com/products/calculator
https://cloud.google.com/compute/docs/machine-types
https://cloud.google.com/storage/pricing
https://www.quickstart.com/blog/comparison-of-windows-server-2016-and-2019-what-is-the-difference/#:~:text=Compared%20to%20the%202016%20version,for%20encryption%20of%20network%20portions.
https://medium.com/@RealNetwork/windows-server-2019-product-family-editions-options-and-versions-explained-plus-comparison-6396f04e38d

WHMCS (Web Host Manager Complete Solution)

This page will contain information regarding WHMCS and details catering towards it processes. Additional pages may be needed to contain the complexity of this service. As needed, those items will be linked throughout this document as well.

Navigational Guides Created

How To Customize Billing.Dang-designs

References	
URL	**Comments**
https://www.whmcs.com/	WHMCS's site for additional information
https://docs.whmcs.com/Documentation_Home	The documentation root (YASSSSS)

WHMCS Support Ticket Setup

This page will document the necessary steps to create the WHMCS support tickets to replace the Jira Service Desk. Firstly, we must determine what services Jira Desk offers so that we can fully understand what WHMCS will be replacing, and what will need to be implemented to ensure that this is successful.

WHMCS Needs

1. Email integration with gmail account for ticket responses and actions **IN PROGRESS**
2. Ring central connection to send over call logs **TO DO**
3. CRM add-on to be configured **TO DO**

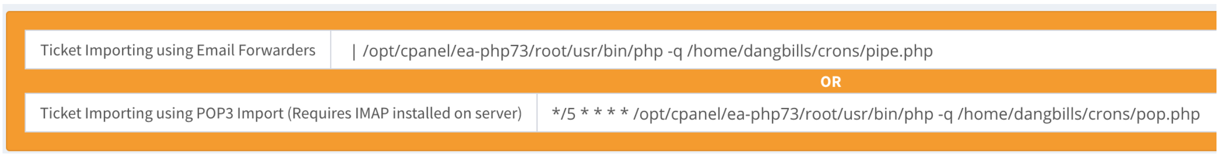
Steps to create support ticket setup.

1. **Set up email support ticket department**

a. **IMPORTANT NOTE FOR TESTING:** dang-design emails cannot be used for testing as this will provide a log stating "Ticket ID Not Found"

2. **Since the email accounts that can be created are based on a separate domain, the POP/IMAP option is the only one available**
    a. Path is specified on Support department creation page for cron jobs
    b. Make sure to include below code so that the cron job doesn't bombard the inbox AND so that the ruleset corrects to not send blank messages to the inbox.

```
>/dev/null 2>&1
```

    c. 

3. Ensure that the settings for the new Gmail account have IMAP and POP enabled (disabled by default)
    a. While they are disabled by default, Duve confirmed that SSL for both are enabled by default.
4. Test email account to ensure working as expected by sending to support ticket section
    a. If errors, go to file manager to pipe.php and update file to permissions 755
5. Other steps may be needed, but those will be listed after further research has been completed.

References	
URL	**Comments**
https://www.hostwinds.com/guide/how-to-setup-and-configure-whmcs-support/	How to setup and configure WHMCS Support
https://www.modulesgarden.com/products/whmcs/crm#features	CRM Information
https://help.whmcs.com/m/support_tools/l/679784-troubleshooting-email-piping-problems	Troubleshooting information for email piping (includes information for logs and POP details).

Yootheme ZOO-AMP Pages

AMP, in a nutshell, is a framework for extremely fast-loading mobile pages. It is a Google project. AMP HTML is an extremely slimmed-down version of HTML and therefore works extremely quickly on mobile.

ZOO Amp is a Joomla plugin that works alongside your Yootheme ZOO installation to create Accelerated Mobile Pages (AMP) of your ZOO items. To read more about Google AMP go to the Google AMP homepage.

ZOO Amp runs your ZOO items through a process and creates AMP-compatible HTML pages… and it happens automatically in the background.

IMPORTANT NOTES:

Google AMP

- For AMP pages to be accessed, Google must first index your AMP pages. It does this by looking for an 'amphtml' link in your <head> code. (more on this later)
- AMP pages are not accessed directly on your site. When Google indexes your AMP page and serves it up on the search, it actually shows the user a google hosted page.
- AMP can carry lots of the usual functionality of a normal web page, like ads, social links, images, and video, so users shouldn't get a worse experience.

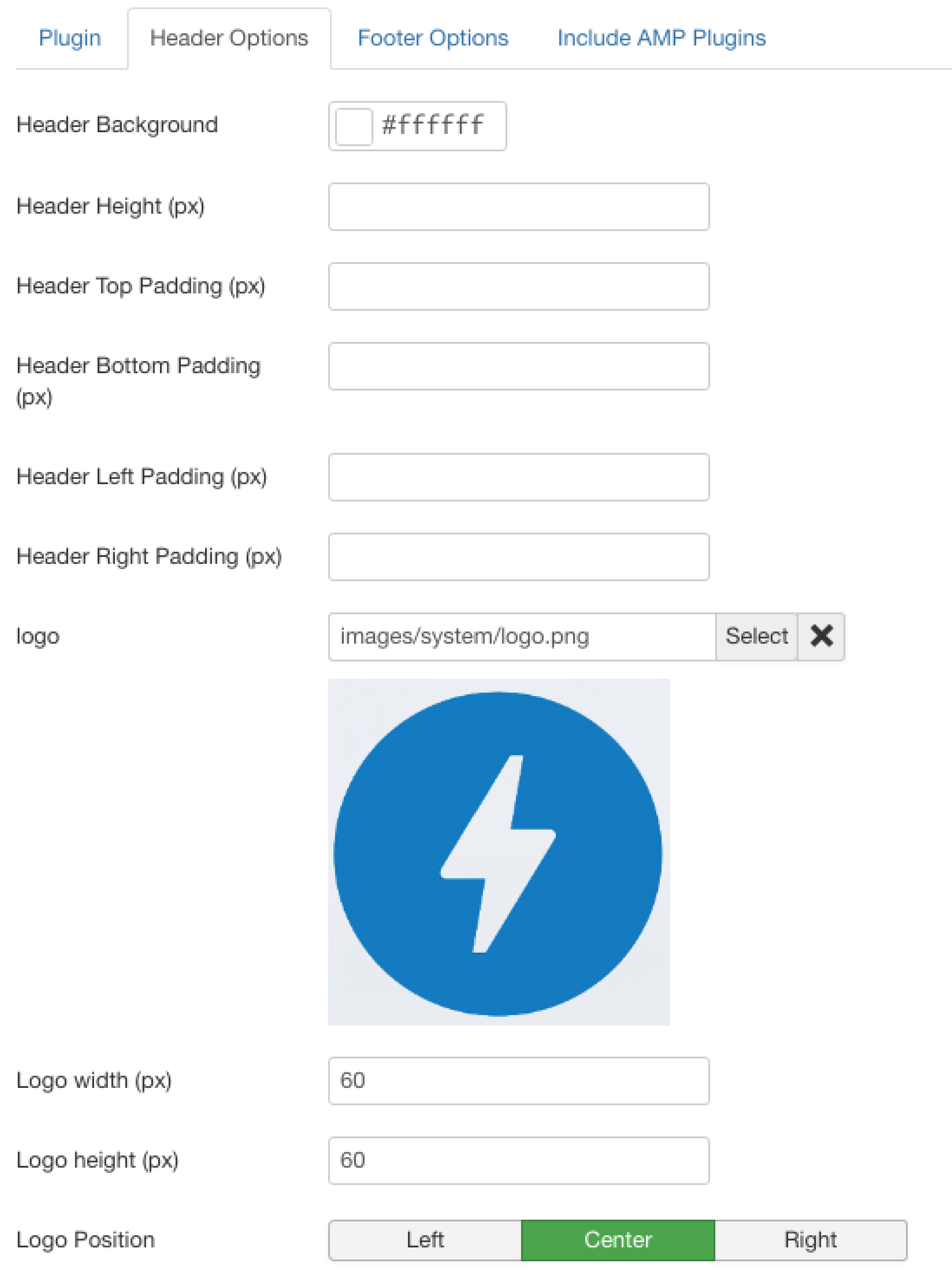
ZOO Amp

- ZOO Amp will only create AMP pages for the layout types you need. Leave the layout blank and no AMP page will be created.
- ZOO Amp automatically adds the 'amphtml' link to the <head> of the normal webpage.
- You can style your AMP page how you like using a combination of Plugin Parameters and a Custom CSS ZOO element.
- ZooModsPlus have created some custom elements to help support ZOO Amp pages, there are:
    - ZooAmpGooleAdsense
    - ZooAmpSocialShare
    - ZooAmpFacebookLike
    - Google Maps Element
- ZOO Amp "runs" when you save an item, so changing style setting in the plugin parameters or the Custom CSS element, will only take effect when you save the item. All previously saved versions will remain unchanged.

ZOO-AMP Plugin Setup (Joomla>Extenstions>Plugins>Zoo AMP **)**
The plugin parameters will allow you to determine the basic style of the AMP page.

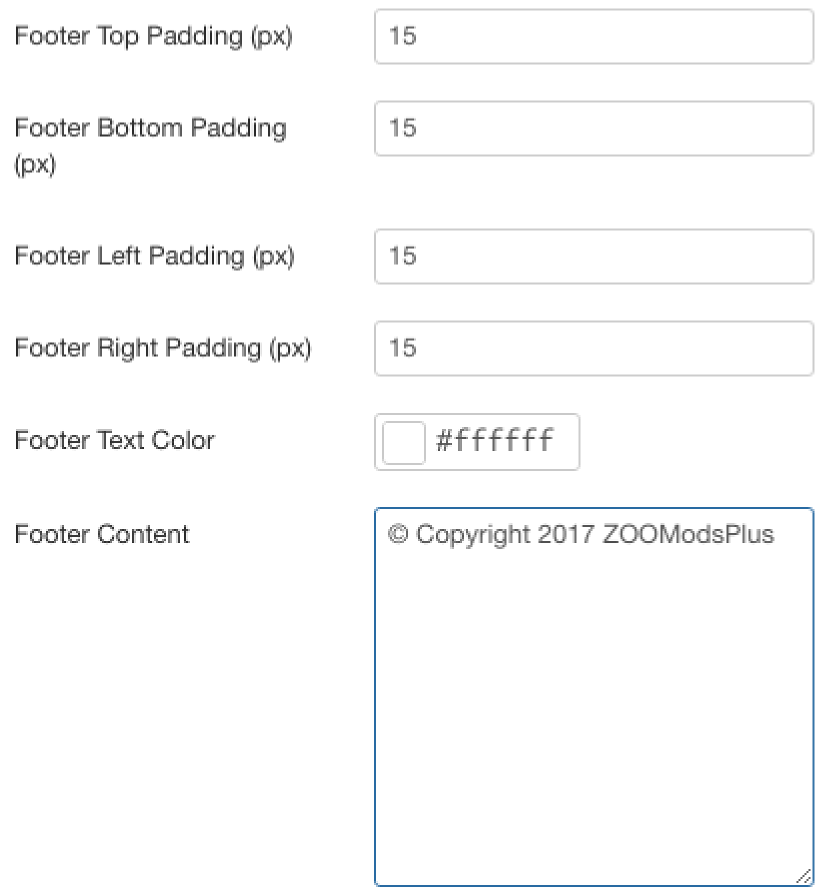
Header Options

Under the “Header Options” tab, you will see a number of parameters to set up the AMP Page header, including the header, background color, and even the logo:



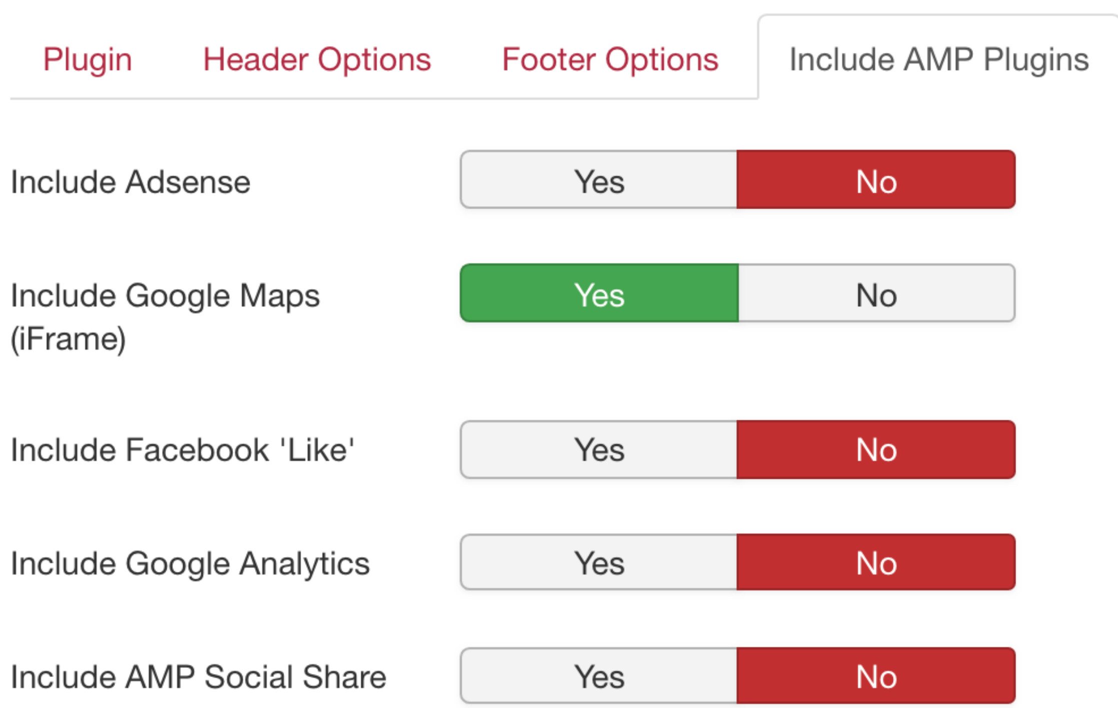
Footer Options

Under the “Footer Options” tab, you will see a number of parameters to set up the
AMP Page footer, including the background color and copyright info:



Include AMP Plugins

Under the “Include AMP Plugins” tab, you will see a number of parameters that determine whether you want to include certain AMP plugins for AMP services. Select the ones you want to load:

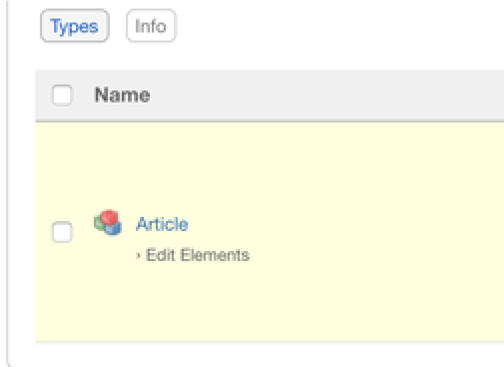


ZOO Setup

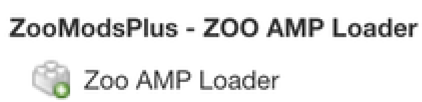
1: Connecting your web page to your AMP page

The first step in getting your ZOO Amp system to work is to create the “ZOO Amp Loader” element and place it in your “full” view. This will insert the required code in the <head> of your page and let Google know that you have an equivalent.

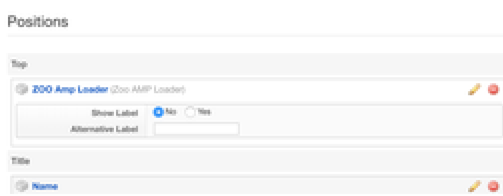
1. Go to ZOO and click the “gear” icon to access your ZOO settings.
2. Click the app you wish to configure
3. Edit the elements of the type you wish to configure:



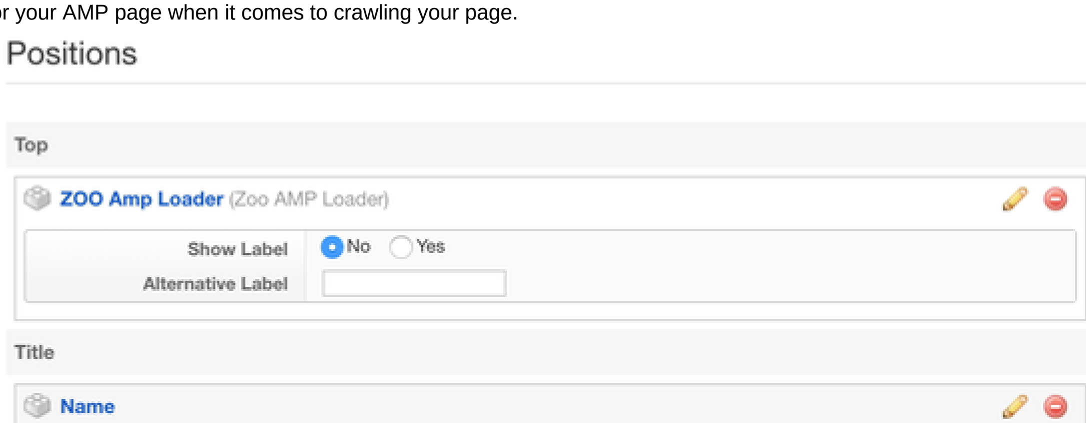
4. Now create the “ZOO Amp Loader” element:

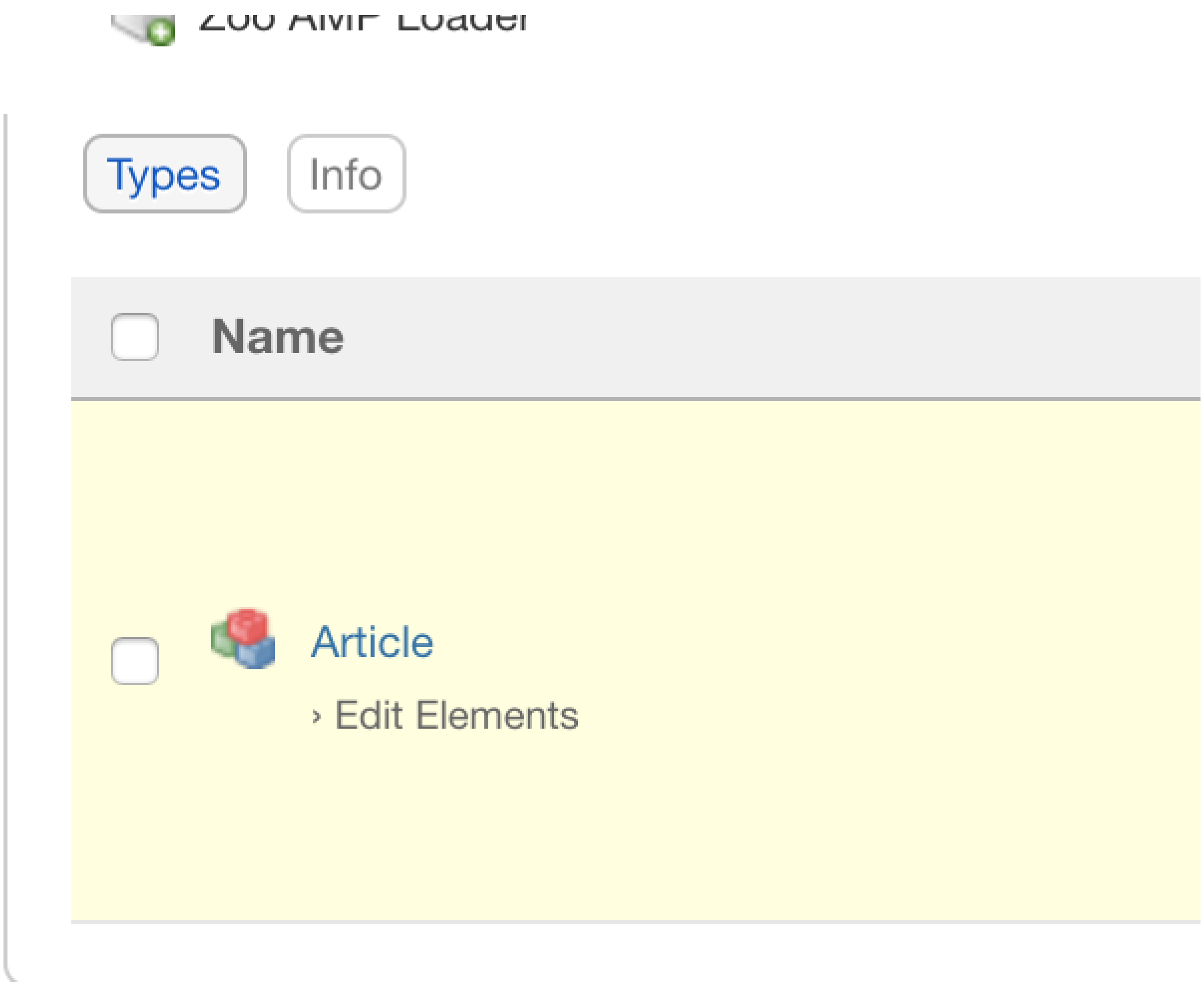


5. And now add it to your full layout:



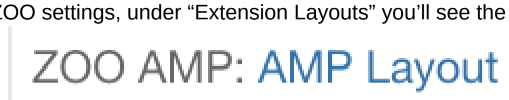
IMPORTANT NOTE: We’ve added the element to the ‘top’ position, however, it doesn’t matter where you add this element, it can be added to any position in the full layout. Your frontend view won’t change. Now that this element is created and placed, Google will now look for your AMP page when it comes to crawling your page.





2: Adding content to your AMP page layout.

In ZOO settings, under “Extension Layouts” you’ll see the layout for ZOO Amp:

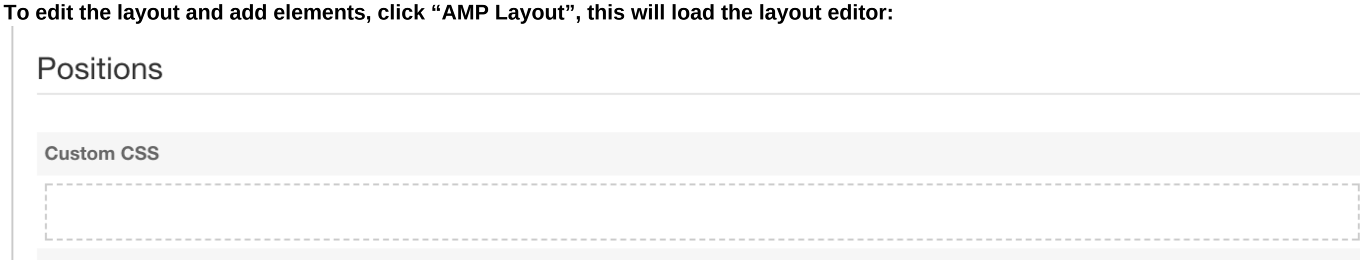


There are 8 distinct sections for you to use in putting together your AMP page.

- Title
- Meta
- Section 1
- Section 2
- Section 3
- Section 4
- Section 5
- Section 6

This should give you enough scope to design an AMP to suit any website.

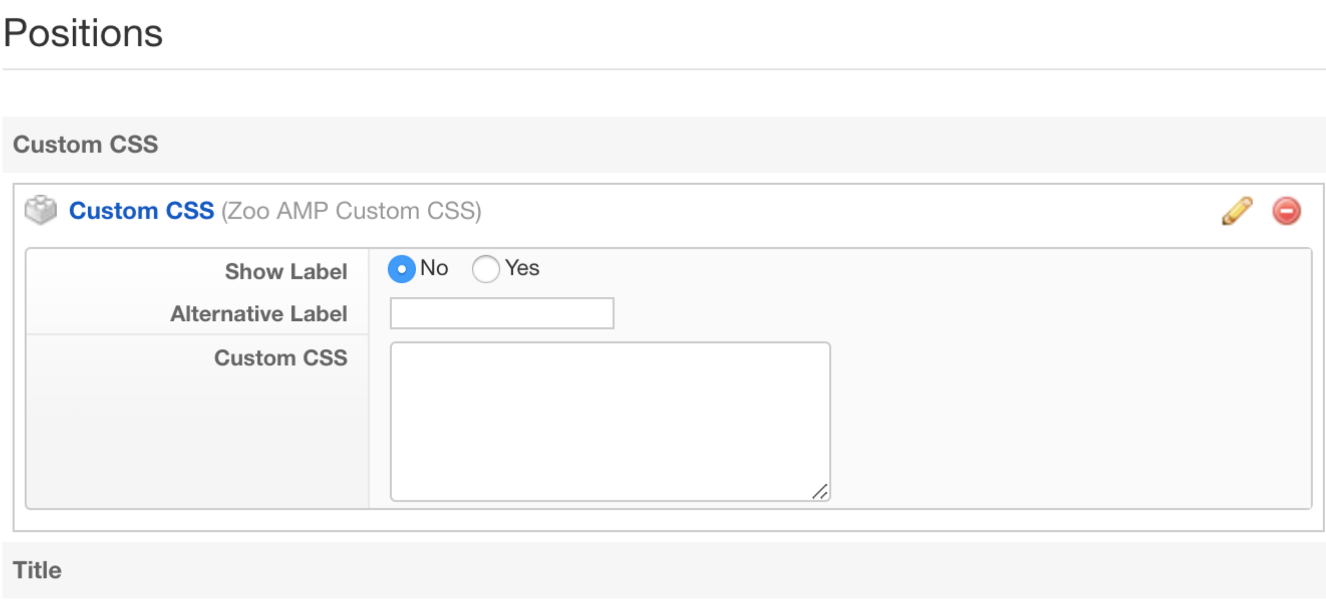
To edit the layout and add elements, click “AMP Layout”, this will load the layout editor:



Now you can drag and drop your content elements into the predefined positions.

- Using the “Custom CSS” position

  You will notice a position at the top of the layout named “Custom CSS”. This is to be used only with the “Zoo AMP Custom CSS” element. Drag this element into this position (and ONLY this element in this position - no other element will work there and will probably break your AMP page).



Now edit the element by clicking the pencil icon. This will open a “Custom CSS” box. This is where you can write your custom CSS.

There are a few built-in classes you can target to style your AMP sections:

- amp-header - this targets the header of the page, where the logo is located
- *amp-main* - this targets the main content div and wraps all the layout sections
- *amp-title* - this targets the 'title' layout position
- *amp-meta* - this targets the 'meta' layout position
- *amp-section1* - this targets the 'section 1' layout position
- *amp-section2* - this targets the 'section 2' layout position
- *amp-section3* - this targets the 'section 3' layout position
- *amp-section4* - this targets the 'section 4' layout position
- *amp-section5* - this targets the 'section 5' layout position
- *amp-section6* - this targets the 'section 6' layout position
- *amp-footer* - this targets the footer of the page, where the copyright is located

Navigation Guides

This page serves no other purpose than to be the parent and link to all navigational guide documents.

- Adding Google AdSense to a Joomla Module Using Custom HTML.
- AWeber Email Marketing
  - AWeber Account Setup & Testing
  - How to use AWeber Campaigns
  - How to use AWeber Email Templates
  - Sending a Broadcast Message from AWeber
  - Migrating Subscribers Between Lists
- Billing.DANG-designs
  - How To Add New Leads & Discovery Call Appointments
  - How To Customize Billing.Dang-designs
  - How To Enter New Call/Meeting Event In Google Calendar/CRM
  - How To Perform CSS Updates On WHMCS
  - How To Update Payment Information For Clients
- Capitalcityshoes
  - Capitalcityshoes Custom Pieces, Modules, & Extensions
  - How To Add A New Sub Store/View
  - How To Add New Punchout Catalog
  - How To Set Up SMTP On Magento 2 For SendGrid
  - How To Update Punchout Site URL On Ariba
  - How To Update Sandbox For Capitalcityshoes With Live Files
  - Payment Restrictions Per Website
  - How To Add Extension On Magento 2
  - How To Import New Users
  - How To Correct User To Be Charged Taxes (Customer Groups)
  - How To Update User Passwords & Advise Them
  - RICS Access
- CBD
  - Setting Up A CBD Store on Sunflora.org
  - Setting Up A New Store In Salesforce
  - Setting Up A New Subdomain View In Google Analytics
  - Setting Up A Product In Magento (Simple & Bundle)
  - How To Set Up Subdomains For CBDRX4U
  - Connect A Fishbowl Client Remotely to the SANDBOX Server
  - How to Add Pictures to Store in CBDRX4U
  - *Confidential* Referral Credit Structure - Founders
  - Creating A Branded Email
  - How To Redirect AMP Page For Closed Stores
  - How To Set Up A 3PL Order Import Template
  - How To Set Up A New Magento Website Domain
- Code Pieces
  - GAS Projects
    - Convert GMT Column To EST
    - Get Web Dev Measure Data From Lighthouse Client
    - OnChange GMT To EST Row Conversion
    - Query Filter Build From Zapier Imports (Harden-Offroad)
    - Send Email When Plugins Expire
    - Sunflora - Group Stores With Primary Parent
    - Watchful List Data Separator (Multiple Sites)
    - Watchful List Data Separator (Singular Site)
  - Java Selenium Automation Projects
    - Create A List of Write-A-Review Links For Stores
    - List All Website Expiration Dates
    - Setting Up GTMetrix Website Testing Automation
    - Setting Up Pingdom Website Testing Automation

- Website @media Responsive Queries
- Where To Update IP Addresses On Change

Adding Google AdSense to a Joomla Module Using Custom HTML.

Issue

Unable to display Google Adsense using **Custom Module**

1. I see blank display though I pasted the code in Custom Module.
2. I see blank display though I pasted the code using code layout in Custom Module.
3. The code disappears upon saving.

Resolution

The **Custom Module** that is accompanied by the Core Joomla is designed to execute the raw codes and scripts. Click Here to see steps to generate Google Adsense Code.

To get this functioning, you need to follow the following steps carefully:

1. Log on to the back end of your Joomla website with an Administrator account.
2. Go to **System > Global Configuration**
3. Change the **Default Editor** to **'editor – no editor'**.
    - If you don't change the default editor to no editor, the WYSIWYG editor will change the JavaScript, even if you edit the HTML source.
4. Go to **Extensions > Modules**
5. Now click on **New** button to create a new module instance and select **Custom** from the list.
6. Paste the Google AdSense code here and configure the position and other options as required by you.
7. **Save & Close** the **Custom** module.
8. Now go to **System > Global Configuration** & change the **Default Editor** back to what it was or an Editor of your choice.
9. Now go to **System > Clear Expired Cache** and click on **Clear Expired Cache** button.

Go to the site's front-end and verify whether your Google Ad is loading up.

Please Note: *If you generated the code a few minutes ago, please allow some time for Google to start pushing Ads on your site. It usually works instantly, however at times may take up to an hour time to begin displaying Ads.*

AWeber Email Marketing

AWeber Account Setup & Testing

1. Initial customer account setup.
    a. Use clients email for billing
    b. Use websites master email (non-DANG address) for sending broadcasts.
    c. AWeber admin use dev@dang-designs.com
2. Add domain access for the sending email account.
    a. Check SPF, DKIM, and DMARC entries (Dan Help)
3. Test sending a broadcast email within AWeber.
    Check received email for passing DKIM signatures using these instructions. https://www.appmaildev.com/en/dkim

Test MX Records: https://mxtoolbox.com

How to use AWeber Campaigns

Campaigns are AWeber's email automation platform. With Campaigns, you can send the right message to the right person at the right time. This not only saves you time but makes creating targeted campaigns simple to set up.

Campaigns can be built using the Actions available - Trigger, Send a Message, Wait, and Apply Tag. Each of the action items can be dragged and dropped onto your campaign canvas. You can also configure your Campaign to allow for multiple entries, which means subscribers can repeat a Campaign they have already completed.

Step By Step Instructions

1. If you haven't done so already, create the messages you would like to be added to your campaign. You can create your messages by hovering over the Messages tab and clicking Drafts.

    Here, click the "Create a Message" button and select the message editor you would like to use to create your message. Our most popular editor is the "Drag & Drop Email Builder."

2. Once your messages are created, hover over the Messages button and click Campaigns.

Then, click the "Create a Campaign" button.

You will be given the option to create a brand new campaign, import an existing campaign, or choose from one of our campaign templates. Click "Create a blank campaign" to get started creating a new campaign.

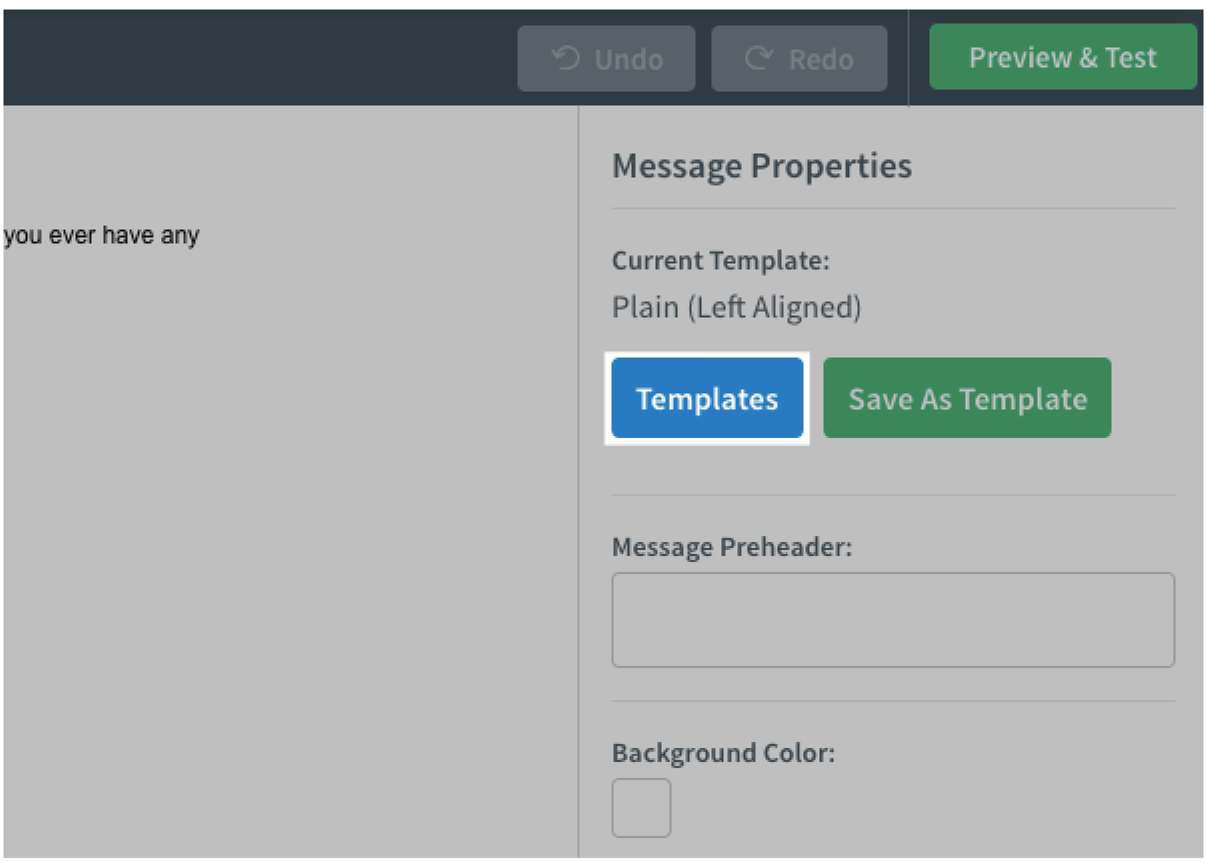
1. Provide your campaign with a name, and then click the "Create" button.
2. When you first create your campaign, you're provided with the "Trigger." Using the "On Subscribe" option, you can select if you would like subscribers to begin your campaign based on their subscribing method. Using the "Tag Applied" option, you can select if you would like subscribers to begin your campaign based on when a tag is applied to the subscriber's information.
3. Once your trigger is selected, you can start adding "Actions" onto your "Canvas." The canvas is the dark grey area where action items can be placed. Each new action dropped onto the canvas will serve as the next action to be taken with the subscriber.
4. To build your campaign, choose the different "Actions" from the left hand sidebar and drop them onto the canvas. The different "Actions" available for campaigns are "Send a Message," "Wait," and "Apply Tag."
5. To customize any of your actions, click the action on your canvas and update the "Settings" located in the right-hand sidebar.
6. If you ever need to reorder the actions in your campaign, you can do so by dragging the action from the canvas and dropping it into the desired place. When moving actions around, the remaining actions will be bumped down in the campaign.
7. Once you've finished creating your campaign, click the "Save & Exit" button located in the bottom right-hand corner of your page.
8. Once you're ready for the campaign to be active in your list, click the "Draft" tab and then click "Activate Campaign."
9. If you ever want to change the status of your campaign, you can do so by clicking the drop-down menu for the campaign's "Status." The different statuses available are "Active," "Pause," "Close," and "Stop."

   Once your campaign is closed or stopped, the campaign cannot be reactivated or paused.

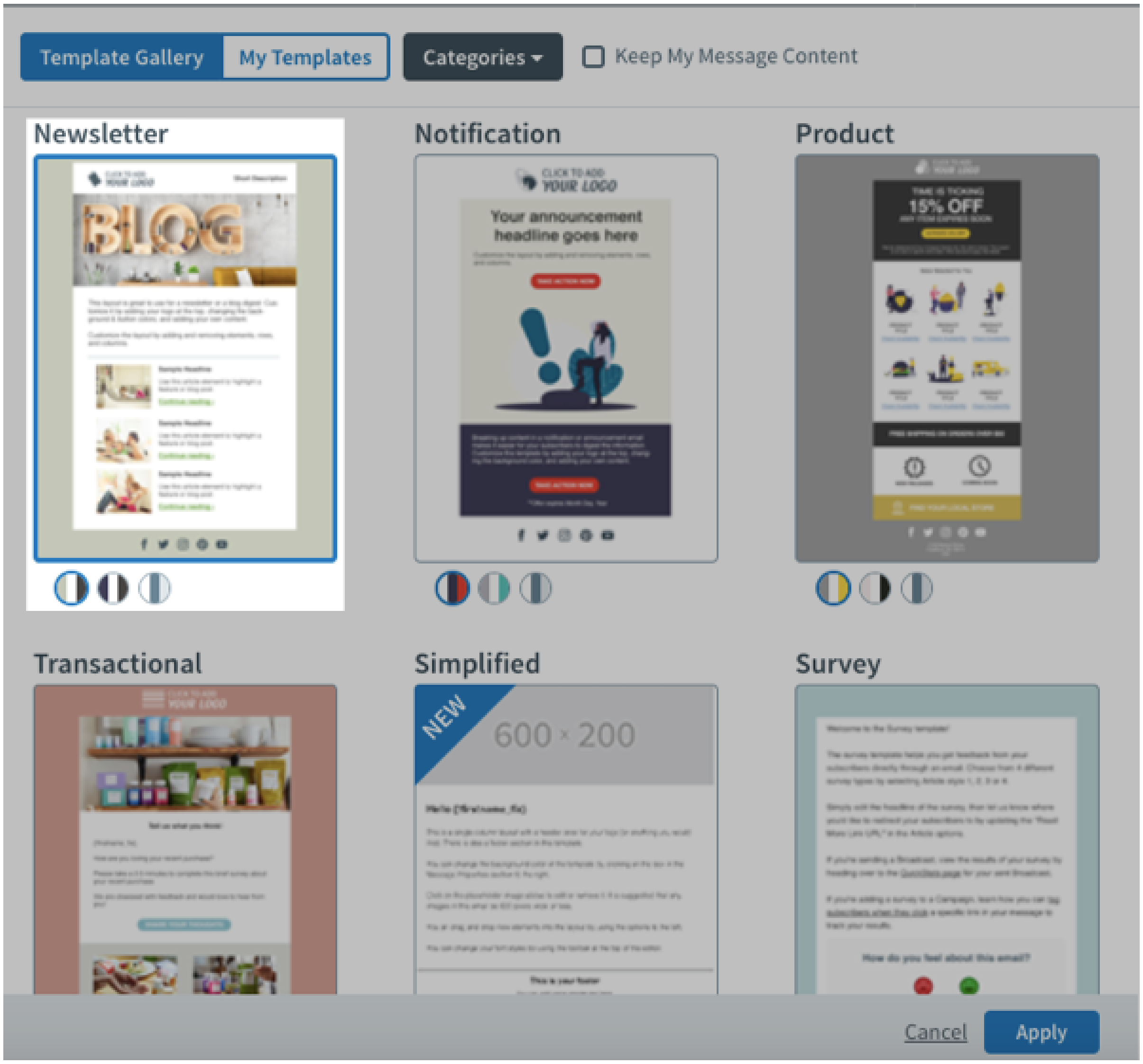
How to use AWeber Email Templates

Step By Step Instructions

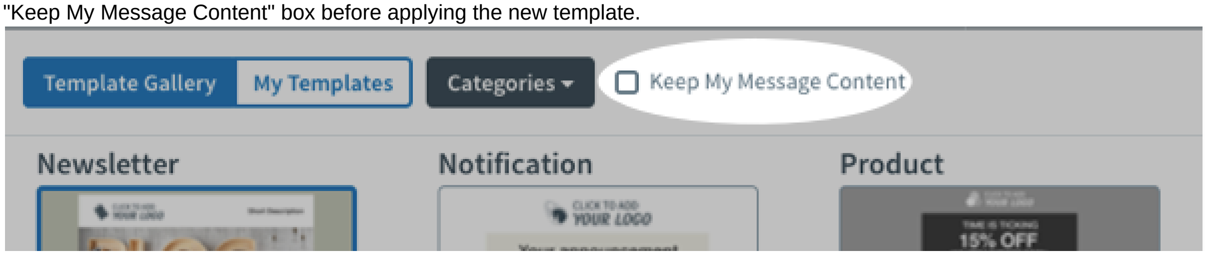
1. When creating a follow-up message or broadcast, from the right-hand sidebar, click the "Templates" button. This will be located under your "Message Properties" section.
   By default, you'll look through AWeber created templates, but if you've created templates of your own, you can switch to them by clicking the "My Templates" button.



2. Click the template you would like to apply to your message. Then, click the "Apply" button.



3. If you'd like to erase whatever content you may have already placed in the message and start over with dummy content, simply uncheck the "Keep My Message Content" box before applying the new template.



Sending a Broadcast Message from AWeber

Use broadcasts to send:

- Newsletters
- Sales Offers
- Seasonal Messages
- Other Date and Time-Specific Mailings

Step By Step Instructions

1. Hover over the Messages tab and click Broadcasts.

   Here, click the "Create a Message" button.

   From the drop-down menu that appears, click "Drag & Drop Email Builder".

2. At the top of the message editor, enter the subject line for your message.

   Note: You can add personalization fields using the "Personalize" button to the right.

   Next, click the "Templates" button located to the right-hand side of the message editor. This will be located under your Message Properties section.

   Many templates have color schemes you can select from located below the template itself. Once you've found the template that you'd like to work with, click the "Apply" button at the bottom of the sidebar.

3. Now it's time to start using the Drag & Drop Email Builder. The editor is made up of two major components: sections and blocks.

   Sections are the areas in a message where you can place your content. Templates come with a number of sections already in place, usually with some dummy text or images included. You can create new sections for content by dragging a Row element into the body of the message to create a new row. You can also click "Add Col Left" or "Add Col Right" from the Row properties on the right-hand side of the editor to add new columns.

   This allows you to add content to the new section of your message.

   Note: You can also change the alignment of the content in the section: sections can be top, center, or bottom aligned.

   Blocks are the actual content in your message (text, images, etc). You can drag and drop these blocks to move them around. To start working with a block, simply click on it. If you wanted to move the block, click and hold down the "Move" option. You'll be able to drag and drop the block into the section you wanted.

   Once you've clicked on the block, you'll see the options to "Copy" and "Delete". If you need to use either of these options, simply click the button and the selected action will occur.

4. To add new blocks to your message, simply drag them from the left-hand sidebar into your message.

   The Headline block creates a headline for your messages. Simply click on the block once you've placed it to edit the text.

   The Paragraph block creates a block of text. Again, just click to edit the text once you've placed the paragraph block.

   The Article block creates a headline, a text area, and a "read more" link. Click the text to edit it, and set the URL of the link in the window to the right. Some templates may include an image as well - if that's the case, you can edit the image via the window to the right as well.

   The Image block allows you to insert an image into your message. The window to the right allows you to upload an image directly to AWeber or enter the URL of the image you'd like to display. You can also have the image link to another page when clicked, and provide the text that will appear when someone hovers their cursor over the image.

   The Video block allows you to add a YouTube or Vimeo video to your message. This will take a screenshot of your YouTube or Vimeo video and hyperlink the image with the URL of the YouTube or Vimeo video.

   The Button block creates a clickable button. To the right, you can change the URL the button is linked to, change the button label, text style, button color, or replace the button with another image.

   The Follow Me block adds buttons to allow readers to follow you on Facebook and Twitter - simply add the social media accounts using the window to the right.

   The Product block creates an image which you can replace with an image of your product, along with text for the product's name, description, and price. Finally, it comes with a button that can take customers to your order page. The image and button can be edited via the right side window.

   The Coupon block is ideal for special offers or discounts, with a text area and a dashed border (think a cut-out coupon) by default. On the right, you can add a background image or change the border.

   Similar to the Image block, the Logo block inserts an image but automatically loads your logo if you've set one up on the "List Settings" page.

   The Signature block adds your signature file to your message. Learn how to edit your signature file.

   The Divider block allows you to add visible separation elements to your message with borders that are dotted, dashed, or solid.

   The Share block allows your readers to share your broadcast to popular social networking sites like Facebook, Twitter, Google+, LinkedIn, and StumbleUpon. Add which services you would like to share to from the Share menu to the right.

   The Row block allows you to add new rows to the body of your message. Once you add a new row, you can add content blocks to that row such as images, text, headlines, etc. You can even add columns to add different blocks side by side.
5. Once you have arranged the message's layout, you can start entering the actual text. When editing the text of your message (remember, just click on the text you want to change), you'll notice a toolbar directly above the element.

   This toolbar will allow you to change the font, size, and another formatting of your text. Simply highlight the text you want to change, and use the toolbar to modify it.

   To create hyperlinks, highlight the text you'd like to turn into a link, then click the chain-link icon in the toolbar. In the popup that appears, simply enter the URL you would like to link to.

To use personalization fields, click the "Personalize" button. This will provide a drop-down menu where you can select which personalization value you would like to add to your message.

When you select Subscriber's Info, Subscriber's Location, or Your Snippets, the menu will expand with more options.

Note: You can click on the field to change the content at any time if you need to do so.

6. Once you've completed writing your message, click "Save & Exit" in the bottom right-hand corner of the editor.

   Note: The Drag & Drop Email Builder will automatically create a plain text version of your message for you.
7. Now that your message is saved, click the "Schedule" button to send or schedule your broadcast.
8. If you wanted to share your message, click the "Edit" button for the sharing section.

   On the "Broadcast Settings" page, you can choose whether or not to archive your message using the "Broadcast Archive" checkbox.

   Then, if sharing is on, you can choose to tweet the message or post the archive link to your Facebook page using the "Social Media" section.

   When you're satisfied, click "Apply."
9. Under the "Who should receive this message?" section, click "Edit" to select the segment of subscribers you would like to receive the message. Here, you can also include/exclude other lists.

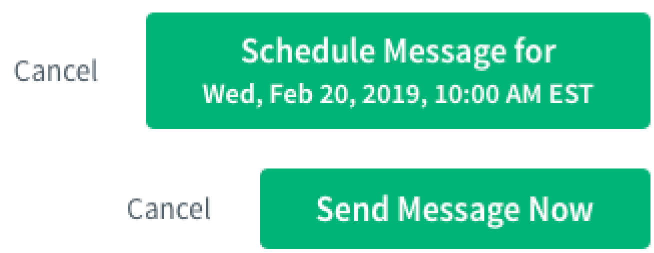
   When you're satisfied, click "Apply."
10. To schedule your message to go out on a future date, click "Edit" under the "When should this message be sent?" section.

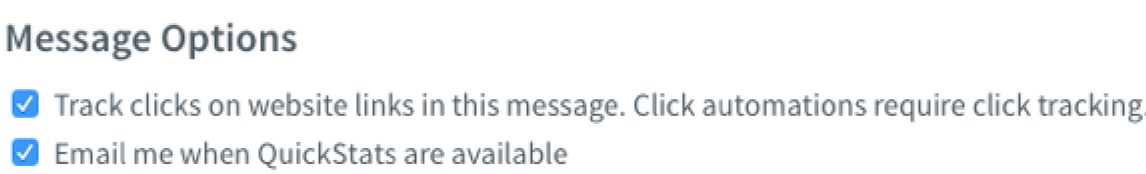
    Here, use the calendar to choose the date you'd like your message to be sent. You'll be able to see the dates, some notable holidays, as well as days you've scheduled other messages to be sent. Just click on the date that you would like the message scheduled for. From the left-hand sidebar, you can choose the time you would like the broadcast sent.

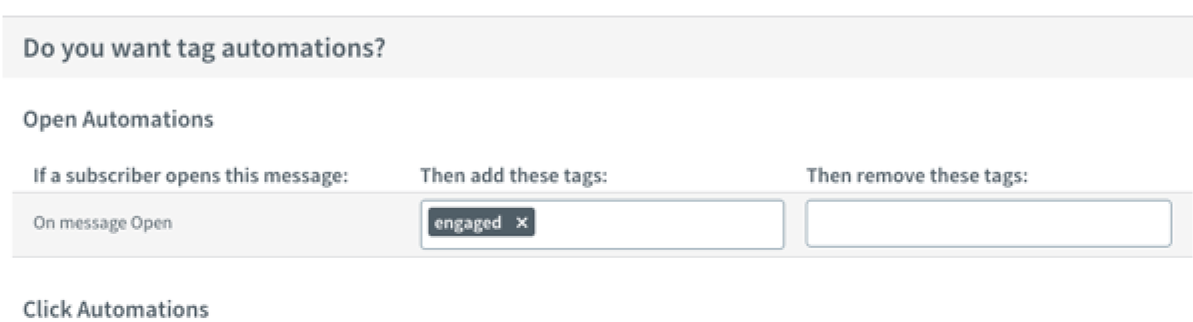
    Once you have a date, time, and time zone selected, click the "Apply" button to schedule the broadcast the be sent at that time.
11. To add open and click automation, click the blue "Edit" button under the "Do you want automation?" section.

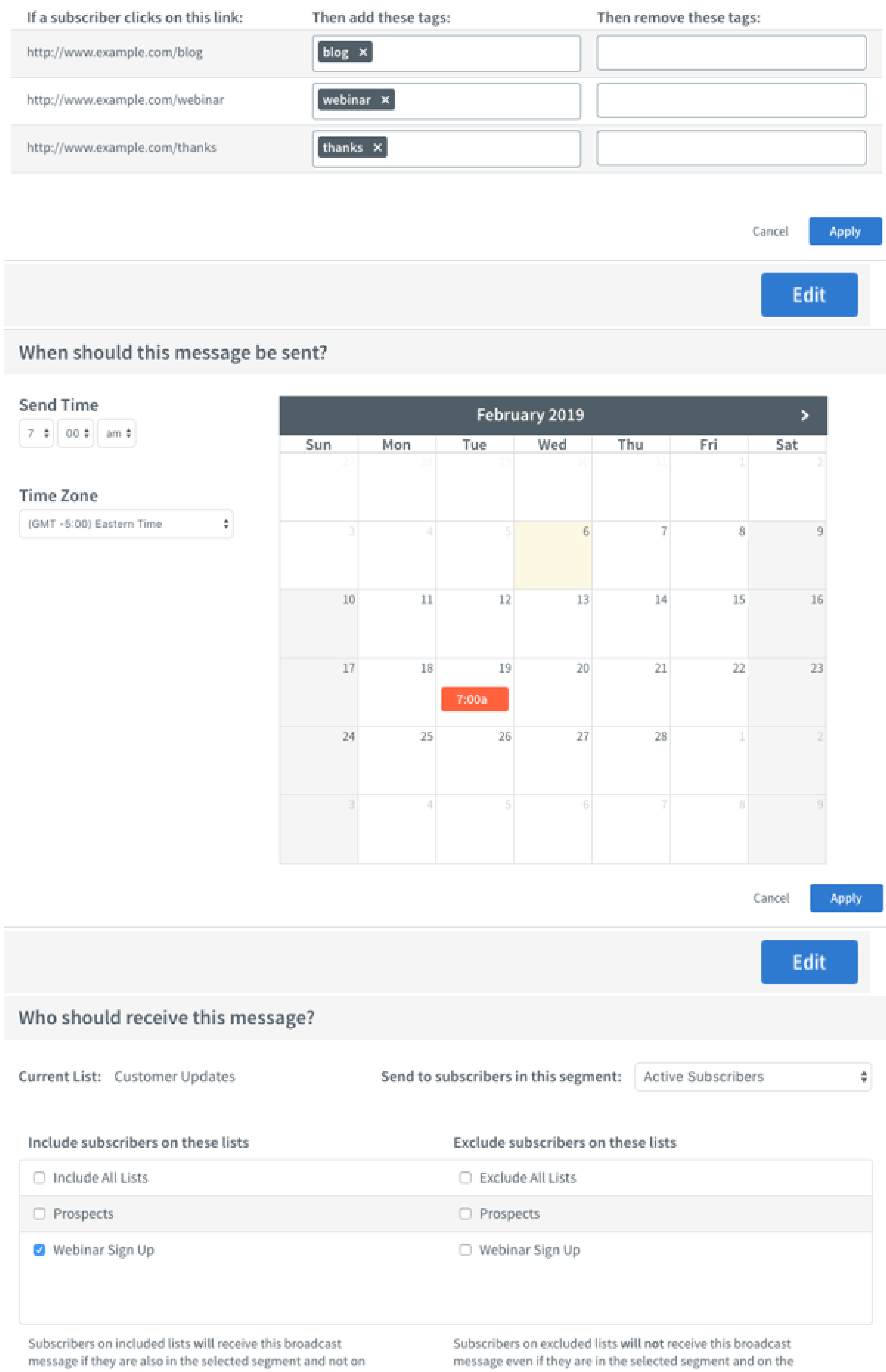
    In this section, you can choose to add or remove a tag if a subscriber has opened your message. Also, all of the links in your message will be displayed. You can choose to add or remove tags when a subscriber clicks on certain links in your message. Once you have applied or removed your tags, click the blue "Apply" button to save your changes.
12. Next, you can select whether or not you want to track when subscribers click the links in your message or are emailed when your QuickStats become available. Click the checkboxes to the left of either option.
13. When you're done, click the "Send Message Now" button for your message to be sent immediately.

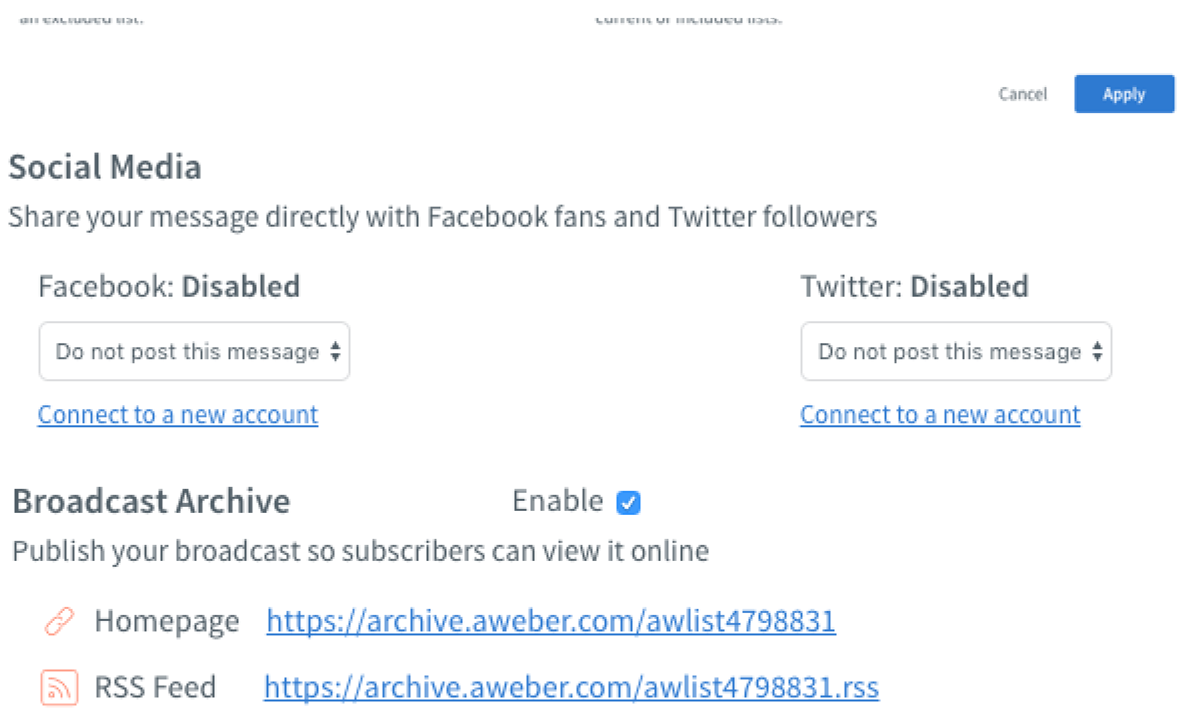
    If you've scheduled your message to be sent on a specific date, the button will be listed as "Schedule Message for [specific date you've selected]."

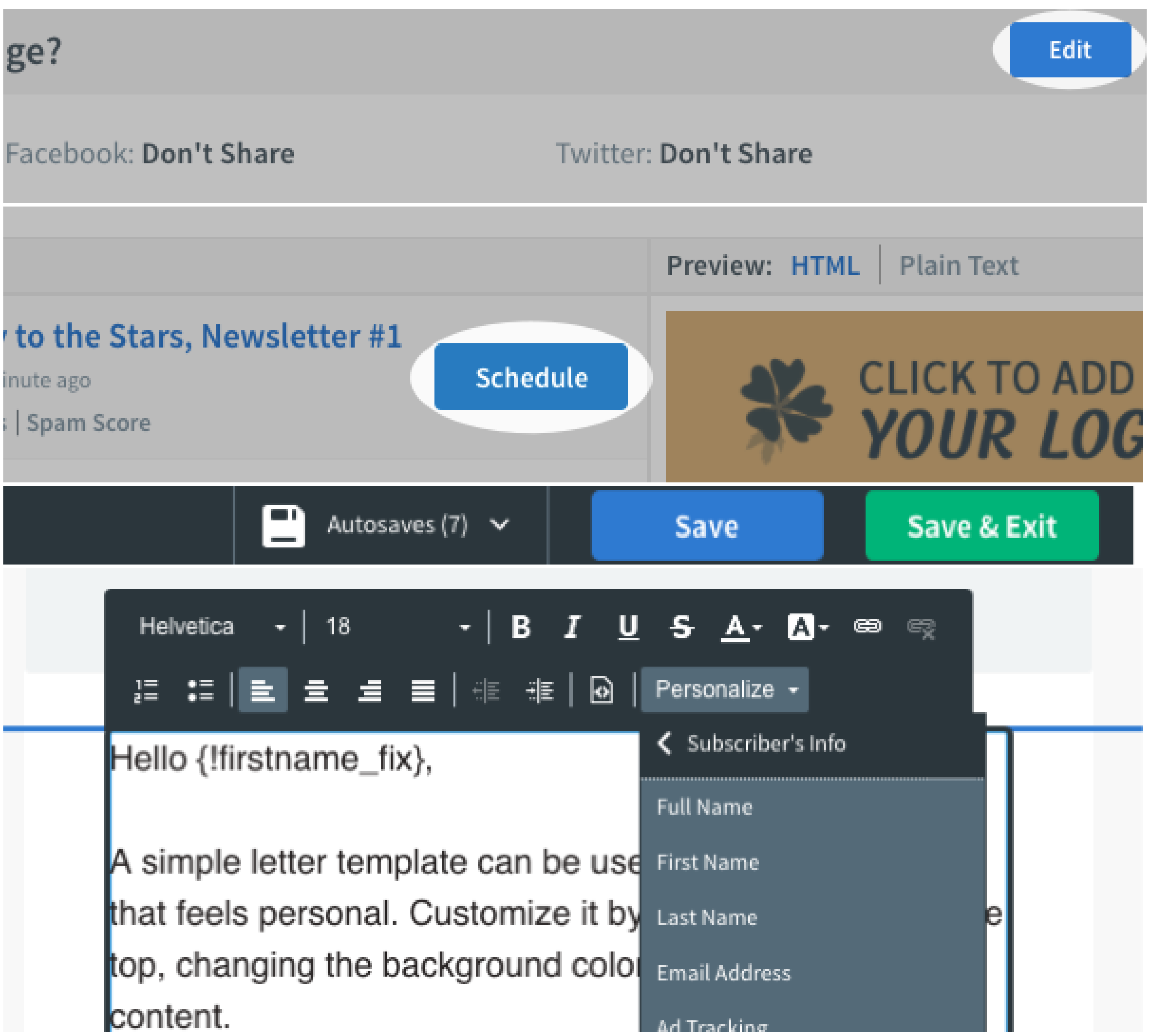


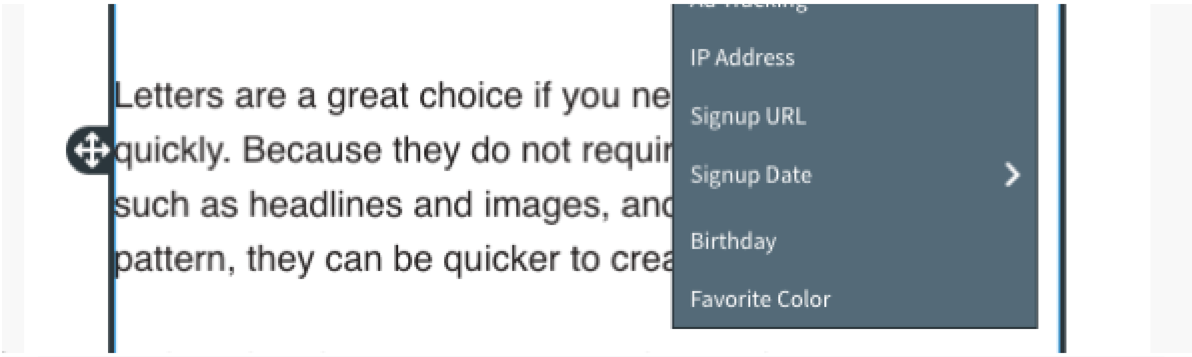


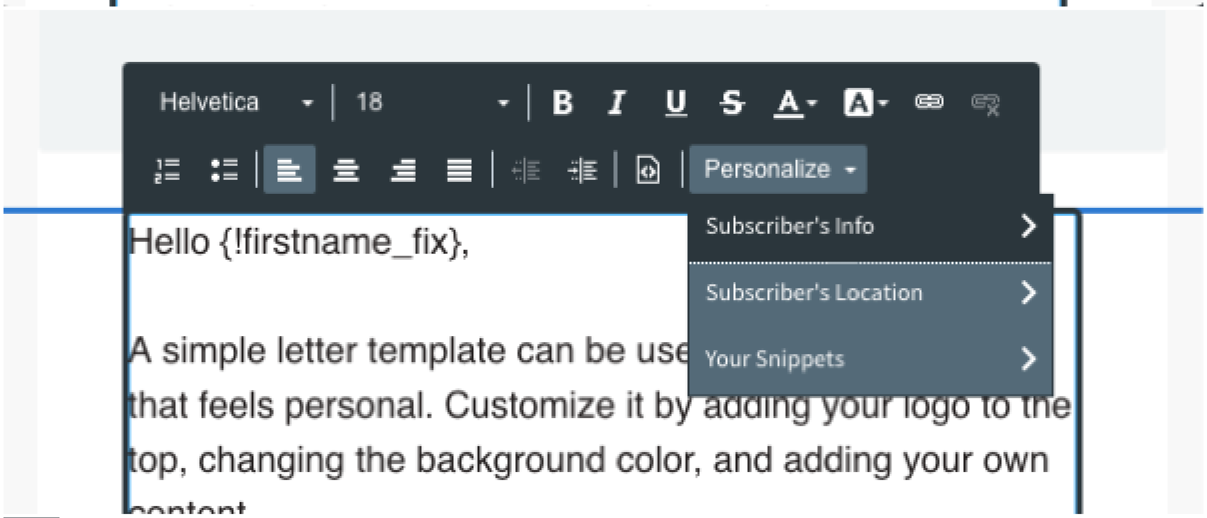


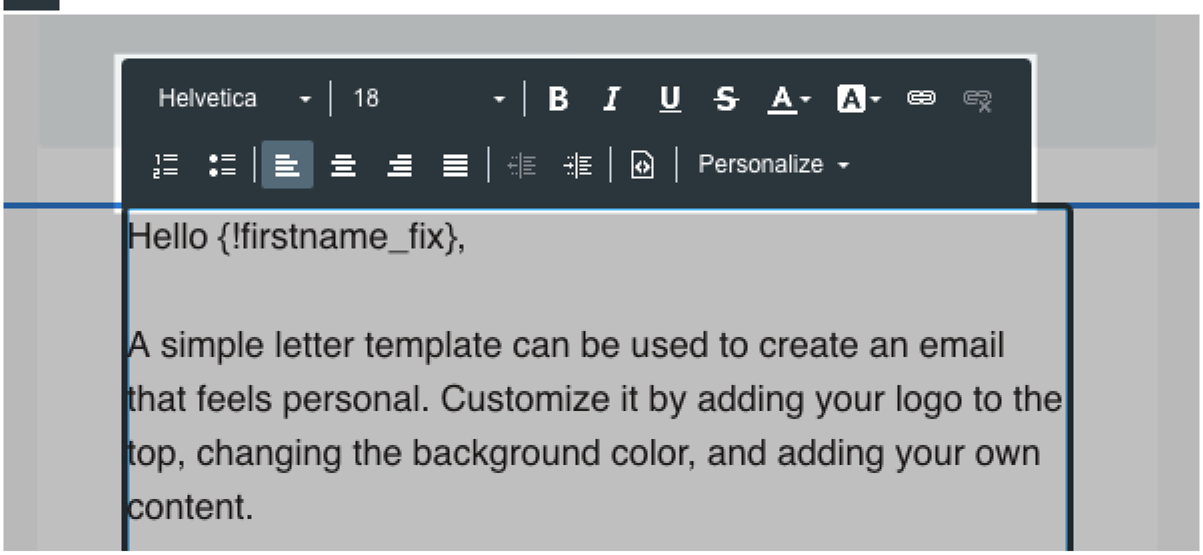

If a subscriber clicks on this link:
Then add these tags:
Then remove these tags:
http://www.example.com/blog
blog ×
http://www.example.com/webinar
webinar ×
http://www.example.com/thanks
thanks ×
Cancel
Apply
Edit
When should this message be sent?
Send Time
7
00
am
Time Zone
(GMT -5:00) Eastern Time
February 2019
Sun
Mon
Tue
Wed
Thu
Fri
Sat
7:00a
Cancel
Apply
Edit
Who should receive this message?
Current List: Customer Updates
Send to subscribers in this segment:
Active Subscribers
Include subscribers on these lists
Exclude subscribers on these lists
Include All Lists
Exclude All Lists
Prospects
Prospects
Webinar Sign Up
Webinar Sign Up
Subscribers on included lists will receive this broadcast message if they are also in the selected segment and not on
Subscribers on excluded lists will not receive this broadcast message even if they are in the selected segment and on the

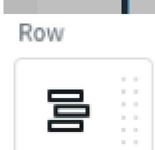

Cancel
Apply
Social Media
Share your message directly with Facebook fans and Twitter followers
Facebook: Disabled
Do not post this message
Connect to a new account
Twitter: Disabled
Do not post this message
Connect to a new account
Broadcast Archive
Enable
Publish your broadcast so subscribers can view it online
Homepage https://archive.aweber.com/awlist4798831
RSS Feed https://archive.aweber.com/awlist4798831.rss


ge?
Edit
Facebook: Don't Share
Twitter: Don't Share
Preview: HTML
Plain Text
to the Stars, Newsletter #1
Schedule
inute ago
Spam Score
CLICK TO ADD
YOUR LOG
Autosaves (7)
Save
Save & Exit
Helvetica
18
Personalize
Subscriber's Info
Full Name
First Name
Last Name
Email Address
Hello {!firstname_fix},
A simple letter template can be use
that feels personal. Customize it by
top, changing the background colo
content.


Letters are a great choice if you ne
quickly. Because they do not requir
such as headlines and images, and
pattern, they can be quicker to crea
IP Address
Signup URL
Signup Date
Birthday
Favorite Color


Helvetica
18
Personalize
Hello {!firstname_fix},
A simple letter template can be use
that feels personal. Customize it by adding your logo to the
top, changing the background color, and adding your own
Subscriber's Info
Subscriber's Location
Your Snippets


Helvetica
18
Personalize
Hello {!firstname_fix},
A simple letter template can be used to create an email
that feels personal. Customize it by adding your logo to the
top, changing the background color, and adding your own
content.


Row


Share


Divider
Signature
Logo
Coupon
Product
Follow Me

Button

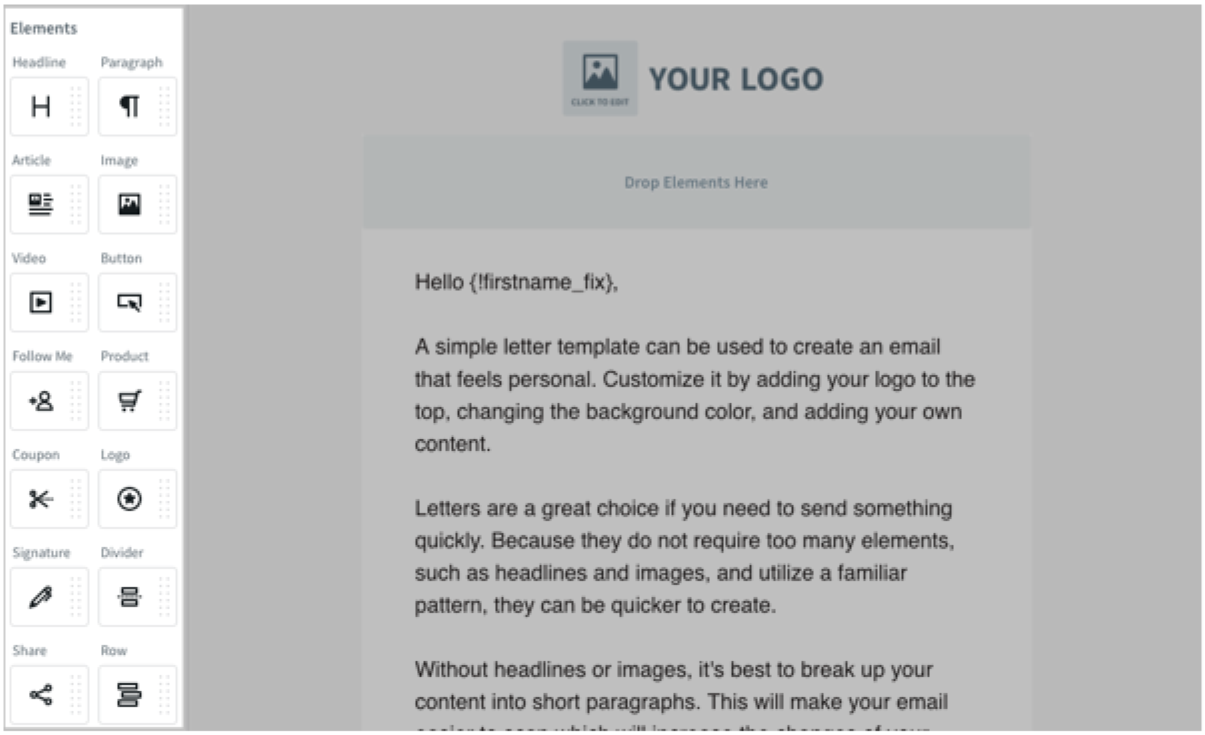
Video

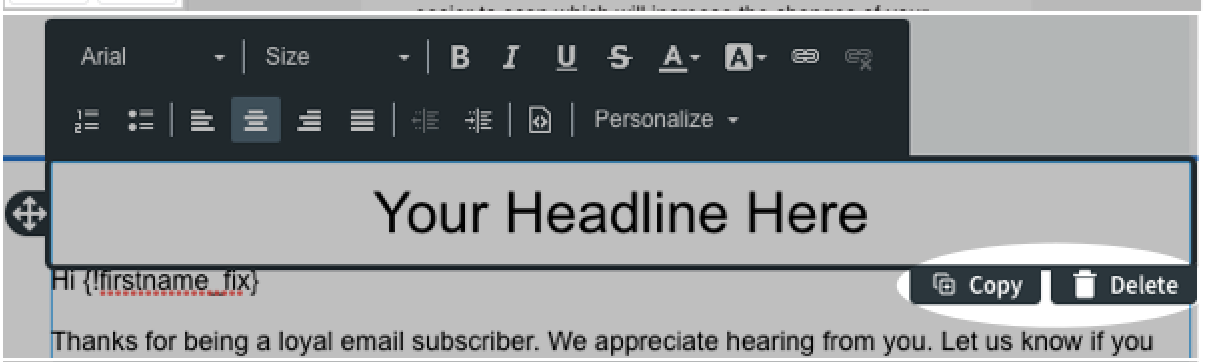
Image

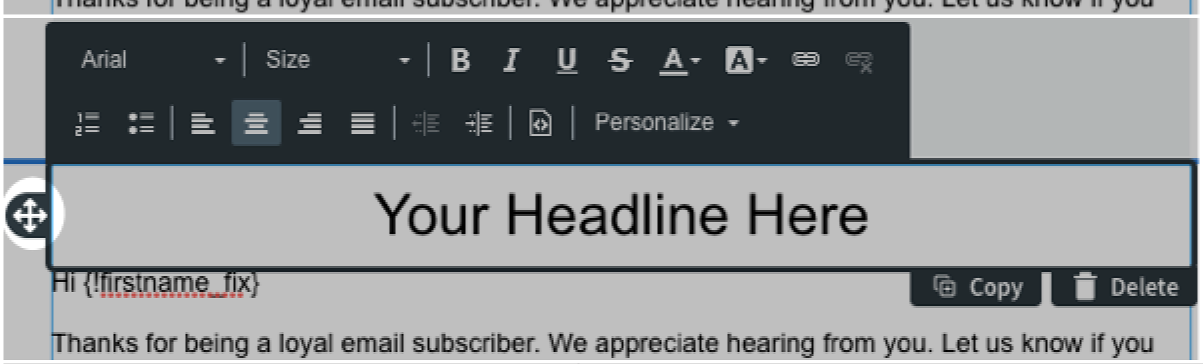
Article

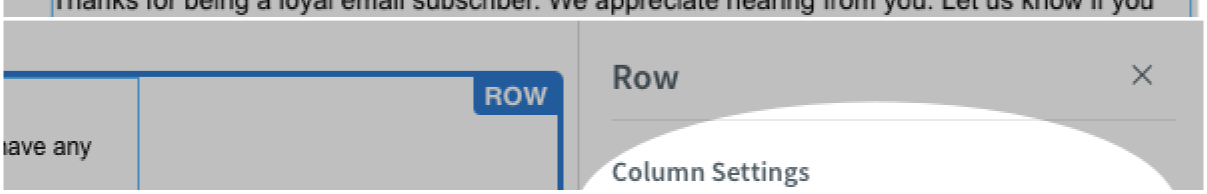
Paragraph

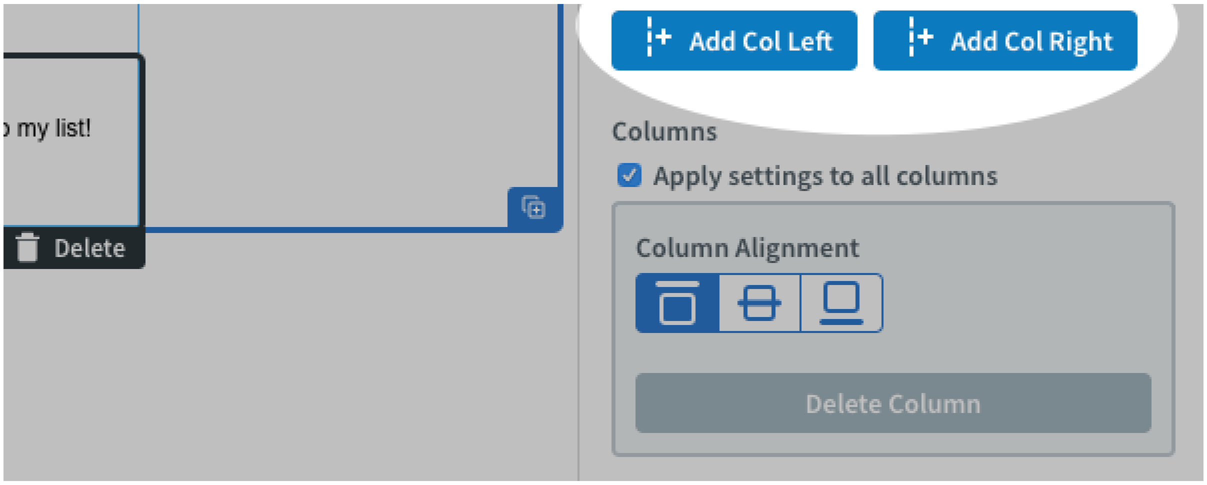
Headline

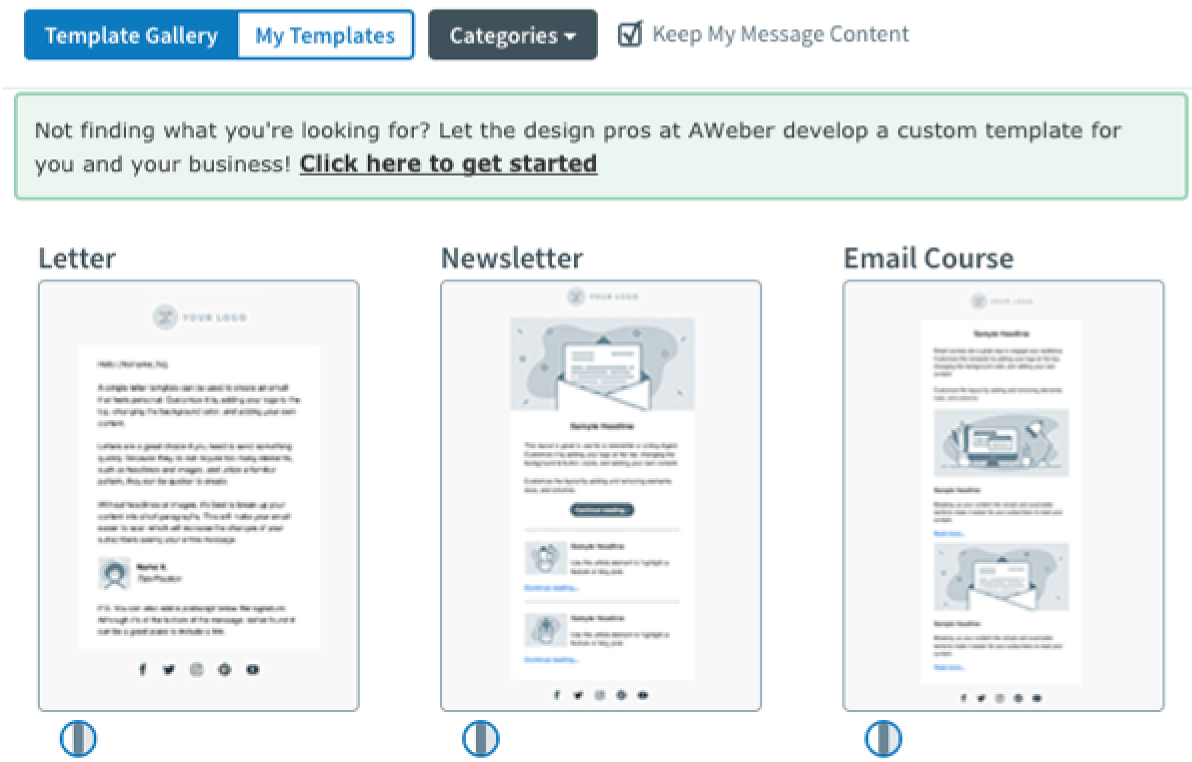

Elements
Headline
Paragraph
Article
Image
Video
Button
Follow Me
Product
Coupon
Logo
Signature
Divider
Share
Row
YOUR LOGO
Drop Elements Here
Hello {!firstname_fix},
A simple letter template can be used to create an email that feels personal. Customize it by adding your logo to the top, changing the background color, and adding your own content.
Letters are a great choice if you need to send something quickly. Because they do not require too many elements, such as headlines and images, and utilize a familiar pattern, they can be quicker to create.
Without headlines or images, it's best to break up your content into short paragraphs. This will make your email

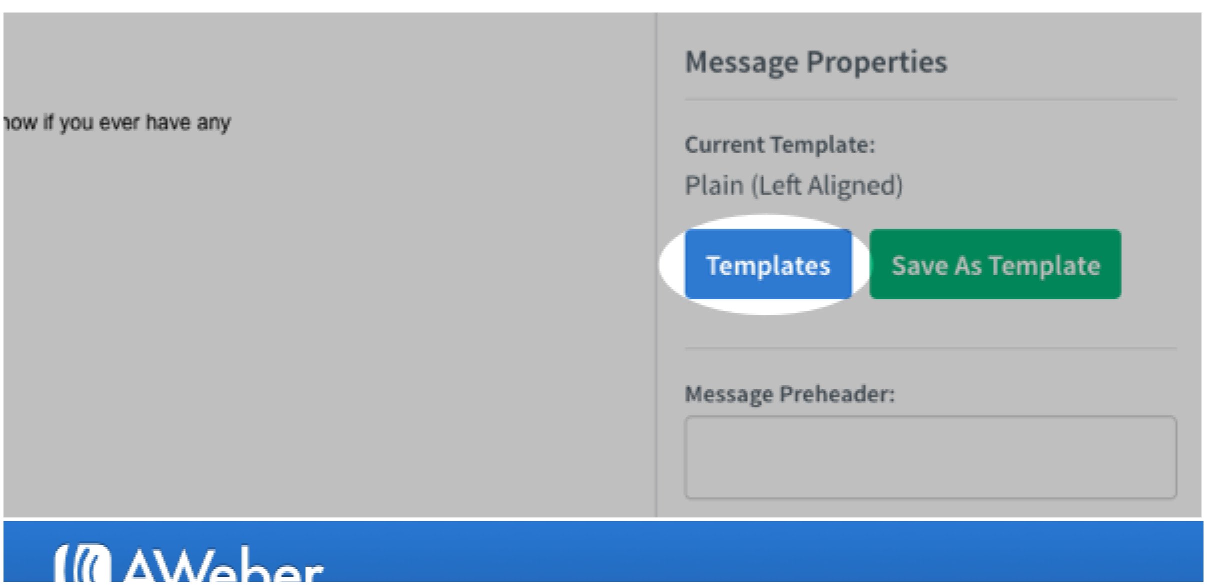

Arial
Size
Personalize
Your Headline Here
Hi {!firstname_fix}
Copy
Delete
Thanks for being a loyal email subscriber. We appreciate hearing from you. Let us know if you

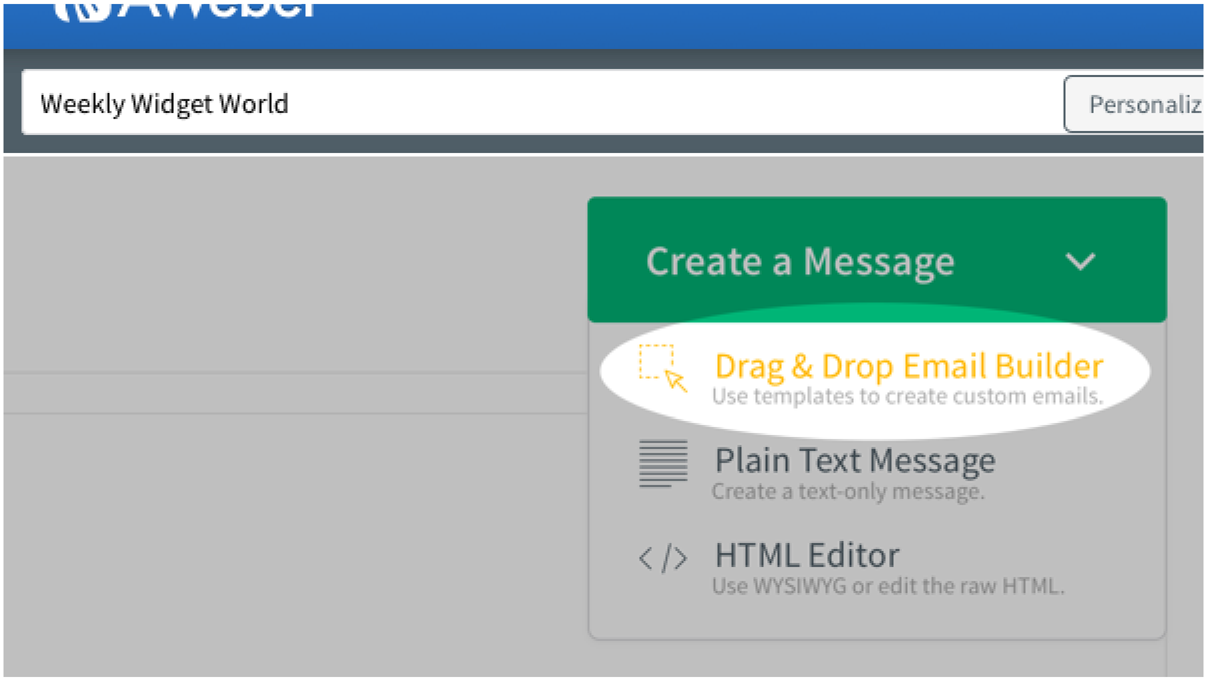

Arial
Size
Personalize
Your Headline Here
Hi {!firstname_fix}
Copy
Delete
Thanks for being a loyal email subscriber. We appreciate hearing from you. Let us know if you

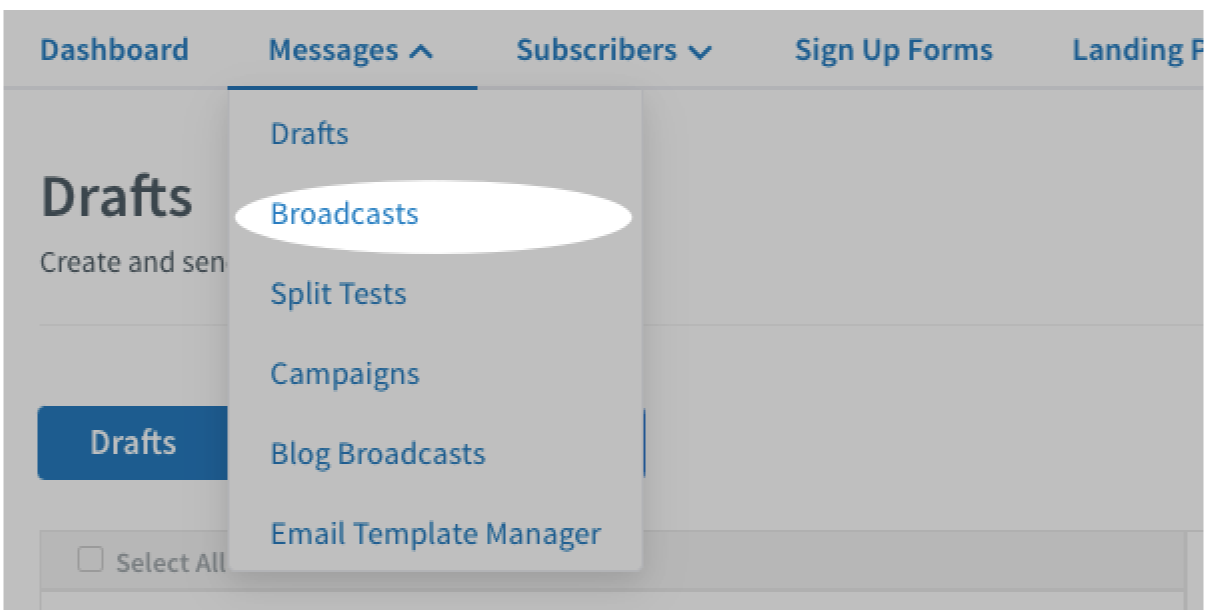

ROW
Row
Column Settings


Add Col Left
Add Col Right
my list!
Delete
Columns
Apply settings to all columns
Column Alignment
Delete Column


Template Gallery
My Templates
Categories
Keep My Message Content
Not finding what you're looking for? Let the design pros at AWeber develop a custom template for you and your business! Click here to get started
Letter
Newsletter
Email Course


ow if you ever have any
Message Properties
Current Template:
Plain (Left Aligned)
Templates
Save As Template
Message Preheader:





Migrating Subscribers Between Lists

*Placeholder

Billing.DANG-designs

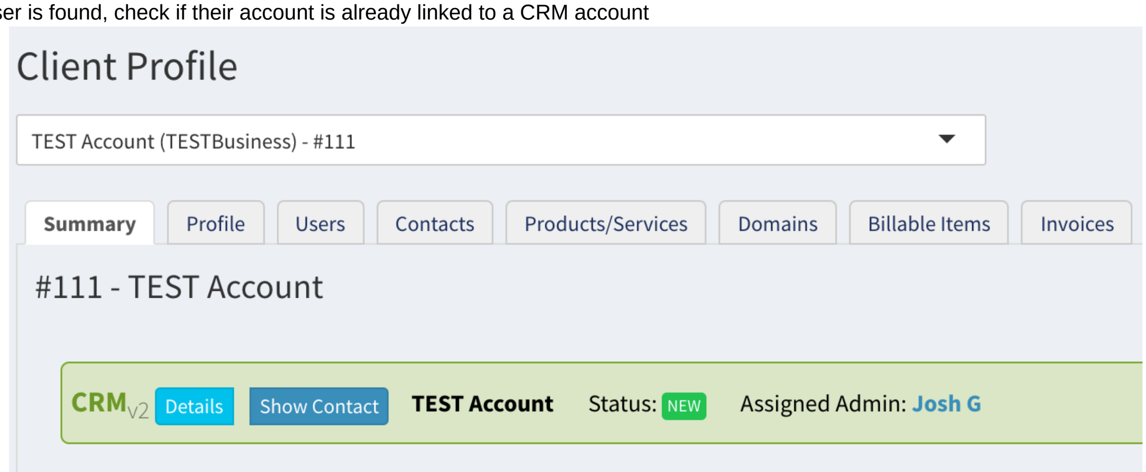
This page will be the parent for future navigational guides pertaining to billing.dang-designs.com. This site uses WHMCS and other documents pertaining to alternative pieces throughout will be located in logical destinations.

- How To Add New Leads & Discovery Call Appointments
- How To Customize Billing.Dang-designs

- How To Enter New Call/Meeting Event In Google Calendar/CRM
- How To Perform CSS Updates On WHMCS
- How To Update Payment Information For Clients

How To Add New Leads & Discovery Call Appointments

This page will document the steps to take with a new lead and discovery call appointment. This guide operates under the assumption that the lead details have been gathered and the user is not in the WHMCS system already.

1. **Use the View/Search Clients option in WHMCS to check for the user**
    a. Be thorough. If the user is not found, continue to the next steps
    b. If the user is found, check if their account is already linked to a CRM account
        i. 

        ii. Would look like this ^^ if already linked!
    c. *If their account IS linked already to a WHMCS account, then the user is either not the same user or not a new lead.*
2. **Click on Addons > CRMv2**
3. **On the top-left-hand side select 'Create Contact'**
    a. *We don't assign the client to a WHMCS client since there should be a contract sent out first before they have an account created.*
    b. Set the assigned admin
        i. Either yourself or the person taking ownership of the client
    c. Leave the 'Assigned Client' field blank
        i. Remember, this is for WHMCS users to link up with their WHMCS account
    d. For 'Contact Type' choose either Prospect DANG or Prospect FMR/DANG
        i. Prospect DANG - users who are going to be clients for dang-designs
        ii. Prospect FMR/DANG - users who are going to be clients for DANG advertising
    e. Enter the first name, last name, and email fields
        i. Be sure to add the email address; if there is not an email address then we need to attempt to capture that before entering them into the system for the following steps.
    f. Leave the status as 'New' and set the country
    g. The other fields may not be essential depending on the business but can be added as needed
    h. Press 'Add' at the bottom of the screen.
4. **Create the event for the discovery call/meeting**
    a. **(inside of CRM)**
        i. Set the meeting type
            1. call or meeting
        ii. Set the date and time
            1. using the calendar option or manual entry in the 'Date' field
        iii. Enter the description for the follow up
            1. This field should include Google Meet information or any other pertinent details regarding the dicovery call /meeting.
        iv. Select 'Add Follow-up' button midway through the page
    b. **(inside of Google Calendar)**
        i. Follow the steps outlined in the confluence linked below
            1. How To Enter New Call/Meeting Event In Google Calendar/CRM
        ii. Basically, ensure that the fields below have the proper information so they can be carried over to the CRM table properly
            1. New Google Calendar item is an Event
            2. The date is set for a future date
            3. The time is set
            4. The customer's email is placed in the description field
5. **You're Done!**

How To Customize Billing.Dang-designs

This document will include details essential to customizing the domain for WHMCS to be precisely the way that we want it. There is one specifically notable detail that has already been covered in another document that will be linked from this page instead. There will

be additional changes made throughout the timeline of this application's life, so future custom changes will be added to this destination.

To begin, the login page needs to force everyone to login if they are a guest on the site (not logged in already). The hook to set this up has been created and clarified for its respective path at the link below.

Billing.Dang-Designs ForceEveryoneToLogin

This hook at the link below ensures that the link, when opened, uses a _self link instead of blank. Thus, the page will open in the same tab instead of opening into a separate window.

Billing.dang-designs _Self Opening Link Hook

CRM Specifics

To correct the CRM to show the proper view for any user

To correct the CRM to show the proper view for any user, a simple adjustment was made in the following pathway. If adjustments are needed to be made, use the Control + F function to search for 'Lara'.

public_html/modules/addons/mgCRM2/templates/ModulesGarden/assets/css/custom.css

```
div.mg-wrapper.body {
  color: #333333;
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-size: 13px;
  line-height: 1.5;
  font-style: normal;
  font-weight: normal;
  padding-left: 240px; /* UPDATED TO FIX VIEW IN LARA THEME*/
  }
```

To correct the CRM to show contacts page (remove SQL error)

1. Navigate to the WHM terminal section.
2. type the following command to get to the root directory
   a. `cd ../`
3. Type the following command to enter the etc directory
   a. `cd etc`
      i. Should you become lost along this path and need to know what files are present, use the following.
      ii. `ls`
4. Type the following command to open the my.cnf file and make edits.
   a. `nano my.cnf`
5. Under [mysqld], add the following line

a. `sql_mode=""`

6. Save the changes and ensure that the file is saved with the same filename

To correct the CRM links opening in separate tabs within the application (NOT WHMCS)

1. Open the File Manager in Cpanel
2. Navigate to public_html > modules > addons > mgCRM2 > templates > ModulesGarden > app > contacts > details > orders > templates > orders.html
3. Use the CTRL + F to search for the following
   a. `_blank`
   b. Change to:

```
_self
```

4. Save the changes.

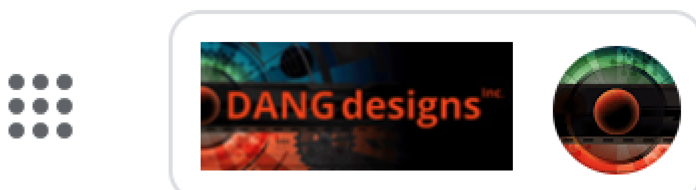
When Switching WHMCS To A New Domain For Testing

1. **Make sure to update the configuration file with the secondary license information**
2. **Change the references for the database to point to the new domain**
3. **Remove the recaptcha key and secret from the database**
   a. Otherwise, the page continuously loads the login page and never moves.
      i. This happens because the JavaScript promise is not being sent over properly and can't connect with its initial domain.

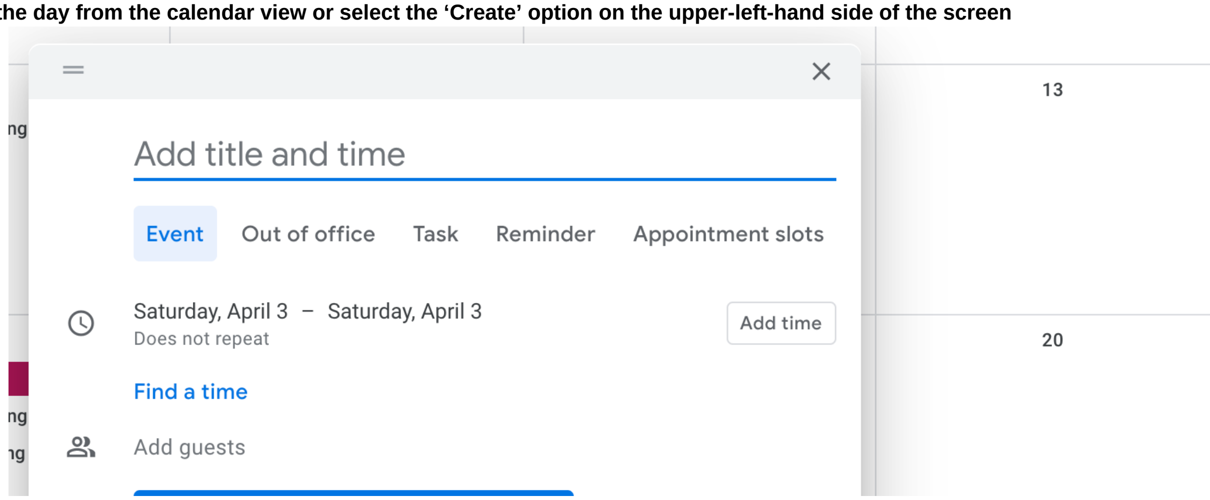
How To Enter New Call/Meeting Event In Google Calendar/CRM

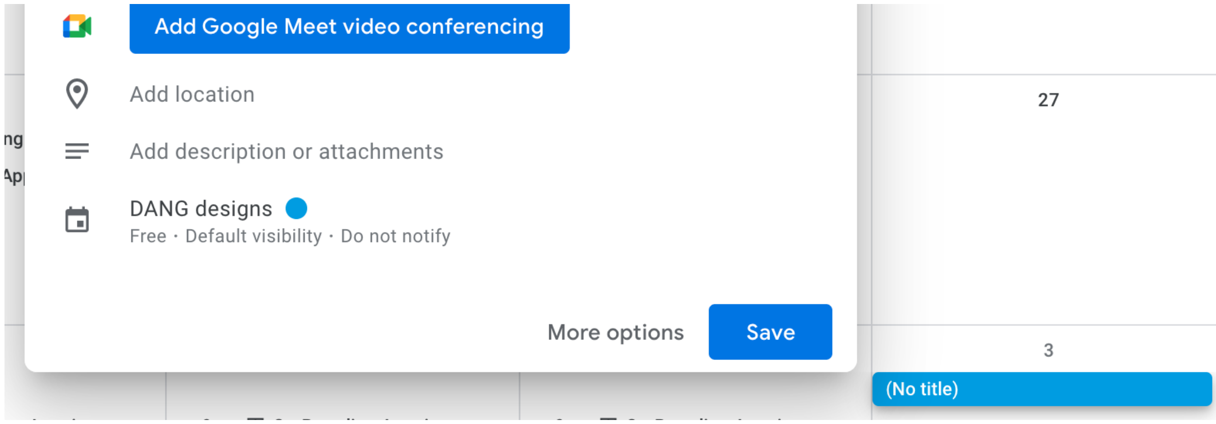
This page will walk a user through the process to add a new **event**, emphasis on the event title, to the Google Calendar of the support@dang-designs.com account so that it is transferred over to the CRM tables for billing.dang-designs. Details that are crucial will be underlined for emphasis as well to ensure that no issues can occur.

1. **Sign into the Google account for support@dang-designs.com**
2. **Navigate to the Google Calendar for the user**
   a. If in gmail, the Calendar can be found in the 9-circle menu next to the account option
      i.

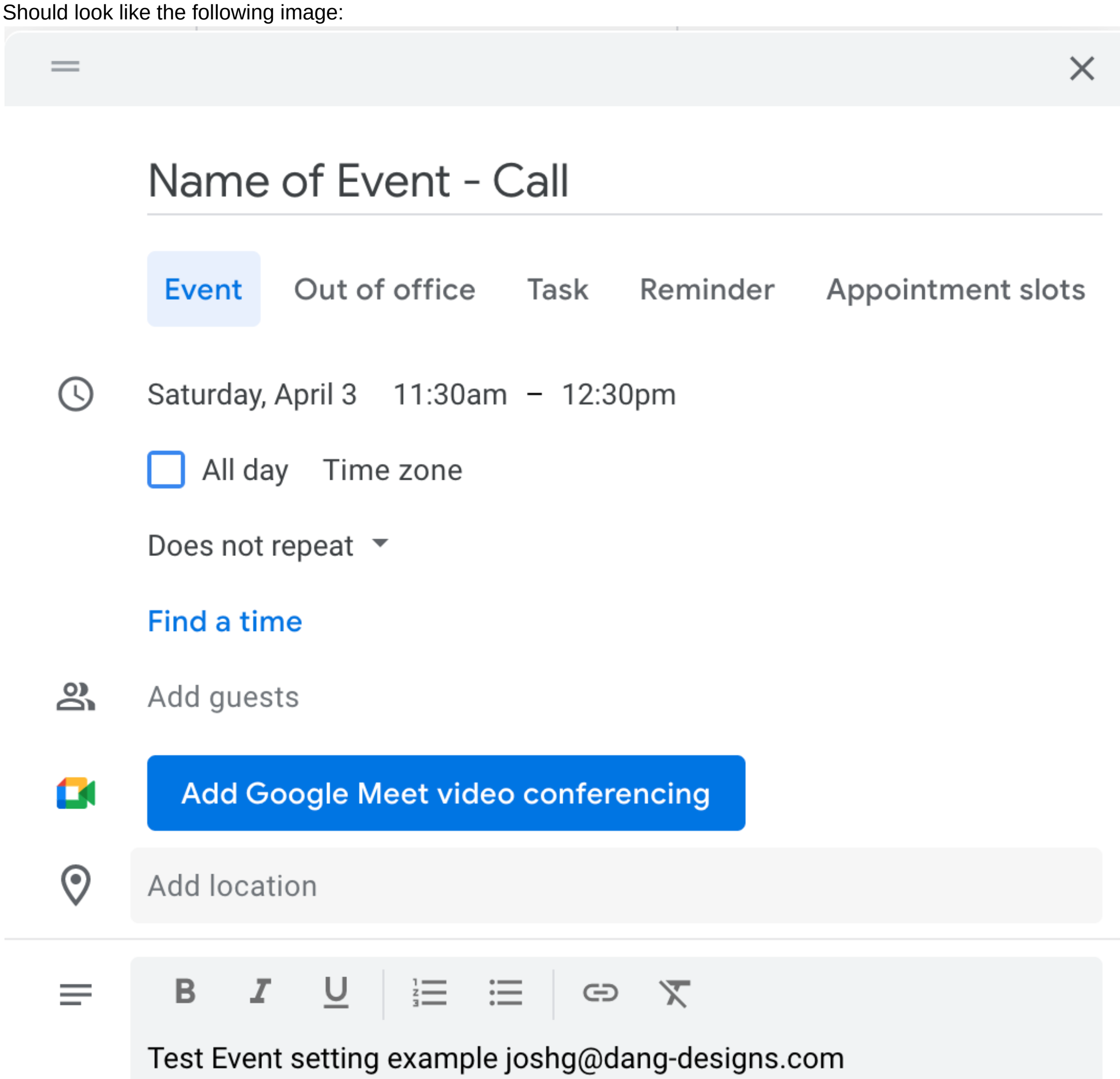


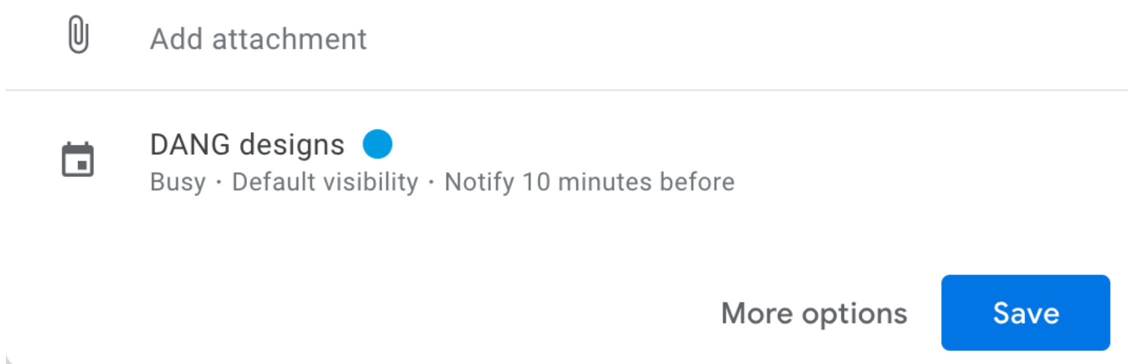
3. **Select the day from the calendar view or select the 'Create' option on the upper-left-hand side of the screen**
   a.



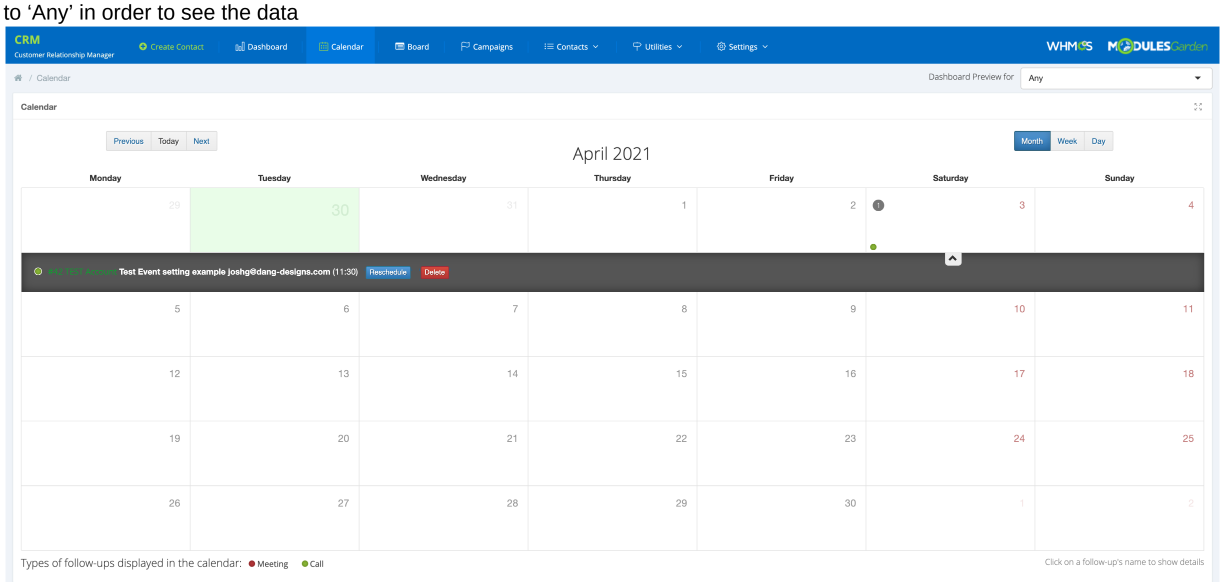


   b. Ensure that the blue square is selected on the 'Event' option
   c. Also ensure that the date for the event being added is in the future
      i. The items being added are based on future events, not dates that have already passed
4. **Create the event for the meeting or call as follows**
   a. Provide the name of the event with the use of the word 'call' or 'meeting'
      i. **Be sure to ONLY have one or the other; default is 'call' if it is not added**
   b. Set the time that the event should be
      i. Do not leave this item as an all-day event as it may not show up correctly in the CRM system
   c. Set the description field
      i. **Be brief while also including the email address of the customer that will be contacted**
      ii. Ensure that the customer's email is listed in CRM
         1. **If the customer is not yet in CRM, add them through the Clients > View Clients area of billing.dang-designs**
      iii. Should look like the following image:
      iv.





   d. Press ‘Save’

5. **Check the CRM calendar after waiting about 6 - 8 minutes (cron set for 5 minutes) to ensure set**
   a. *Be aware that the user that is entering the information is the AUTOMATION account and your preview will need to be updated to ‘Any’ in order to see the data
   b. 

6. **You’re Done!**

How To Perform CSS Updates On WHMCS

This page is used to provide clarity on the ways to make changes on WHMCS for the user/client and for the admin. It’s important to note that these are two separate sections that located in two separate locations entirely.

- **For changes to the main template where clients would be seeing them:**
  - /home/dangbills/public_html/templates/control/css/custom.css
- **For changes to the admin template where administrators would be seeing them:**
  - /home/dangbills/public_html/taygray/templates/lara/custom/custom.css

How To Update Payment Information For Clients

This page will document the steps required to update the payment information for clients to ensure that they are able to pay for the services and products that they have from DANG designs.

1. **Obtain the new payment method (ACH, Credit Card, etc.) information from the client**
   a. This one can be done in real-time as needed, but step 1 will presume that you have the user’s information to update.
   b. Always make sure to confirm the billing address is still updated and accurate within our records to reduce possibility for errors.
2. **Sign into the WHMCS site as a support or admin user**
   a. https://billing.dang-designs.com/taygray
3. **Navigate to Clients > View/Search Clients and find the client to update the information for**
   a. You can search by client/company name, email address, or phone number
   b. Select the user’s account by selecting the ID, First Name, or Last Name
      i. If there is only one user returned by the criteria used to search, WHMCS will automatically send you into the client’s account.
4. **(If expiry date is only piece needed to be changed) Under ‘Pay Methods’, select the card needing to be updated**
   a. A popup box will appear that allows you to update the expiration date; all other changes will need to follow step 5 instead.
   b. Press save after updating
5. **(If new credit card needs to be added). Under ‘Clients Information', select ‘Login as Owner’**
   a. The screen will redirect to the home page that the user sees. Navigate to Billing > Payment Methods
   b. Select ‘Add New Credit Card’ button

        i. Enter the description (Visa/Mastercard/etc)
        ii. Enter the card number
        iii. Enter the Expiry Date
        iv. Enter the CVV
        v. Press ‘Save Changes’
6. **You’re Done!**

Capitalcityshoes

This page will host all items segmented for capitalcityshoes.com (live or sandbox) and display their children items for easier navigation.

- Capitalcityshoes Custom Pieces, Modules, & Extensions
- How To Add A New Sub Store/View
- How To Add New Punchout Catalog
- How To Set Up SMTP On Magento 2 For SendGrid
- How To Update Punchout Site URL On Ariba
- How To Update Sandbox For Capitalcityshoes With Live Files
- Payment Restrictions Per Website
- How To Add Extension On Magento 2
- How To Import New Users
- How To Correct User To Be Charged Taxes (Customer Groups)
- How To Update User Passwords & Advise Them
- RICS Access

Capitalcityshoes Custom Pieces, Modules, & Extensions

This page will log all of the customization pieces as well as the extensions and their respective functionality. When making the transition from Magento 1 to Magento 2, we need to capture every instance that could cause issues down the line.

Customizations

1. Landing page for B2B, GHS, FLPV
2. Index page for B2B, GHS, FLPV

Modules

These will be listed by company, then by enabled module, then detail whether the module is needed in the transition.

1. **Inchoo**
    a. Inchoo_LoginAsCustomer
        i. This extension allows the login as a user function which is already a feature offered by Magento
2. **Amasty**
    a. Amasty_Audit
    b. Amasty_Base
    c. Amasty_Brands
    d. Amasty_Catcopy
    e. Amasty_Commonrules
    f. Amasty_Conf
    g. Amasty_Customerattr
    h. Amasty_Geoip
    i. Amasty_Paction
    j. Amasty_Payrestriction
    k. Amasty_regions
    l. Amasty_Rolepermissions
    m. Amasty_Smtp
        i. https://amasty.com/smtp-email-settings-for-magento-2.html
            1. Alternative provided by same source
    n. Amasty_StoreCredit
        i. https://amasty.com/store-credit-and-refund-for-magento-2.html
            1. Alternative provided by same source
    o. Aoe_Scheduler
3. **CM?**
    a. Cm_RedisSession
    b. Cminds_Core
    c. Cminds_Coupon
    d. cmsmart_Megamenu
        i. https://cmsmart.net/magento-extensions/magento-mega-menu-extension
            1. Alternative provided by same source
4. **Itoris**
    a. Itoris_Installer & Itoris_StoreLoginControl
        i. https://www.itoris.com/magento-2-store-login-access-permissions.html
            1. Alternative provided by same source
5. **MageWorx**

a. MageWorx_CustomOptions & MageWorx_All
    i. https://www.mageworx.com/magento-2-advanced-product-options-suite.html
        1. Alternative provided by the same source

How To Add A New Sub Store/View

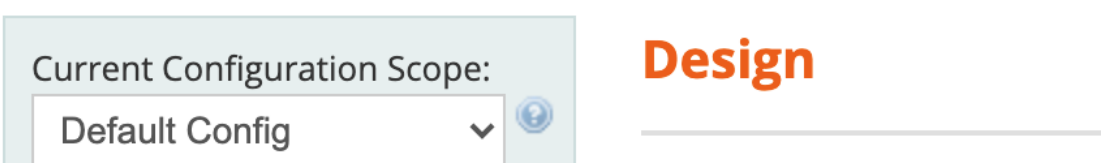
This page will document the steps needed to add a new store/view to a website on Capitalcityshoes.com. This example listed below will document the steps needed to create a new store for the B2B location specifically for a GHS themed store. Updates will be made to this document as processes or procedures are updated as well.

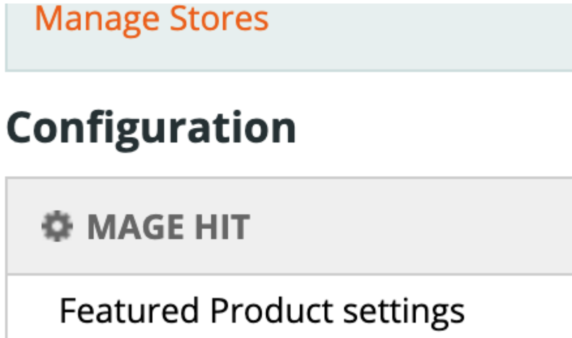
Always Perform Any New Operations On The Sandbox Site First

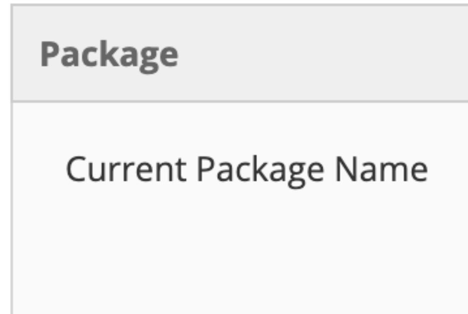
1. **Navigate to the backend of the Magento site for Capitalcityshoes and sign in**
    a. Link should be something like the following:
    b. https://shop.capitalcityshoes.com/index.php/dep2820
2. **Navigate to System > Manage Stores**
3. **Select the blue button in the right-hand side that says 'Create Store'**
    a. Select the website that the store will belong to
    b. Provide an appropriate name for the new store
    c. Select the root category that most appropriate fits the new store
        i. This can be easily updated at a later time if needed.
    d. Press 'Save Store'
4. **Select the blue button int he right-hand side that says 'Create Store View'**
    a. Select the store that was previously created
    b. Provide an appropriate name for the store view
    c. Use the same conventions for the code that are set for other stores
        i. i. e. b2b_storename
    d. Select the status as 'Enabled'
    e. Set 'Sort Order' as 0
    f. Press 'Save Store View'
5. **Add the store view logo name to the database**
    a. Open up the database for the Magento site and find the table called 'core_store'
        i. Should be on page 7
    b. Expand the list to find the new store added
    c. Double-click into the logo field and update the string with the name of the logo that will be added
        i. i.e. moorebass.png
    d. Press enter
6. **Add the store view logo to the files**
    a. Open up the file manager for the Magento site and navigate to media/images/websites
    b. Upload the image of the new store
        i. Double-check the spelling of the image to ensure that it matches the logo name added to the database; if the name isn't 100% consistent, the image will not display.
7. **Navigate to CMS > Pages to add a home page (or add it to the list of homepages)**
    a. Find the home page section for the store view to belong
    b. For B2B, it should look something like this

B2B Homepage	home	2 columns with right bar	b2b.capitalcityshoes.com Bureau of Prisions Bureau of Prisions Dept. Management Services - Safety Footwear Dept. Management Services - Safety Footwear Dept. of Environmental Protection - Apparel & Footwear Dept. of Environmental Protection - Apparel & Footwear FDOT Turnpike Enterprises FDOT Turnpike Enterprises Florida Dept of Agriculture Florida Dept of Agriculture Florida Lottery Division of Security Florida Lottery Division of Security Grady Health System

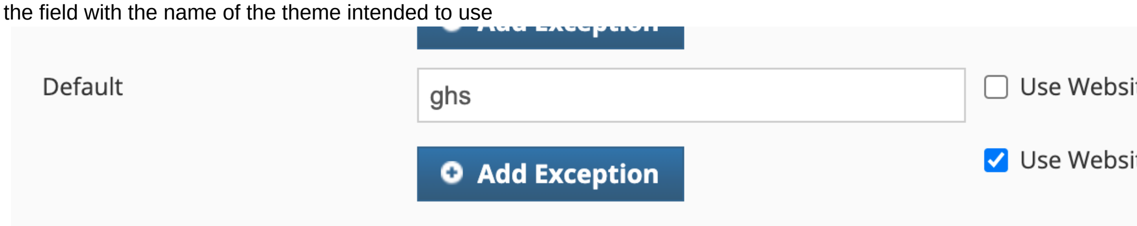
    c. Click into the Homepage and scroll down within the store view to find the new store view added
    d. Hold down 'CMD' (Mac) or 'CTRL' (Windows) and click on the store to add
        i. This should highlight the field like the others in the list.
        ii. Once highlighted, click away from the list
    e. Press 'Save Page'
8. **Update the theme for the store view scope** (Required if GHS clothing allowance needed for B2B store)
    a. Navigate to System > Configuration > Design
    b. Adjust the Current Configuration Scope to the new store defined
        i.







  c. De-select the checkbox for the default Default (under Themes)
  d. Update the field with the name of the theme intended to use
    i.



  e. Press 'Save Config'

9. **Update the logo for transactional emails**
    a. On the same location, scroll down to 'Transactional Emails'
    b. De-select the 'Use Website' option for the Logo Image field and choose the file of the image
    c. Depending on the image used, update the width and height dimensions.
    d. Press 'Save Config'
10. **You're Done!**

How To Add New Punchout Catalog

This page will serve to show a user how to add a punchout catalog on the ariba network. The location for the ariba network is found at https://ariba.com/login with the selection of the 'Suppliers' option for the username and password. Once there, follow the following steps to add a punchout catalog.

1. **On the navigation bar at the top of the screen, select 'Catalogs'**
2. **Select 'Create Punchout Only'**
3. **Enter the details for the punchout catalog**
    a. Catalog Name should be specific to the punchout site being created
    b. Punchout URL should be the one provided by Avetti
        i. i. e. https://punchoutgateway.avetti.ca/preview/punchout.xml?vid=20150913312
    c. UN/SPSC code is the same as the one used for the other catalog
        i. Can be found be selecting the other catalog, going to content, 6th row from the left.
    d. Brief description can be a general description of what the catalog is for
4. **Press 'OK'**
5. **You're Done!**

How To Set Up SMTP On Magento 2 For SendGrid

This document will cover the steps to set up SMTP on Magento 2 with SendGrid as the third party option. Since a large number of the steps match with Magento 1 and Magento 2, similarities will be pointed out throughout the documentation for backwards installations as well.

1. **Download the Mageplaza SMTP (free) extension for Magento 2**
    a. Create an account on Mageplaza
    b. Select the free SMTP extension and proceed to checkout
        i. You will receive an email to the email address provided that directs you to the license page to get a copy of the license key
    c. Download SMTP extension using composer
        i. Use the terminal to move to the Magento 2 installation public_html folder
        ii. Sign in as the site user (i.e. shopccshoes)

```
1. su shopccshoes
```

        iii. Perform the following commands:

1.
```
sudo composer require mageplaza/module-smtp
php bin/magento setup:upgrade
php bin/magento setup:static-content:deploy -f
```

   d. Your SMTP extension has now been downloaded and requires additional setup within the backend of the Magento 2 site
   e. * For Magento 1, the Amasty SMTP plugin would need to be used instead
2. **Enable Mageplaza SMTP**
   a. Go to Stores > Configuration > Mageplaza Extensions > SMTP
   b. Next to ‘Enable Mageplaza SMTP’, change to ‘Yes’
3. **Configure SMTP Configuration Options**
   a. Scroll down in the configuration page for SMTP to find SMTP Configuration Options
   b. Select ‘Load Settings’
   c. Select ‘SendGrid’ from the options listed
      i. Details about the connection will auto-populate in the respective fields; however, these pieces will need to be adjusted
   d. For CapitalCityShoes, change the port number to **2525**
      i. If 2525 is found to not work, check the firewall settings on the VM to confirm what port is available for this network traffic and use that one
      ii. The live server uses port 587 instead
   e. Use the TLS Protocol
   f. For authentication, use the ‘LOGIN’ option
      i. The username for Magento 1 and Magento 2 is ‘**apikey**’
      ii. The password is the API key created through SendGrid for that site
4. **If no new API key has been created yet**
   a. Go to SendGrid.com and sign in
   b. Go to Settings > API Keys
   c. Select ‘Create API Key’
   d. Match the naming convention to be similar to the other API keys already created
      i. ex: **CapitalCityShoes-Mail-APIKEY-Magento2-SANDBOX**
   e. Use the ‘Restricted Access’ option
   f. Select the ‘Full Access’ option for Mail Send
   g. Select ‘Create and View’
   h. In the next screen, you are shown the new API key password; copy that password over into the Magento 2 password field for the SMTP configuration
      i. Magento 1 requires the same process
5. **Send a test email**
   a. Open the option ‘Send Test Email’
   b. Ensure that the option for ‘Send To’ is set to an email appropriate for testing
      i. i.e. dev@dang-designs.com
   c. Click ‘Test Now’
   d. If the test fails, attempt to troubleshoot the port and server firewall rules
6. **You’re Done!**

How To Update Punchout Site URL On Ariba

This page will describe the details needed to update the URL for the global value for the site to redirect to when the user is connecting to the MyFloridaMarketplace site. This step requires that the main location of the site needs to be adjusted in general and caution should be taken with regards to making any type of modification to this value.

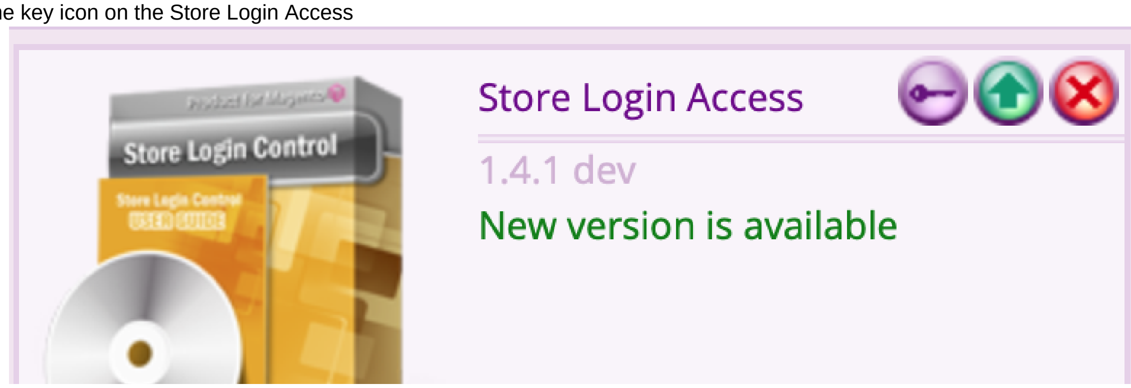
1. **Log into the Ariba network as the production user**
2. **On the right-hand side, there is a circular icon that represents the user, left-click this**
3. **Go to Settings > Electronic Order Routing**
4. **When the page loads, select the hyperlink named 'Configure cXML (native) integration**
5. **Update the URL for the profile URL and Punchout URL fields**
   a. The punchout URL, if still using Avetti, should be similar to the current URL
      i. https://punchoutgateway.avetti.ca/preview/punchout.xml?vid=20150913312
6. **Press ‘Save’**
7. **You’re Done!**

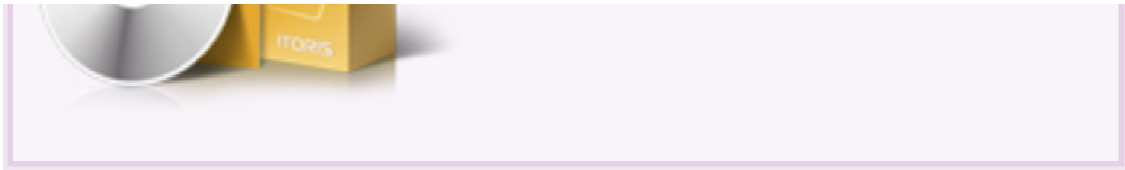
How To Update Sandbox For Capitalcityshoes With Live Files

This page will walk a user through how to update the files on the development sandbox environment for Capitalcityshoes.com.

1. **Download a copy of the files from the live capitalcityshoes domain.**
2. **Download a copy of the database from the live capitalcityshoes domain.**
   a. Use the Backup option within the cpanel from the live site to pull a copy of the database (*should be a .gz file extension*)
   b. Push the new copy of files and database to GitHub as applicable to a new branch
3. **Navigate to the development WHW for capitalcityshoes and sign in with the root account.**
   a. https://test-vm.capitalcityshoes.com:2087
4. **Remove the existing database on the development Cpanel and restore a MySQL backup from the one pulled from the live site.**
   a. *Should errors occur, ensure that the WHM for the PHP version supports the file size by checking MultiPHP INI Editor*

5. **Pull down a copy of the local.xml file from the development environment.**
    a. located in public_html/app/etc/local.xml
6. **Compress the dev folders and files and delete all files and folders under public_html**
    a. This is done to keep a copy just in case of issues.
7. **Upload the copy of files/folders pulled from the live environment and extract them.**
8. **Update the connection data in the local.xml file**
    a. located in public_html/app/etc/local.xml
    b.
    ```
    <connection>
      <host><![CDATA[localhost]]></host>
      <username><![CDATA[shopccsh_mag17dd]]></username>
      <password><![CDATA[Ac518bIryxCllAVzCT]]></password>
      <dbname><![CDATA[shopccsh_cloud_shop]]></dbname>
      <initStatements><![CDATA[SET NAMES utf8]]></initStatements>
      <model><![CDATA[mysql4]]></model>
      <type><![CDATA[pdo_mysql]]></type>
      <pdoType><![CDATA[]]></pdoType>
      <active>1</active>
    </connection>
    ```
9. **Update the index.php file and put “sandbox” in all the URLs between the prefixed and the URL.**
    a. i.e. b2b.sandbox.capitalcityshoes.com
10. **Update the records in the database in the core_website table to include sandbox**
    a. Should be on or around page 7
    b. i.e. b2b.sandbox.capitalcityshoes.com
11. **Using the core_config_data table, do a search for “%capitalcityshoes.com%” and update all references to sites to include the “sandbox” prefix.**
12. **Update Itoris plugin information and tables**
    a. The database holds a few tables that may need to be updated to match the sandbox site and sandbox license key
        i. itoris_storelogincontrol_redirect and itoris_storelogincontrol_settings
        ii.
        ```
        update `itoris_storelogincontrol_redirect` set custom_url
        = "https://punchout.sandbox.capitalcityshoes.com" where
        custom_url = "https://punchout.capitalcityshoes.com"
        ```
        iii.
        ```
        update `itoris_storelogincontrol_redirect` set custom_url
        = "https://shop.sandbox.capitalcityshoes.com" where
        custom_url = "https://shop.capitalcityshoes.com/"
        ```
            1. There should be one or two urls in this list that don’t meet this criteria; update those manually.
    b. Log into the sandbox site and go to System > ITORIS Store
    c. Open the key icon on the Store Login Access
        i.





ii. The license key may need to be reset to account for the update; to do that, go here and login
1. https://www.itoris.com/

d. Select ‘Add’ and apply the license key

13. **After all items listed have been updated accurately, ensure that there are no other caches leftover that may cause issues.**
    a. In the rare circumstance we had during initial set up, there was a cache storage leftover in the server at /var/tmp and the file was listed as Magento
    b. Other caches may need to be removed manually
        i. Inside the public_html folder, this can be found in var/cache; delete all entries titled like “mage--0”
    c. If attempted to login multiple times, ensure that the MySQL user is not locked
        i. This can be checked in the terminal by using the following commands

ii.
```
sudo -i
```

iii.
```
sudo mysql -u root
```

iv.
```
Select user, host, account_locked from mysql.user;
```

v.
```
alter user 'USER'@'HOST' account unlock
```

1. This should be done for the user account when the account_locked column is listed as 'Y'

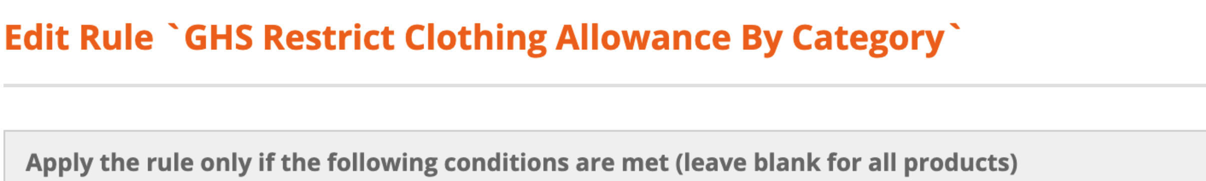
14. **You’re Done!**

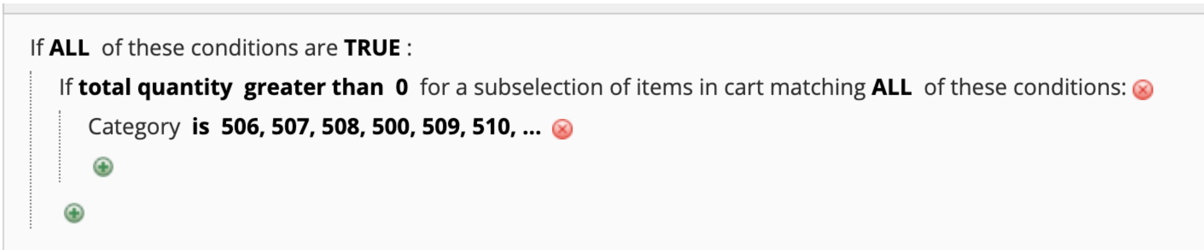
Payment Restrictions Per Website

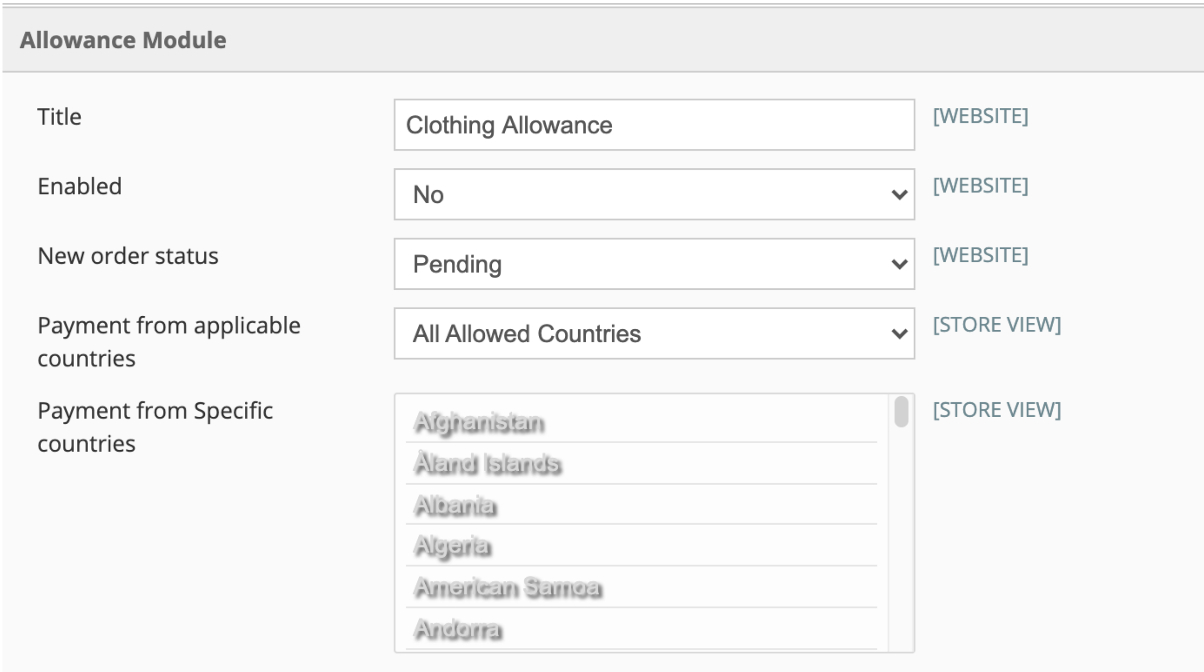
This page will document the payment restrictions for clothing allowance and other payment forms, and how those changes will need to be made for the site to function properly.

IMPORTANT The path to the function is located in **Sales > Payment Restrictions**

1. **Set the restrictions for the payment methods, the name, and the status of the rule**
    a. Name - Choose something that makes sense and explains exactly what the rule is for
    b. Status - Default is active unless meant to be turned off for testing.
    c. Methods - Select the appropriate methods needed to ensure that the method is available to use by the customer.
        i. This is where clothing allowance is set to be an option on the front end for orders.
2. **Set the Stores and Customer Groups**
    a. Under the ‘Apply In’ area, set the Admin Area field to ‘Yes’ and select the stores needed to be within the payment restriction group
        i. Ensure to use ‘CTRL’ + [click] or ‘CMD’ + [click] to make multiple selections
    b. Under the ‘Apply For’ section, customer groups can be selected to whom this restriction would apply for specifically.
3. **Days & Time and Coupons Fields are currently unused but allow for specific days of the week that these restrictions would allow or disallow**
4. **Under ‘Conditions’, you can set the requirements that would be required in order for the rule to be applied *(further scope control)***
    a.





5. **Enable the payment restriction per scope or globally on the System > Configuration > Sales > Payment Methods area**
    a. Globally, the clothing allowance section under the Allowance Module is disabled; otherwise all sites would have it set for them by default.
    b. If needed to be added to a specific site, change the scope to the appropriate website and set the Clothing Allowance > Enabled to 'Yes'
    c. 

6. **You're Done!**

How To Add Extension On Magento 2

This page will provide the steps required to add new extensions to Magento 2 from Amasty. There are a total of 5 plugins (as of this writing) that will need to be installed and have their own individual funtionality.

Plugins

- Color Swatches Pro
- Customer Attributes
- Hide Price
- Mass Product Actions
- Store Credit & Refund

1. **Login to Amasty customer portal**
    a. https://amasty.com/amcustomer/account/products/
    b. If the link auto-redirected back to the main dashboard, navigate to the products page
2. **Download the extension files to the local system**
    a. Navigate under Products below the 'Packages' section
    b. Press the 'Download' button for each extension intended for download
3. **Expand the folder to obtain the details and installation guides**
4. **Find the location within the folder path to create a new zip file to upload**
    a. For instance, if the folder structure is app > code > Amasty, then app and code folders are not needed. If Amasty already exists in the system, then copy the contents of the new information within the Amasty folder
    b. Make a zip file of the contents
    c. Copy that new zip folder to the server and extract it
5. **Upgrade, compile, and redeploy the site**
    a. Navigate to the terminal of the server and use the following commands

b.
```
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy -f
```

6. **Confirm that the new plugin is available on the site after installation**
7. **Perform these steps for the remaining plugins (individually)**
    a. These steps should be done piecemeal to avoid complications between extensions or the likelihood that pieces within the new extension may be ignored
    b. For Hide Prices, composer will need to be used to require another dependency

    i.
    ```
    su m2devcapitalcity
    composer require laminas/laminas-dom
    ```

8. **You're Done!**

How To Import New Users

This page will provide clarity on how to import users to ensure they have the appropriate views and balance (as needed). The scripts used are listed inside of another document that will be referenced within this document to provide all the pieces to make the updates accordingly.

IMPORTANT: All imported users should be tested on the Sandbox prior to making any changes on live.

1. **Get list of new users from client and set up the sheet to be prepared for import**
    a. an example of Moore Bass customers are located here
        i. https://docs.google.com/spreadsheets/d/1vw4dsE-a2Insxf4K1PfFgSQ8dZGSMHCoxAiFwdcHjxg/edit#gid=157783566
        ii. The headers all reference specific points in the tables that are needed to set users, passwords, website, etc.
        iii. am_is_activated = 1 for **NOT ACTIVATED**, am_is_activated = 2 for ACTIVATED
            1. To allows the users to receive an activation email at the time of writing this, the users must be activated in the system after being imported.*
    b. The group_id field references the Customer Group that the user would need to fall into; this determines whether the user receives taxes or not.
2. **Prepare the code file to update the views and credits (as needed) for the new users**
    a. Source code for these processes are found here
        i. Capitalcityshoes - Import Users View & Credit Update
    b. *IF amounts need to be updated, update the emails and amounts needed for script one.*
        i. Enter the new emails that will be used for their user email
            1. For sandbox, ensure the emails are referencing the dev instead of the live email addresses
                a. EX: joshg+user1@dang-designs.com
        ii. (AS NEEDED) Enter the amounts and ensure there are an equal number of amounts to emails
            1. meaning that email1, amount1, email2 amount2, etc.
        iii. Update the SQL statement that will be performed to make the insert statement
            1. Provided one is:

                a.
                ```
                Insert into amasty_storecredit_customer_balance
                (customer_id, website_id, amount,
                base_currency_code, subscribe_updates) values
                ($customer_id, 6, $amounts[$i], NULL, 1)
                ```

            2. The website_id (labeled as '6' in values) may need to be updated for a site other than b2b
                a. Website_id and code for import can be found in table **core_website**
    c. Update the script for views
        i. These are made separately to not have any types of conflict with one another.
        ii. On script two, enter the new emails needed to correct the views
        iii. Update the SQL Insert statements as needed
            1. Provided one is:

                a.
                ```
                Insert into itoris_storelogincontrol_customer
                (customer_id, store_id) values ($customer_id,
                ```

```
66)
Insert into itoris_storelogincontrol_customer
(customer_id, store_id) values ($customer_id,
79)
```

   i. The numbers on the end (66 and 79) reference the stores themselves
   ii. 66 is the default view for B2B and 79 is Moore Bass
   b. Update the store numbers to the ones that are needed for the new users
      i. These values can be found as the store_id in core_store table.

3. **After everything has been prepped and tested on the sandbox, import the users on the live**
   a. Sign into the live site
      i. https://shop.capitalcityshoes.com/index.php/dep2820
   b. Navigate to System > Import/Export > Import
   c. Set the 'Entity Type' to Customers
   d. Leave 'Import Behavior' to 'Append Complex Data'
   e. Choose the file as a CSV file from the sheet created to import
   f. On the top-right, press the 'Check Data' button
      i. If errors are found, correct them and check the data again.
      ii. Otherwise, press 'Import'
   g. Confirm the new users are present in Customers > Manage Customers
4. **Run the scripts to update the user views and balances**
   a. Save the scripts modified earlier on the live site on the root directory in the public_html folder
      i. Should already be present under the names of **ImportedUsersCreditUpdate.php** and **ImportedUsersStoreViewUpdate.php**
   b. Run the scripts by navigating to "site/[nameOfFile]"
      i. **Be sure not to run this twice or it will cause issues for the database**
5. **Update the system to send activation emails when users are activated**
   a. Navigate to System > Configuration > Customer Attributes
   b. Under 'Customer Account Activation Settings', Change 'New Customers Require Activation' to 'Yes'
   c. Under 'Notify Customer About Activation' change to 'Yes'
   d. Set the 'Customer Notification E-mail Template' to the template needed
   e. Press 'Save Config'
   f. Adjust 'New Customers Require Activation' back to 'No' and Press 'Save Config' again
6. **Activate the new customers and update their password to ensure they receive the email and password for login**
   a. Navigate to Customers > Manage Customers
   b. Enter the user account
   c. Ensure that the Store Login Access Settings are accurate, and (as needed) that the Store Credit is listed correctly
   d. Click 'Addresses' and remove the default empty address
   e. Click 'Account Information' and update 'Account Activated' to 'Yes'
   f. Press 'Save and Continue Edit'
   g. Paste the new password in the 'New Password' field at the bottom of the page and press 'Save Customer'
   h. Perform the aforementioned process for every customer imported
7. **Test the new accounts by logging in as them and ensuring that the views, balance, etc. all look accurate**
8. **You're Done!**

How To Correct User To Be Charged Taxes (Customer Groups)

This page will cover the concept of customer groups and what they do in Magento. Basically, customer groups are set to tax classes that determine whether or not the customer in said group will incur taxes upon purchases. Customer groups can be found in Customers > Customer Groups on the Magento site. They have only two values ('Retail Customer' or 'Tax Exempt').

NOTE: A list of stores, themes, and customer groups can be found here https://docs.google.com/spreadsheets/d/1sF3LnnsOf4La5SmatHjAdAYGZ82fOudX7YV8TnDKJ04/edit#gid=0

1. **If a new customer group needs to be created…**
   a. Navigate to Customer > Customer Groups in Magento site
   b. Select 'Add New Customer Group'
   c. Enter the name of the new Customer Group and ensure to set whether they should receive taxes (Retail Customer), or should be tax exempt
   d. Press 'Save Customer Group'
   e. For singular user, update their Customer group on Customers > Manage Customers
   f. For multiple customers, research the customer_entity table and update their group_id using the script below in SQL

i.
```
update `customer_entity` set group_id = 21 where
entity_id between 1754 and 1809
```

        1. This example was from the sandbox site, so will not reflect the same values as the live site
2. **If a customer group needs to be updated that is already existing…**
    a. Navigate to Customer > Customer Groups
    b. Select the customer group that needs to be updated
    c. Update the ‘Tax Class’ field to the proper value
    d. Press ‘Save Customer Group’
3. **You’re Done!**

How To Update User Passwords & Advise Them

This page will discuss the process for updating a user’s password and how it should be provided to them. Unlike the standard best practice, users with this site should be provided passwords that are less secure as a practice.

If a user needs to have their password updated for any reason by a backend admin user, follow the steps listed below:

1. **Pull up the user’s account on the backend**
    a. Customers > Manage Customers
2. **Select Account Information and scroll down to Password Management**
3. **Enter a new password for the user similar to the following:**
    a. Shoebox678!
    b. Copy this password to your clipboard to send later
4. **On the top right, press ‘Save Customer’**
    a. A new password update email will send out to the user
5. **With the new password and the customer’s email address, pull up the website they are assigned to and sign in with the account**
    a. http://b2b.capitalcityshoes.com , http://ghs.capitalcityshoes.com , etc.
    b. Enter the user’s email and password into the fields
    c. If there are no issues with sign in, continue on to step 6.
6. **Open a new email and send the user a personal message with their new password**
7. **You’re Done!**

RICS Access

Sign in for RICS Management Software

URL: https://shoeboxtallahassee.enterprise.ricssoftware.com

Username: dangdesigns

Password: Dang12345!

CBD

This page will host all the children for any CBD navigation guide.

- Setting Up A CBD Store on Sunflora.org
- Setting Up A New Store In Salesforce
- Setting Up A New Subdomain View In Google Analytics
- Setting Up A Product In Magento (Simple & Bundle)
- How To Set Up Subdomains For CBDRX4U
- Connect A Fishbowl Client Remotely to the SANDBOX Server
- How to Add Pictures to Store in CBDRX4U
- *Confidential* Referral Credit Structure - Founders
- Creating A Branded Email
- How To Redirect AMP Page For Closed Stores
- How To Set Up A 3PL Order Import Template
- How To Set Up A New Magento Website Domain

Setting Up A CBD Store on Sunflora.org

Before Setting Up

- Make sure the affiliate has an affiliate agreement, lease, resale certificate and certificate of insurance. Found on salesforce
- Also, check to see who referred them. This is in salesforce under the contact of the affiliate. Then go to “Details.”
    - Once you found who referred them go to sunflora.org/StoreOrgChart.php
    - Find this store and make sure there aren’t any special deals
        - You can find a special deal by checking the parent stores at each level
        - Keep in mind all D stores under Gibsonia (B) will be set up as “D store split between Nola and Gibsonia”

How To Set Up An Affiliate to Order

1. Go to https://sunflora.org/index.php/mastercontrols/
2. Login using admin panel credentials set up by Dan

3. Go to “Customers” -> “Manage Customers”
4. Click “Add New Customer”
    a. Associate to Website: aff.sunflora.org
    b. Account activated: Yes
    c. Group: Affiliate Store
    d. First Name: Owner First Name
    e. Last Name: Owner Last Name
    f. Email: storelocation@cbdrx4u.com
    g. Location: Store City, State
    h. Send From: Affiliate Portal View
    i. Password: same as the email password
5. Click Save Customer
6. Go to Affiliate Pro -> Manage Affiliates -> Active Affiliates
7. Click “Add Affiliate”
    a. Affiliate Account: storelocation@cbdrx4u.com
    b. Group Name: Select a group and make sure to check for any special deals. Also if this is a second store you will need to go back and make the parent account “with Subs”
    c. Shipping Postcode: Zipcode from store address
    d. Withdrawal Notice Email: storelocation@cbdrx4u.com
    e. Withdrawal Method: Manual
    f. Click “Save Member”
8. Search store by email and reopen member information
    a. Referral Code: Store Number and State (ex. 009LA)
    b. Affiliate Parent: email of the parent account
        i. If this is a second store it will be the first store's email
    c. Click “Save Member” and check the store org chart to see if it was set properly.
9. Email Affiliate their login credentials

    Subject: Affiliate Account Activated
    Your affiliate account has been activated. You can now order your products through the affiliate portal on aff.sunflora.org.

    To access order please visit:
    aff.sunflora.org

    Log In Using:
    XXXXXX@cbdrx4u.com
    XXXXXXXXX

    Before placing your first order please sign up for a time to go through your first order with us. Using the following link https://dang-designs.agilecrm.com/calendar/brian

    How To Refer Customers To Your E-Commerce Page
    LINK to refer customers: XXXXX

    Referrer Code is: XXXXX

    Either method is sufficient.
    Creating your own personalized email invitation is as easy as logging into your account on SunFlora.

    Click on:
    >>My Referral Code
10. Once the affiliate account has been activated email Taylor and Kristen at Sunflora with the following information; Name, Store #, email, first or multi-store, and the parent store if there is one.

Setting Up A New Store In Salesforce

This page describes the process for creating a Salesforce account for a new store.

1. **Log into Salesforce.com**
2. **Go to Accounts and select ‘New’**
3. **Enter all data for the new account**
    a. Account Name = store location and state
        i. EXAMPLE: Lancaster, PA
    b. Parent Account = referring person’s first store
        i. EXAMPLE: Nora, IN
    c. Store Number
        i. Can be AF or FR
            1. EXAMPLE: FR-0109
    d. Phone
        i. Not always available and can be left empty
    e. Email
    f. PPP Zone Number

    g. Geofencing zip code = zip code
    h. Website
    i. Address
        i. Make to enter all data into the shipping section of the form
            1. Shipping Street
            2. Shipping City
            3. Shipping State/Province
            4. Shipping Zip Code
            5. Shipping Country
4. **Press the save button**
5. **You're Done!**

Setting Up A New Subdomain View In Google Analytics

This page defines the process for setting up the Google Analytics View for a new subdomain for CBDRX4U.com. The process requires that the view be updated from within the dgysel@cbdrx4u.com account in Google Analytics to ensure that the data is accurate.

1. **Log into Google Analytics**
    a. Ensure that you are using the dgysel@cbdrx4u.com email account
2. **On the bottom left-hand side of the page, Select 'Admin'**
3. **Select the account**
    a. Generally 2018 - 2019, 2020
        i. based on when the subdomain was added
    b. *International stores have their own account*
4. **Select the property**
    a. This relates to the state that the new subdomain was added to
5. **Select 'Create View' and create it for the new subdomain.**
    a. **Reporting View Name** = [store number] city, state [month year]
        i. EXAMPLE: #### cityname, st 10/20
    b. Select 'Create View'
    c. Add a filter
        i. Go to Filters > Add Filter
        ii. Filter name == **Reporting View Name**
        iii. Change filters > add filter > include only traffic to subdirectories that end with
        iv. Enter the subdirectory (EXAMPLE: /honolulu)
        v. Select 'Save'
6. **Add the new view to the Google Structured Data Component on CBDRX4U site**
    a. Log into the cbdr4u site for administrator
    b. Navigate to components > Google Structured Data > Items
    c. Scroll down and select the newly created view
    d. Save the component and clear the cache
7. **Test that the site is working**
    a. Open the 'New CBD Store Owner Info' Sheet and find the URL
        i. https://docs.google.com/spreadsheets/d/13jMFw7Sfm4ZVPvTz7J5fsLSUMQkkTFupHjF_a4h3Dgc/edit#gid=157328540
    b. On the Google Analytics tab, navigate to the Overview tab to see all active users
    c. Ensure that you see yourself on the map
8. **You're Done!**

Setting Up A Product In Magento (Simple & Bundle)

This page will walk a user through creating a product on the Magento site for simple (retail) and bundle (AFF, AFP, FR, etc.) products.

1. Log into Magento
2. Add Simple Product (Craig, start here)
3. Go to this location for product

How To Set Up Subdomains For CBDRX4U

This page will express how to make particular changes to a .htaccess file for the subdomain. This document is primarily for simple redirects and specifics towards cbdrx4u. However, the information is able to be used with any cpanel on any domain. For this document and explanation, we will be using the subdomain savannahga.cbdrx4u.com. Moreover, there will be screenshots that express each piece along the way so that there is minimal to no confusion. These screenshots will have explanations along with examples to assist with understanding and duplication as needed.

The Problem

Subdomains were being indexed and were entirely available if a user went to the link through the subdomain with something else on the end of the query string.

For example: savannahga.cbdrx4u.com cbdrx4u.com/find-us/georgia **CORRECT**

For example: savannahga.cbdrx4u.com/find-us/new-hampshire savannahga.cbdrx4u.com/find-us/new-hampshire **INCORRECT**

Intention: savannahga.cbdrx4u.com/find-us/new-hampshire cbdrx4u.com/find-us/new-hampshire **CORRECT**

The Solution

The initial attempt was to create a rule in the .htaccess file for the site that would alter this behavior for the redirect. However, due to rules within the main .htaccess file, the site would redirect to the index page — due to thinking that the link was incorrect — instead of providing the proper destination URI to the user. Therefore, the site required its own .htaccess file and a correction to the subdomain's Document Root field. Below is a picture of the savannahga subdomain in Cpanel after the proper corrections to the Document Root. As you can see, instead of going to /public_html directly, the path specifies a folder called "savannahga" (which is a folder created when the subdomain is created initially).

***UPDATE:* All subdomains now have a folder that is is based on the subdomain name in the 'subdomains' folder; therefore, the Document Root would be '/public_html/subdomains/savannahga'. When updated in CPanel, the link under 'Redirection' should look something like 'https://cbdrx4u.com/$1'**

Subdomains	Document Root	Redirection	Actions
savannahga.cbdrx4u.com	/public_html/savannahga	https://cbdrx4u.com/find-us/georgia/	Remove Manage Redirection

The next step is to update the .htaccess file inside of the folder for the subdomain (in this instance, "savannahga" which is located in /public_html /savannahga) to provide the redirection rules. As a main rule, we want all traffic coming from the store fliers to be led to the store's page (whether www or no prefix). Secondly, we want to ensure that any traffic that uses the subdomain with anything attached to the ending query string is directed to the cbdrx4u.com main domain along with whatever query string the user presented to the system. The whole chunk looks something like this:

```
RewriteCond %{HTTP_HOST} ^savannahga\.cbdrx4u\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.savannahga\.cbdrx4u\.com$
RewriteRule ^/?$ "https\:\/\/cbdrx4u\.com\/find\-us\/georgia\/" [R=301,
L]

RewriteCond %{HTTP_HOST} ^savannahga\.cbdrx4u\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.savannahga\.cbdrx4u\.com$
RewriteRule (.*)$ "https\:\/\/cbdrx4u\.com\/$1" [R=301,L]
```

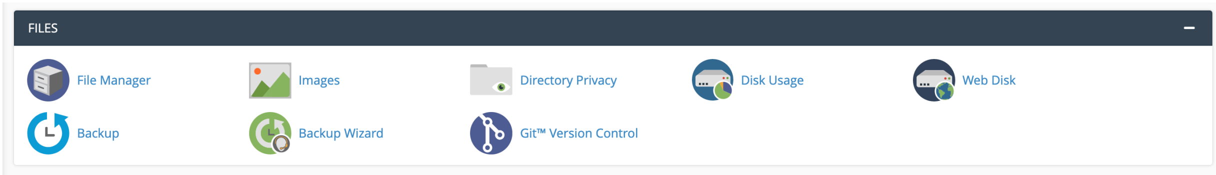
The Explanation

While all of these rules are pretty simple, I'll explain further what the meat of them are. Rewrite rules require two pieces, a condition and a rule. We'll start with the first two lines listed above.

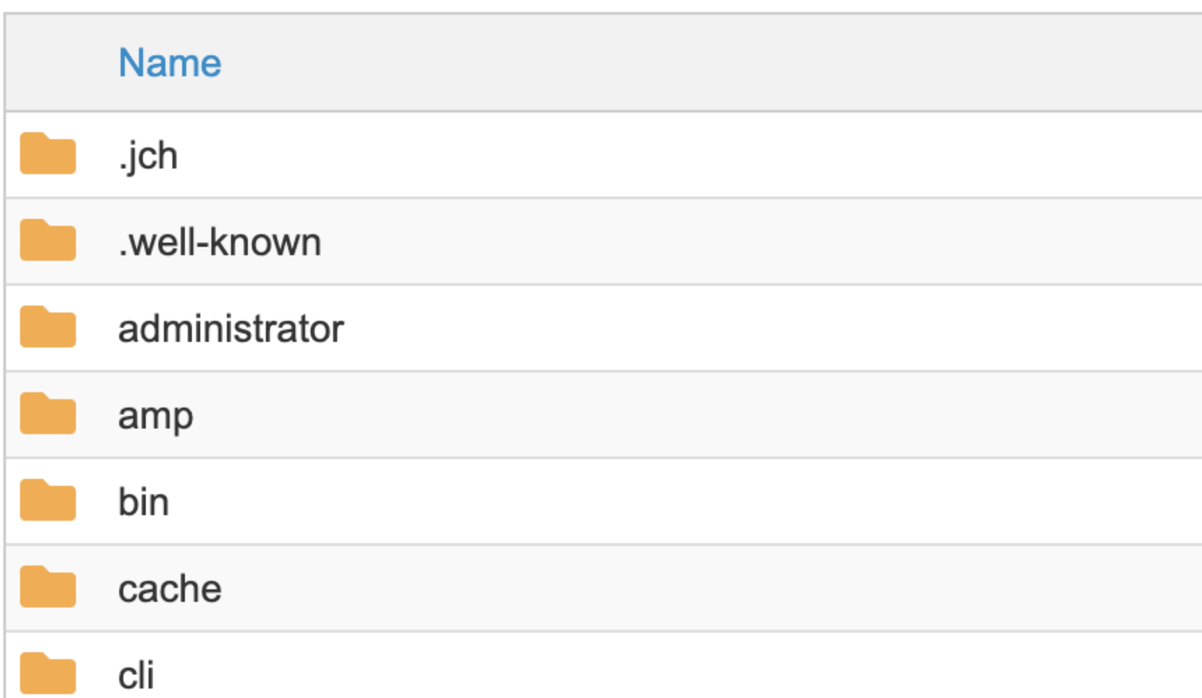
- In this line (`RewriteCond %{HTTP_HOST} ^savannahga\.cbdrx4u\.com$ [OR]`)
  - The %{HTTP_HOST} is a variable that essentially declares the "https://" portion of a URI.
  - The carrot (^) before savannahga means "begins like this" essentially.
  - The backslashes (\ ) aren't actually required but are considered good business practice; they escape the character presented so that the compiler won't throw an exception. You'll see that this is done for forward slashes in the following lines as well.
  - Finally, the [OR] statement indicates that there is another condition to check for (It can be this condition, OR it can be this one)..
- In this line (`RewriteCond %{HTTP_HOST} ^www\.savannahga\.cbdrx4u\.com$`)
  - The www is specified to captured those entries as well as non-www entries
  - There is a dollar sign ($) which references the remainder of the query string, using PHP, and can be called later by using either %{QUERY_STRING} OR $1; In this portion, it can honestly be omitted since it is never used.
- In this line(`RewriteRule ^/?$ "https\:\/\/cbdrx4u\.com\/find\-us\/georgia\/" [R=301,L]`)
  - The rewrite rule declares what we do with our conditions if met. The first portion (/?$) is a mixture of regex and PHP
    - ^ matches position just before the first character of the string
    - $ matches position just after the last character of the string
    - ? matches one or more characters
  - As previously mentioned, the special characters are escaped with the backslash (\ )
  - `[R=301,L]` specifies two pieces
    - R specifies the redirect type; 301 is a permanent redirect, and 302 is a temporary redirect
    - L specifies that this is the last rule in this run.
- In this line (`RewriteRule (.*)$ "https\:\/\/cbdrx4u\.com\/$1" [R=301,L]`)
  - Similar to before, we have regex being used that specifies with (.*) that **EVERYTHING** that meets the previously stated conditions should be set to this rule.
  - The $1 references whatever was on the end of the query string after where the dollar sign ($) was called (`cbdrx4u\.com$`)

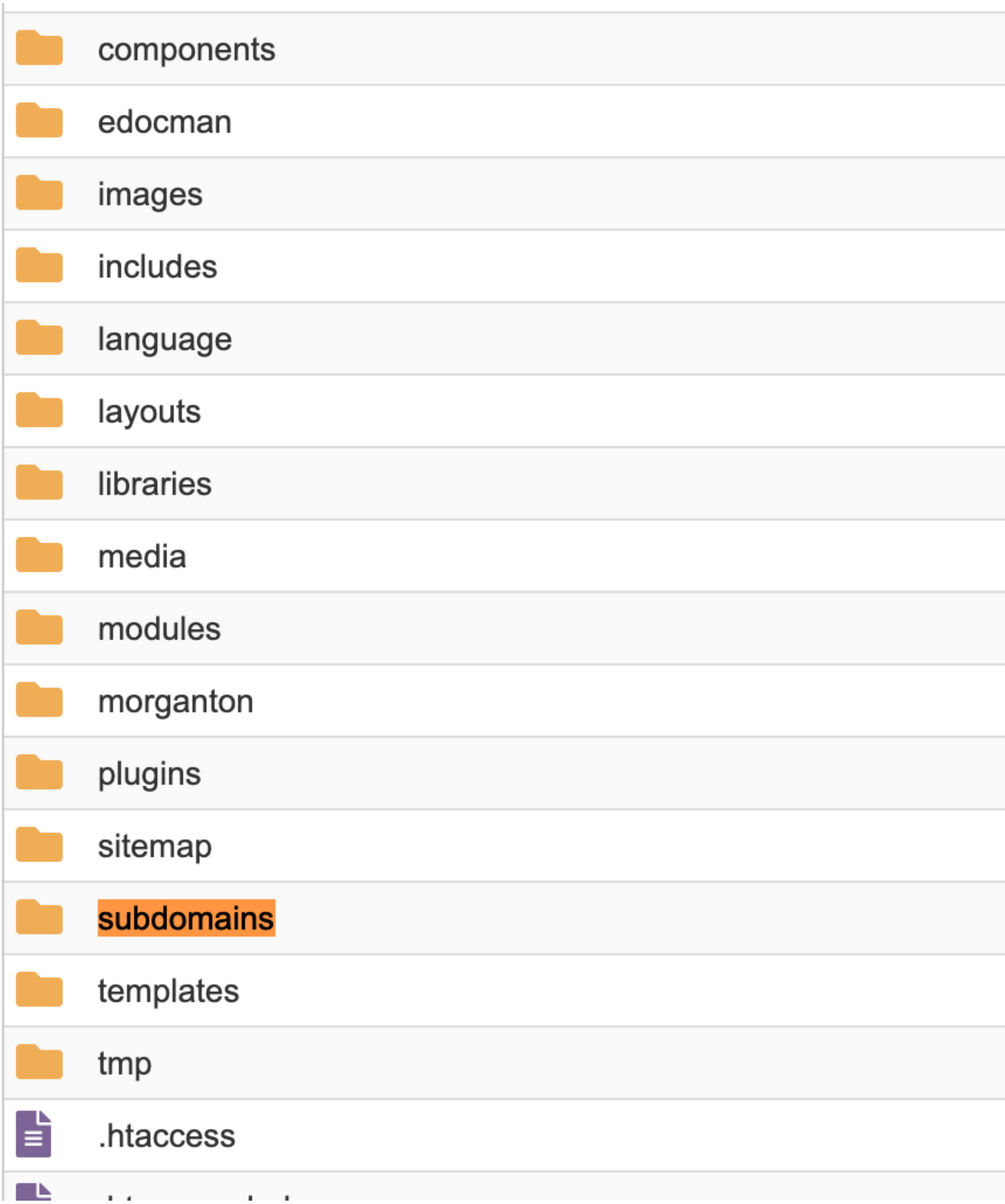
Setting Up New Subdomains

1. **Log into the cpanel for CBDRX4U.com**

2. **Select the 'File Manager' option listed on the home cpanel**
   a. 

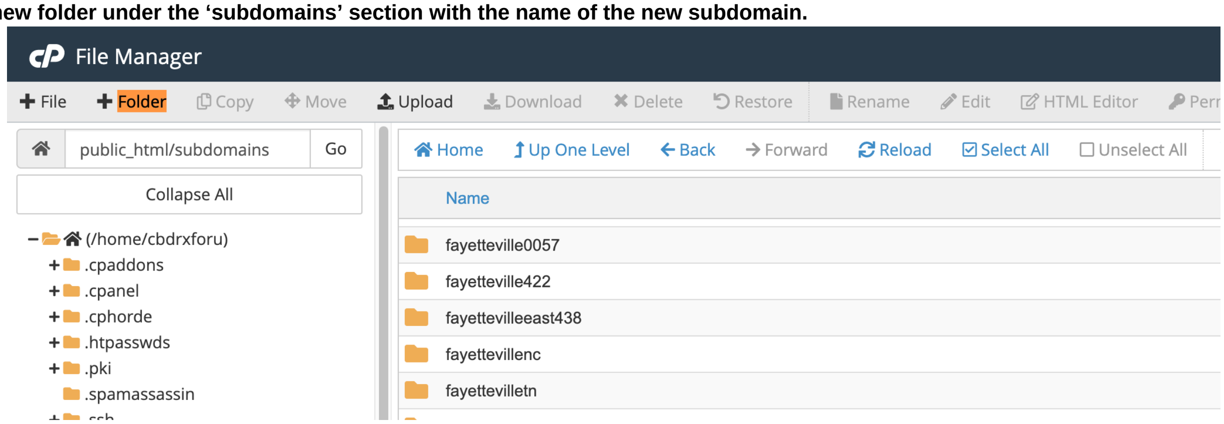
3. **Navigate to public_html > subdomains**
   a. 

   b. 

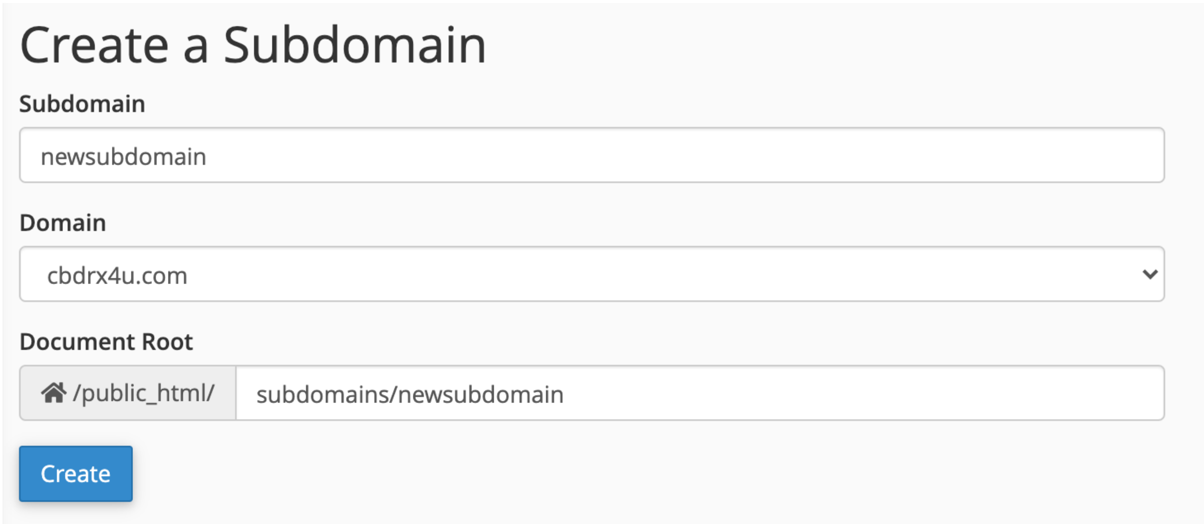


4. Add a new folder under the ‘subdomains’ section with the name of the new subdomain.

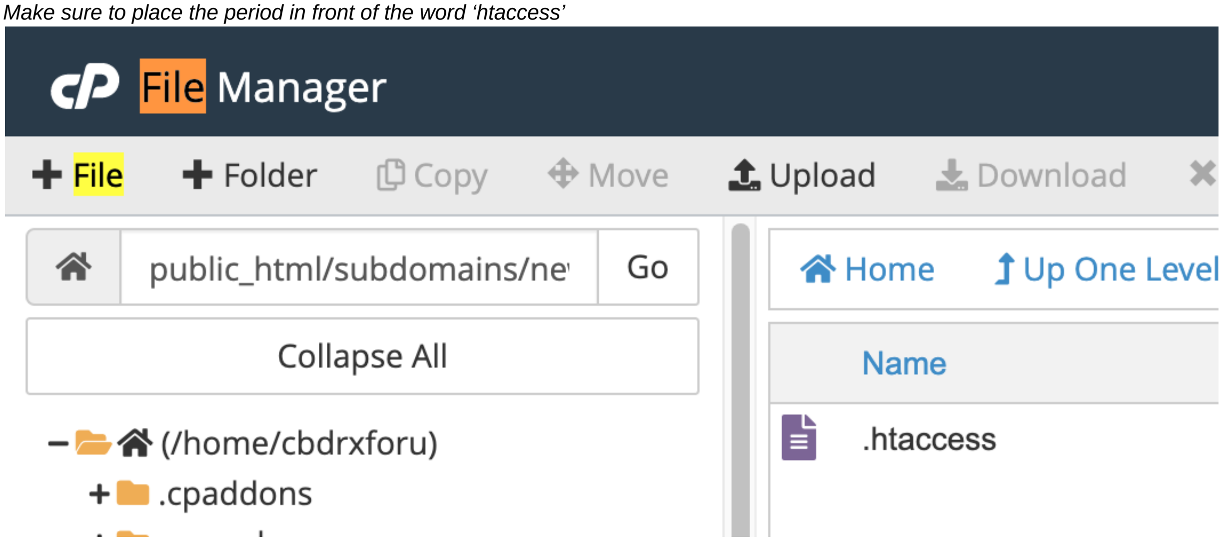
a.



b.



5. **Navigate inside of the new folder created**
6. **Select ‘File’ to create a new file called ‘.htaccess’**
   a. *Make sure to place the period in front of the word ‘htaccess’*
   b.



7. **Right-click on the .htaccess file and add the contents required**
   a. If the site will redirect to truecbdrx4u OR is located in Alabama, use this code

i.

```
RewriteEngine on

# php -- BEGIN cPanel-generated handler, do not edit
# Set the “ea-php74” package as the default “PHP”
programming language.
<IfModule mime_module>
  AddHandler application/x-httpd-ea-php74 .php .php7 .
phtml
</IfModule>
# php -- END cPanel-generated handler, do not edit
RewriteCond %{HTTP_HOST} ^birmingham\.cbdrx4u\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.birmingham\.cbdrx4u\.com$
RewriteRule ^/?$ "https\:\/\/truecbd4u\.com\/find\-us\
/alabama\/birmingham" [R=301,L]
```

```
RewriteCond %{HTTP_HOST} ^birmingham\.cbdrx4u\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.birmingham\.cbdrx4u\.com$
RewriteRule (.*)$ "https\:\/\/cbdrx4u\.com\/$1" [R=301,L]
```

ii. Every instance of birmingham would need to be replaced with the new subdomain name and the rewrite rule will need to be updated to point to the correct path to the new store's page.
iii. For example, if the new subdomain is 'newsubdomain' and the store location is in 'alabaster' then the new piece will look something like this:

1.

```
RewriteEngine on

# php -- BEGIN cPanel-generated handler, do not edit
# Set the “ea-php74” package as the default “PHP”
programming language.
<IfModule mime_module>
  AddHandler application/x-httpd-ea-php74 .php .php7
.phtml
</IfModule>
# php -- END cPanel-generated handler, do not edit
RewriteCond %{HTTP_HOST} ^newsubdomain\.cbdrx4u\.
com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.newsubdomain\.
cbdrx4u\.com$
RewriteRule ^/?$ "https\:\/\/truecbd4u\.com\/find\-
us\/alabama\/alabaster" [R=301,L]

RewriteCond %{HTTP_HOST} ^newsubdomain\.cbdrx4u\.
com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.newsubdomain\.
cbdrx4u\.com$
RewriteRule (.*)$ "https\:\/\/cbdrx4u\.com\/$1"
[R=301,L]
```

b. If the site is staying in CBDRX4U, use this code

i.

```
RewriteEngine on

# php -- BEGIN cPanel-generated handler, do not edit
# Set the “ea-php74” package as the default “PHP”
programming language.
<IfModule mime_module>
  AddHandler application/x-httpd-ea-php74 .php .php7 .
phtml
</IfModule>
# php -- END cPanel-generated handler, do not edit
RewriteCond %{HTTP_HOST} ^columbus\.cbdrx4u\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.columbus\.cbdrx4u\.com$
RewriteRule ^/?$ "https\:\/\/cbdrx4u\.com\/find\-us\
```

```
/georgia\/columbus" [R=301,L]

RewriteCond %{HTTP_HOST} ^columbus\.cbdrx4u\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.columbus\.cbdrx4u\.com$
RewriteRule (.*)$ "https\:\/\/cbdrx4u\.com\/$1" [R=301,L]
```

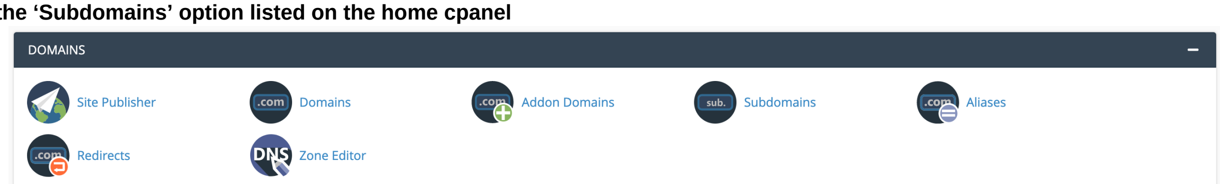
ii. Every instance of columbus would need to be replaced with the new subdomain name and the rewrite rule will need to be updated to point to the correct path to the new store's page.

iii. For example, if the new subdomain is 'newsubdomain' and the store location is in Mandan North Dakota then the new piece will look something like this:

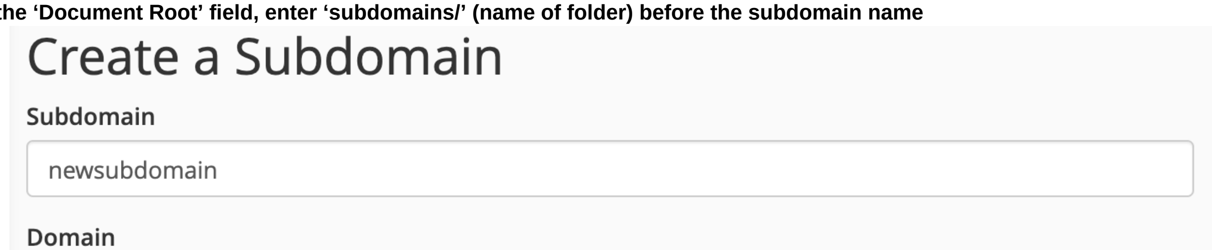
1.
```
RewriteEngine on

# php -- BEGIN cPanel-generated handler, do not edit
# Set the “ea-php74” package as the default “PHP”
programming language.
<IfModule mime_module>
  AddHandler application/x-httpd-ea-php74 .php .php7
.phtml
</IfModule>
# php -- END cPanel-generated handler, do not edit
RewriteCond %{HTTP_HOST} ^newsubdomain\.cbdrx4u\.
com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.newsubdomain\.
cbdrx4u\.com$
RewriteRule ^/?$ "https\:\/\/cbdrx4u\.com\/find\-us\
/north-dakota\/fargo" [R=301,L]

RewriteCond %{HTTP_HOST} ^newsubdomain\.cbdrx4u\.
com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.newsubdomain\.
cbdrx4u\.com$
RewriteRule (.*)$ "https\:\/\/cbdrx4u\.com\/$1"
[R=301,L]
```

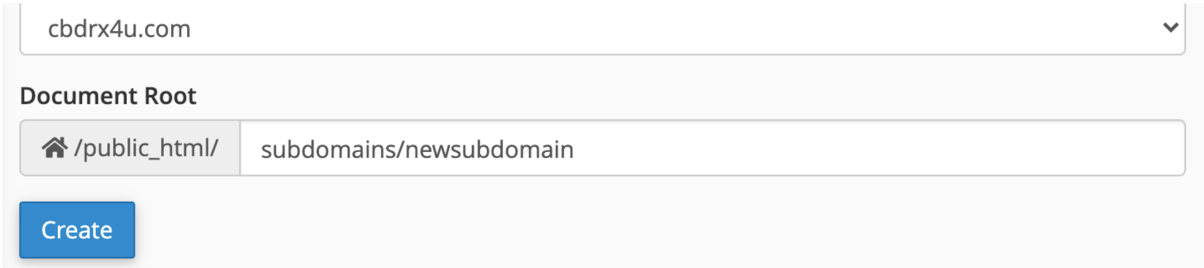
c. Save the changes and close the document

8. **Navigate back to the main cpanel section**
9. **Select the 'Subdomains' option listed on the home cpanel**

a. 

10. **Under the 'Create a Subdomain' field, Add the subdomain name under 'Subdomain' field**
11. **Under the 'Document Root' field, enter 'subdomains/' (name of folder) before the subdomain name**

a. 



12. **Press 'Create'**
13. **Make sure that your new subdomain looks similar to this**
    a.

Subdomains	Document Root	Redirection
absecon460.cbdrx4u.com	/public_html/subdomains/absecon460	https://cbdrx4u.com/$1

14. **You're Done!**

Connect A Fishbowl Client Remotely to the SANDBOX Server

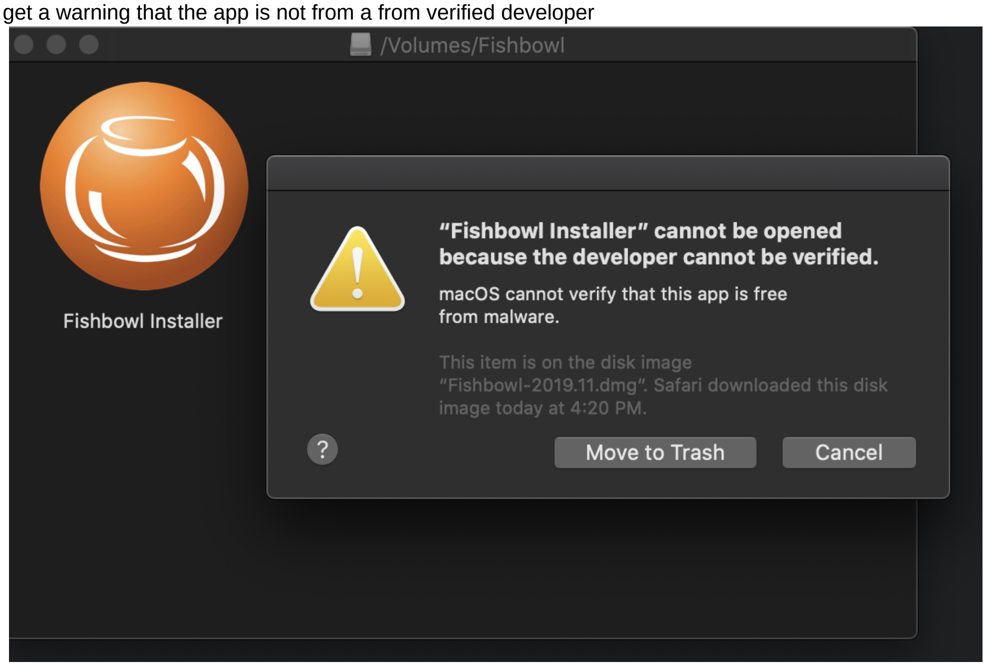
This page will provide guidance on how to connect a remote Fishbowl client to Windows2 Sandbox hosted on Google Cloud Platform.

WARNING: This is a copy of the live instance and may have connection points to other live services. Please do not run any scheduled tasks without ensuring the task is properly connect to another sandbox service and not a live service (i.e. Quickbooks, Magento, Shipstation)

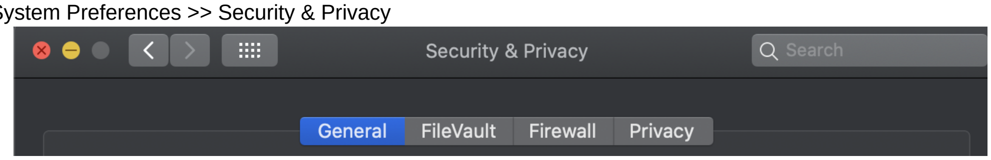
1. **Download Fishbowl Client on machine -**
    a. **Download version 2019.11 - https://www.fishbowlinventory.com/support/updates/download**

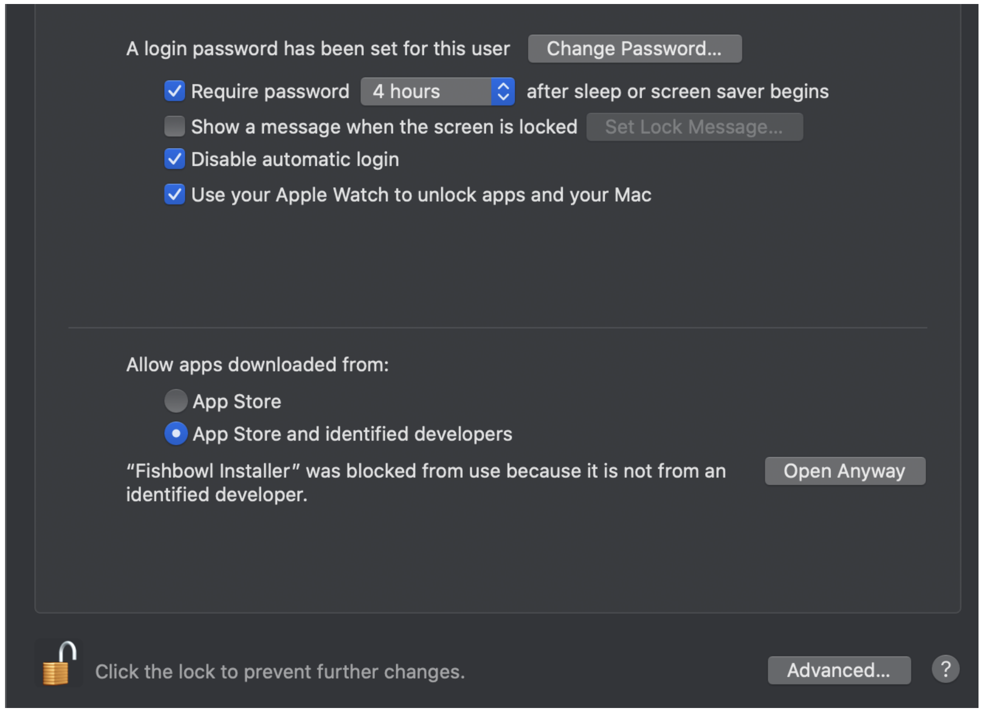
    Fishbowl 2019 (Former Version - 2019.11)
    - Windows Version (64 bit)
    - Windows Version (32 bit) (**Server not supported**)
    - Mac Version (Limited Support, Must have 10.7.5 or newer)
        - Download into a shared folder (typically "Public") in your user files
    - Linux Version (Limited Support, Tested on Ubuntu)

    b. **For Mac Version,** the client will need to be added in the Root account after allowing the install from your main admin user:
        i. https://support.apple.com/en-us/HT204012
        ii. Download using an admin user to the Users Public Folder (you will need to access the installer from the root user navigating to this shared folder)
        iii. then start the installation in your admin user
            1. you will get a warning that the app is not from a from verified developer
                a.

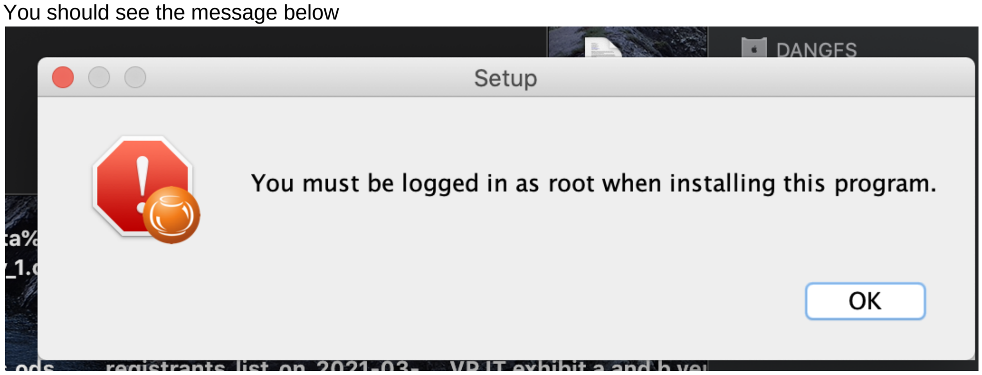


                b. **Select "Cancel"**
            2. Open System Preferences >> Security & Privacy
                a.



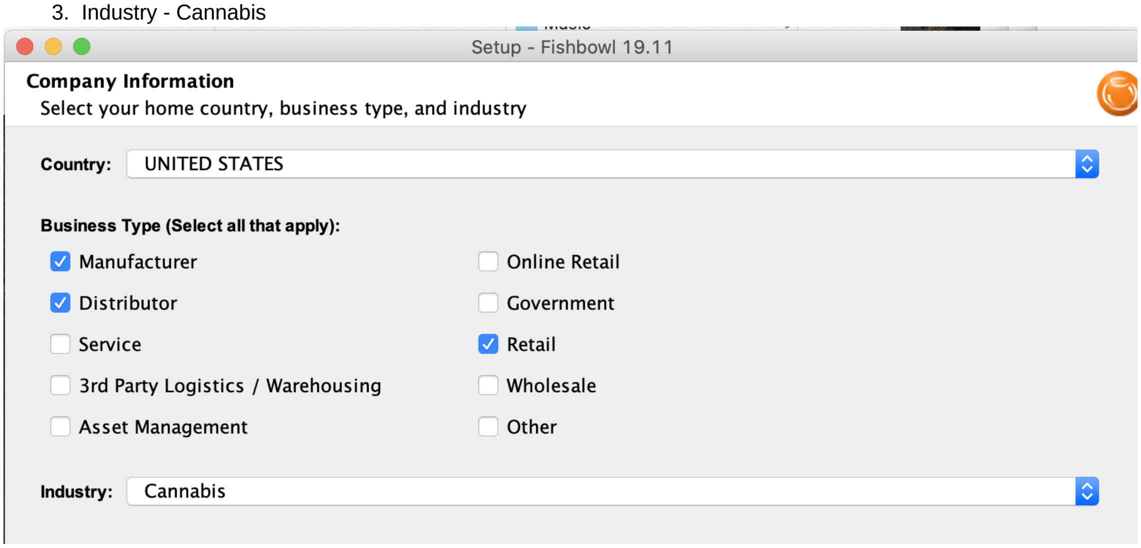


3. **Unlock the settings and click "open anyway"**
4. **Open the installer again**
    a. You should see the message below



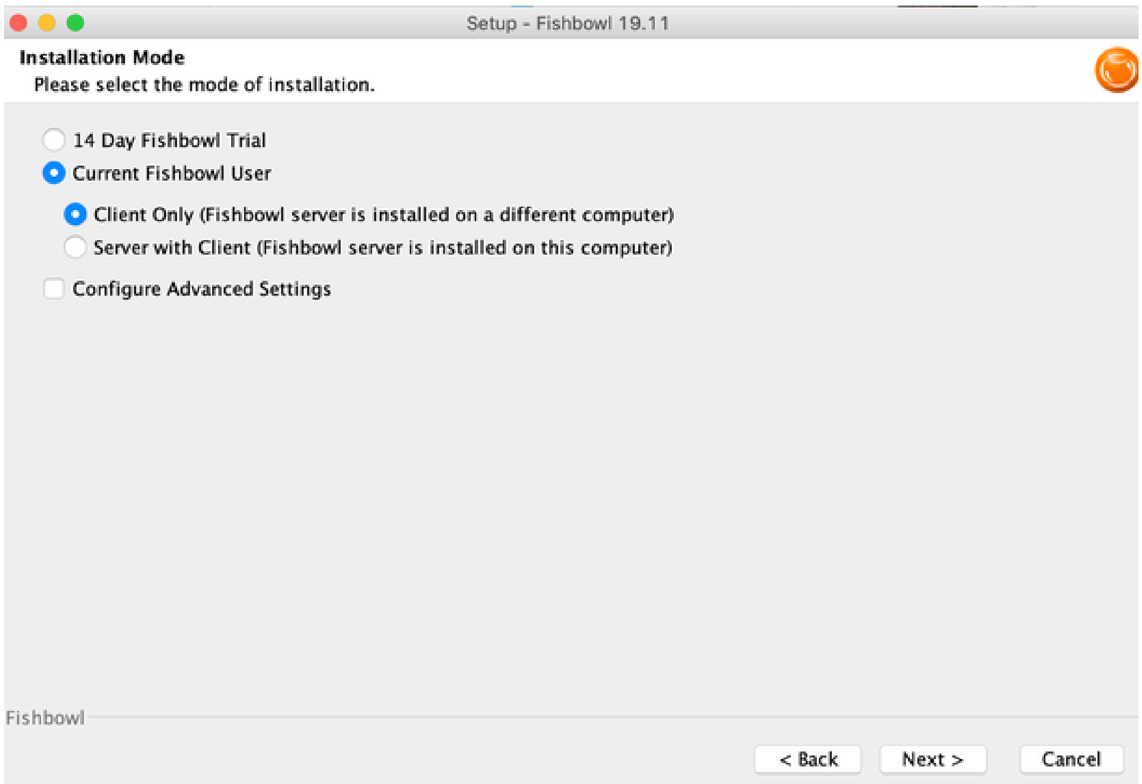
5. **Switch to the root user** using the "log in window" **- Do not log out the admin user**
6. **Open the Fishbowl-2019.11.dmg** from the admin user's public folder

2. **Run the Installer** (if you get the not from a verified developer message as the root user, repeat the steps above)
    a. **Company Information**
        i. Select the
            1. Country - US
            2. Business Type (doesn't effect the Client Only option)- Manufacturer, Distributor, Retail
            3. Industry - Cannabis
        ii.

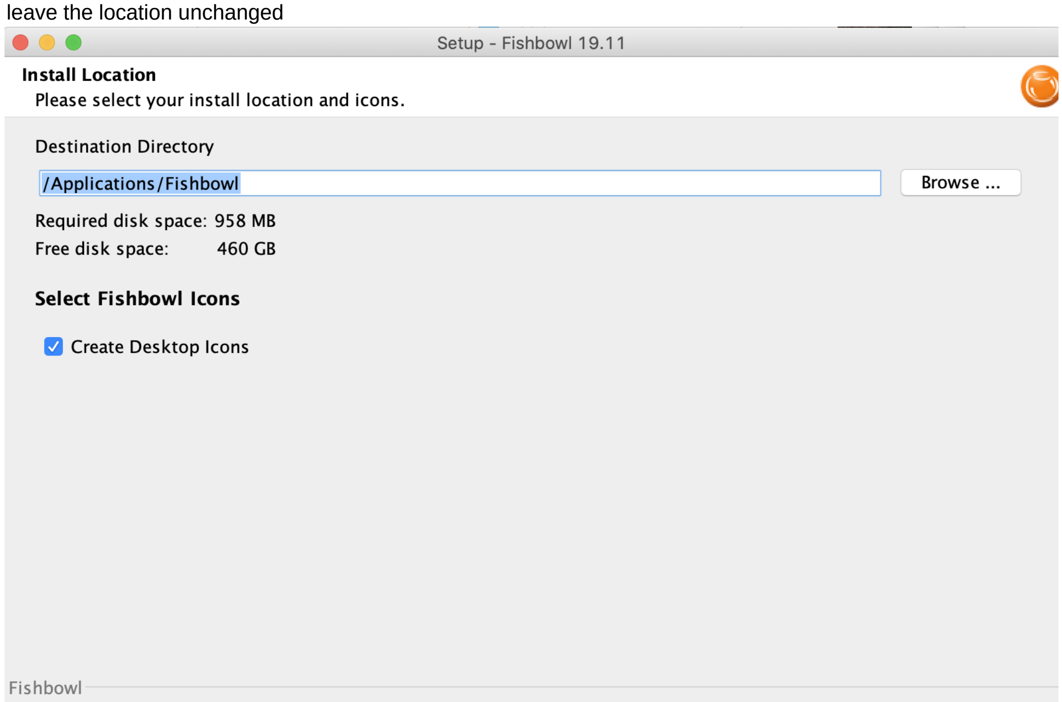


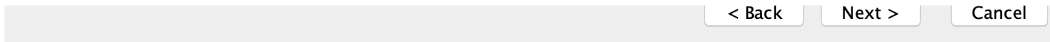
b. **Client Only:** This option will install the Client version of Fishbowl

i. Do not install the “Server with Client”

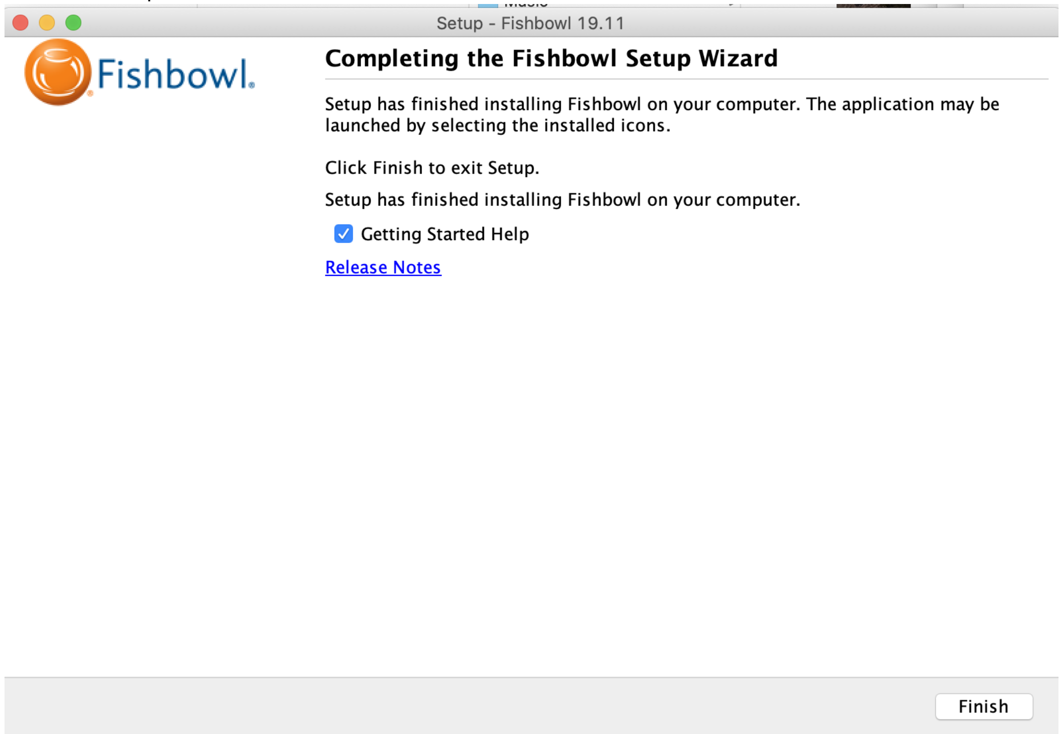


ii. You can leave the location unchanged

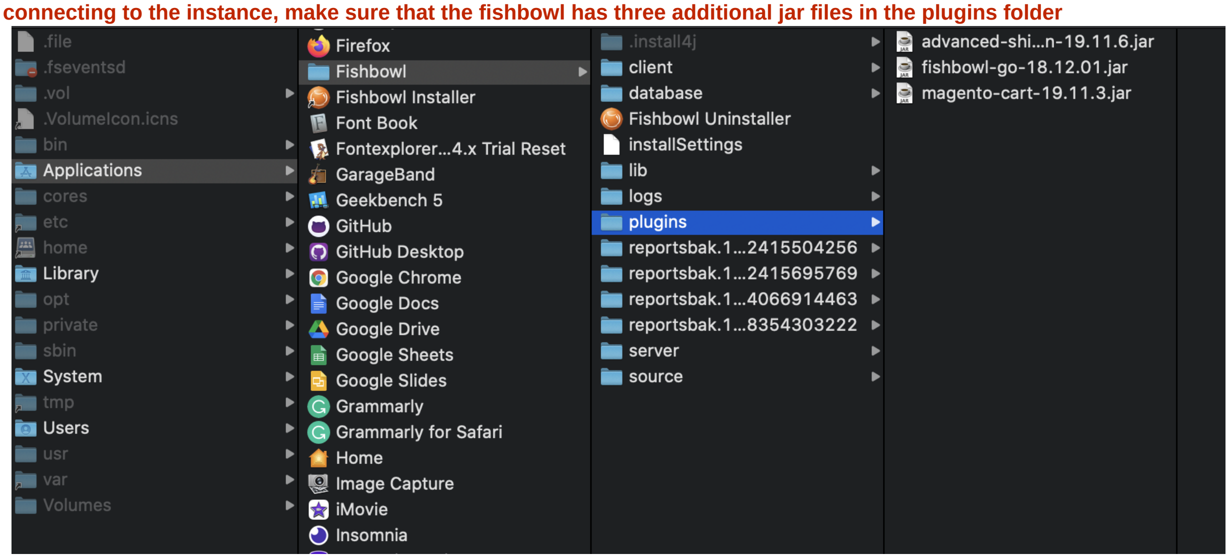




1. Finish the set up



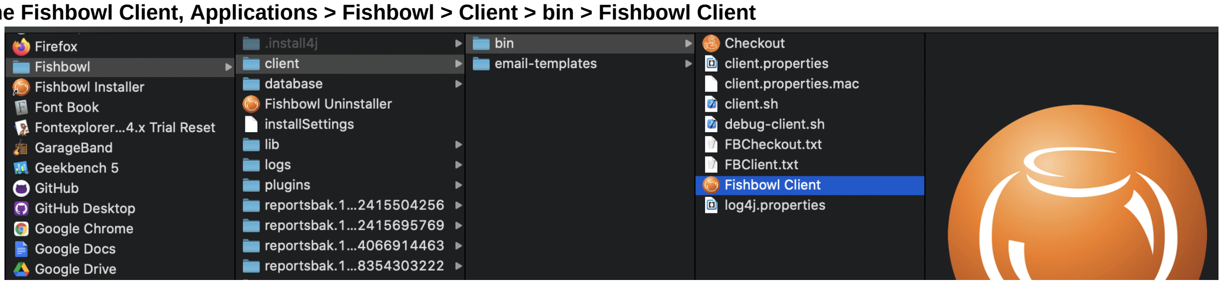
2. Now you may **log out of the Root User**
3. **Log back in** to your normal user
4. **Before connecting to the instance, make sure that the fishbowl has three additional jar files in the plugins folder**

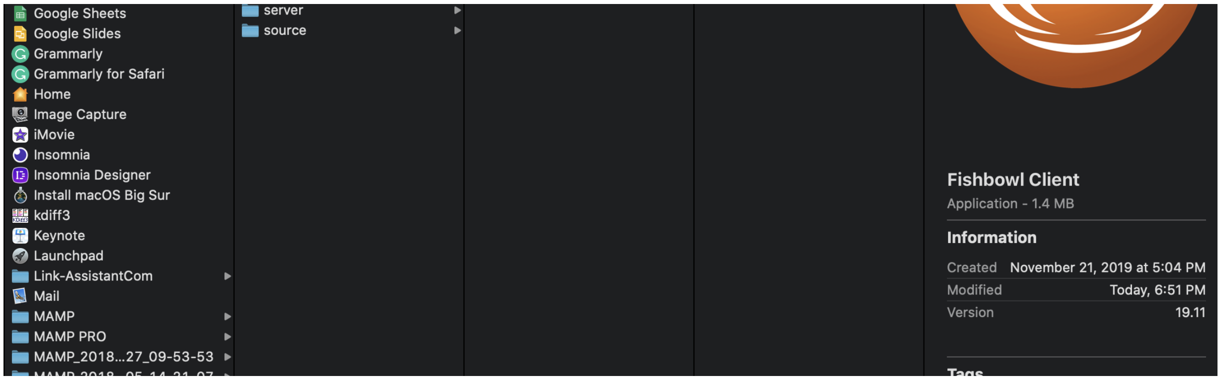


a. These files are attached to the email
   i. Advanced-shipstation
   ii. Fishbowl-go
   iii. Magento-cart

5. **Open the Fishbowl Client, Applications > Fishbowl > Client > bin > Fishbowl Client**

a.



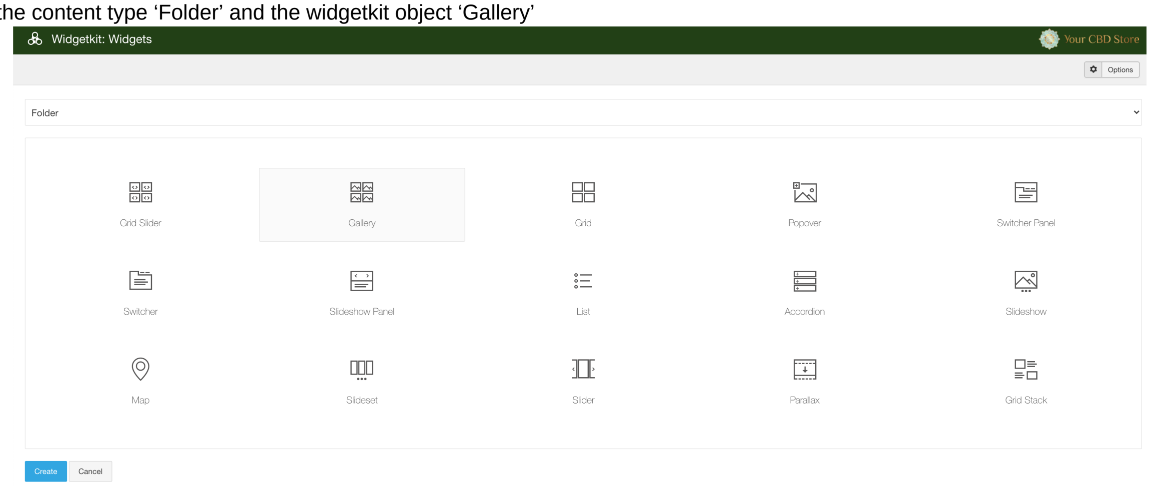


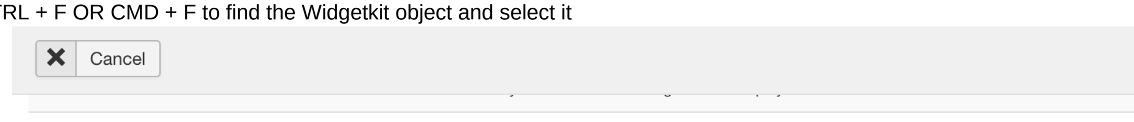
6. **Set the connection criteria**
   a. Username: advic
   b. Password: Ur&bM_U-2W
   c. **GCP Windows Server 2 (sandbox) IP :** 35.243.225.145
      i. Ensure that the Server Port field is set for 28192
7. **Press 'Login' to connect to the instance**
   a. Should any issues arise, please call Dan at: 813-461-1773
   b. **WARNING: Please do not run any scheduled tasks**

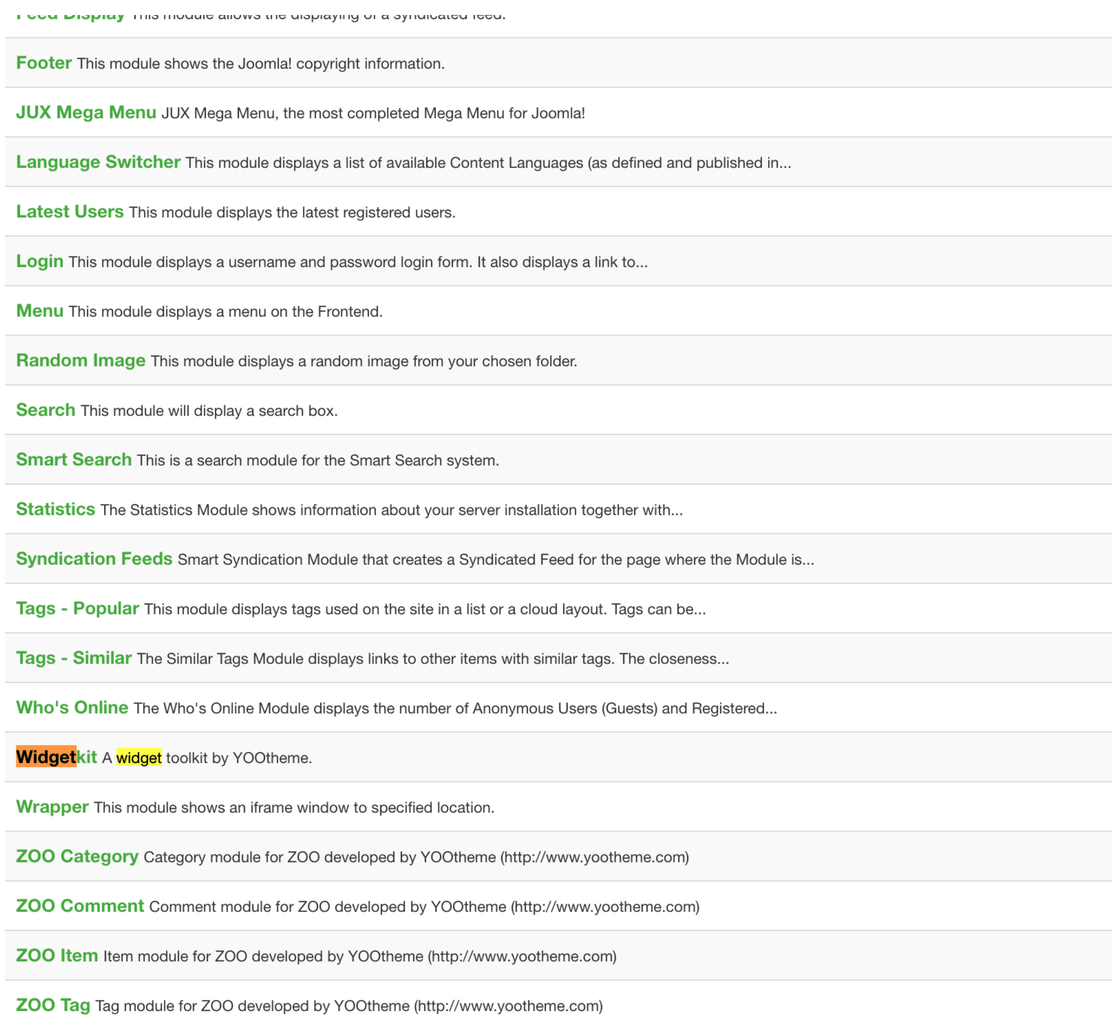
7. **You're Done!**

How to Add Pictures to Store in CBDRX4U

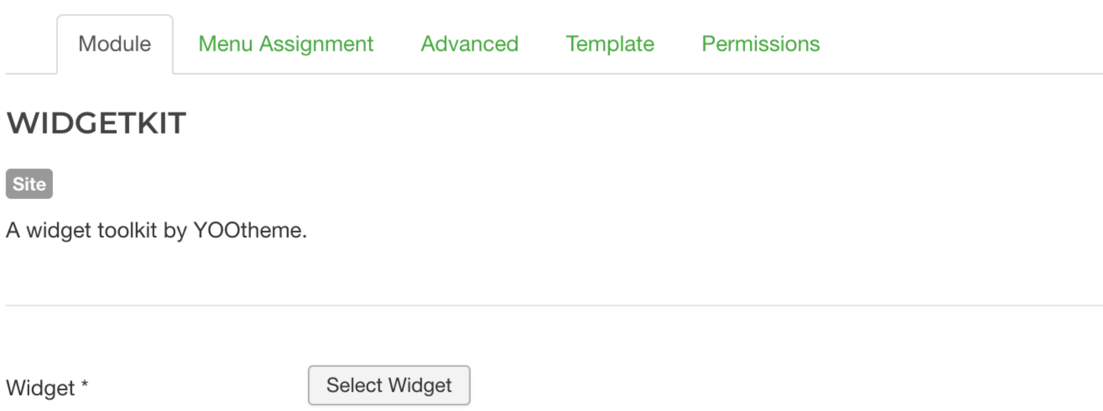
This page will outline the steps required to add photos to a new store menu item created in CBDRX4U. This process is performed in a way to be as dynamic as possible while retaining the visible consistently across all web page items.

1. **Add the new photos into the CPanel File manager**
   a. While this may seem like a simple task, it's important that the same convention is followed for each item to ensure consistency across the board.
   b. The files for these images are stored in home/cbdrxforu/public_html/images/000-Store-Galleries/{STATE}/{STORENAME-STORENUMBER}
2. **Create a Widgetkit Folder object inside of Joomla site backend (administrator)**
   a. Log into the site at the following web address
      i. https://cbdrx4u.com/administrator
   b. Navigate to Components > Widgetkit
   c. Press the green New button in the top-left hand corner
   d. Select the content type 'Folder' and the widgetkit object 'Gallery'
      i. 

   e. Press **Create**
   f. Enter the appropriate folder path and leave all other defaults the same
      i. This should appear something like **images/000-Store-Galleries/{STATE}/{STORENAME-STORENUMBER}**
   g. Press Save & Close at the top-left of the page
3. **Create Module that references the Widgetkit object created**
   a. Navigate to Extensions > Modules
   b. Press the green New button in the top-left hand corner
   c. Use CTRL + F OR CMD + F to find the Widgetkit object and select it
      i. 

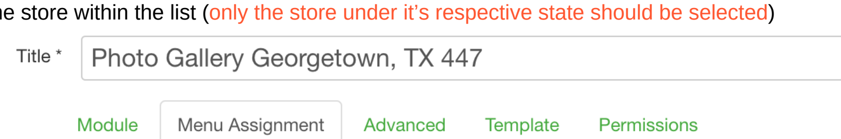


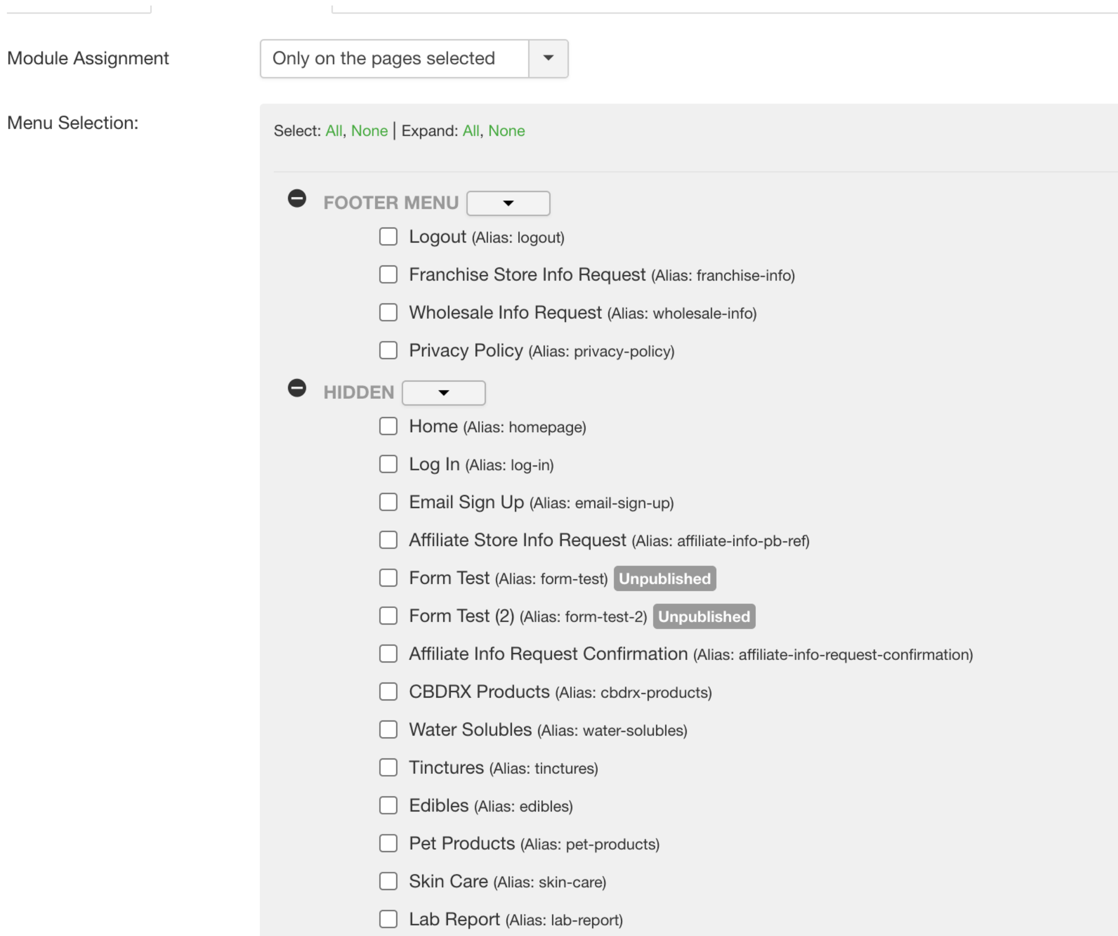
d. Set the title for the module as Photo Gallery {STORECITY}, {STATE} - {STORENUMBER}
e. Use the 'Select Widget' Button to open the menu and select your Widgetkit object
   i.



f. Press Save

4. **Assign the module to the menu item within the new created modules settings**
   a. Navigate to the 'Menu Assignment' link within the same module
   b. Check the store within the list (only the store under it's respective state should be selected)
      i.





5. **You're done!**

***Confidential* Referral Credit Structure - Founders**

General Referral Credit Structure

Credit - A percent from referral store's orders you receive as founders.

Self Credit - A percentage you receive on your own orders

Founding Store Scenario 1 - Self Credit

When you open a second store you receive a self credit when you place an order from either store. All orders placed in Store 1 will receive 10% self credit in the Store 1 account. All orders placed in Store 2 will receive a 10% self credit.

Formula: (20÷Total # Stores) Multiplied by # of Sub Stores=Self Credit%

2 Stores: (20÷2)1=10%

3 Stores: (20÷3)2=13.33%

4 Stores: (20÷4)3=15%

5 Stores: (20÷5)4=16%

6 Stores: (20÷6)5=16.66%

7 Stores: (20÷7)6=17.143%

8 Stores: (20÷8)7=17.5%

9 Stores: (20÷9)8=17.78%

10 Stores: (20÷10)9=18%

11 Stores: (20÷11)10=18.18%

<u>Founding Store Scenario 2- Referral Credit</u>

When a referral is ready to open you will start receiving 20% credit from their orders. If you have 2 stores the credit will show up in your Store 1 account.

Founding Store Scenario 3 Referral Store Opens Multiple Stores

Your referral opens a second store. Now the 10% referral credit will be split in addition to the 10% founding store credit. Now you'll receive a 15% credit when either store orders. Unlike scenario 1 this credit will decrease with each new store, but the volume will increase.

The referral stores are grouped by owner and not effect by other owner groups. Each owner is its own individual referral branch.

Referral Stores Formula: 10+(10÷Total # Referral Stores in owner group)

1 Referral Store: 10+10÷1=20%

2 Referral Stores: 10+10÷2=15%

3 Referral Stores: 10+10÷3=13.33%

4 Referral Stores: 10+10÷4=12.5%

5 Referral Stores: 10+10÷5=12%

6 Referral Stores: 10+10÷6=11.66%

7 Referral Stores: 10+10÷7=11.57%

8 Referral Stores: 10+10÷8=11.25%

9 Referral Stores: 10+10÷9=11.11%

10 Referral Stores: 10+10÷10=11%

11 Referral Stores: 10+10÷11=10.91%

<u>Founding Store Scenario 4</u>

If your referral store refers a new store owner and they open a store. You will now start receiving 10% from all of their orders.

How to redeem your credit

Credits will show up in your sunflora.org. You'll be able to use this on your next order just like cash! When placing the order your credit will show up as a payment method. To see all your credits sign into your account on aff.sunflora.org and go to "My Account."

Then go to "Credit History." If you have any questions on a credit please contact us at support@dang-designs.com.

<u>Wholesale Credit</u>

When you refer a wholesale account, you will receive 14% of their inventory purchases every time they order. Wholesale credits will remain 14%, regardless of the amount of wholesale accounts you refer. There are a few exceptions to this in regards to ancillary products, please contact your wholesale division for further information.

Joe Griffith: (727) 235-0720

Example: *When they spend $1000, you will receive $140.*

<u>Online Retail Credit</u>

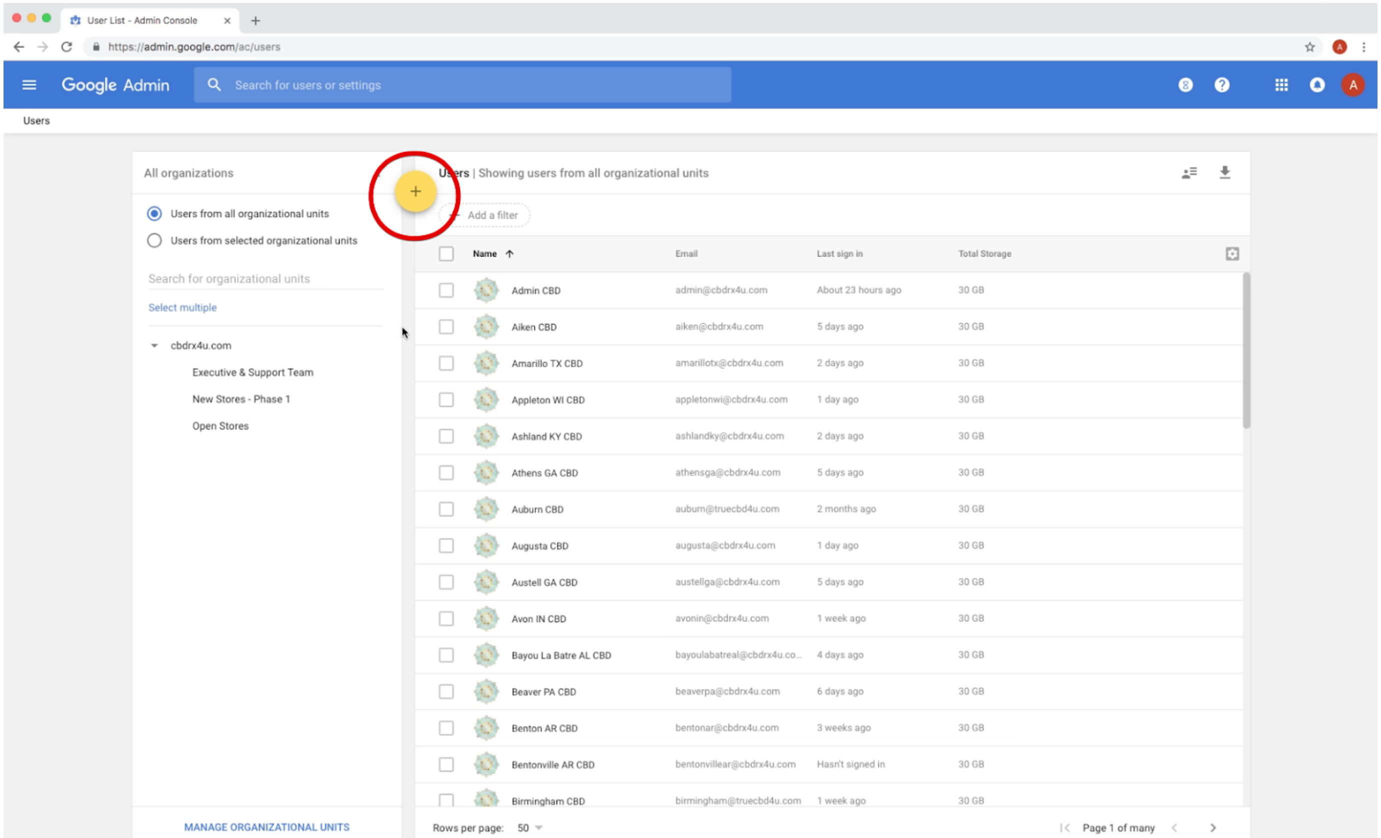
You may direct clients to your online store and still receive credit for their purchases; you will receive a 50% credit (minus any discounts) on their purchase when they order through the website.

Creating A Branded Email

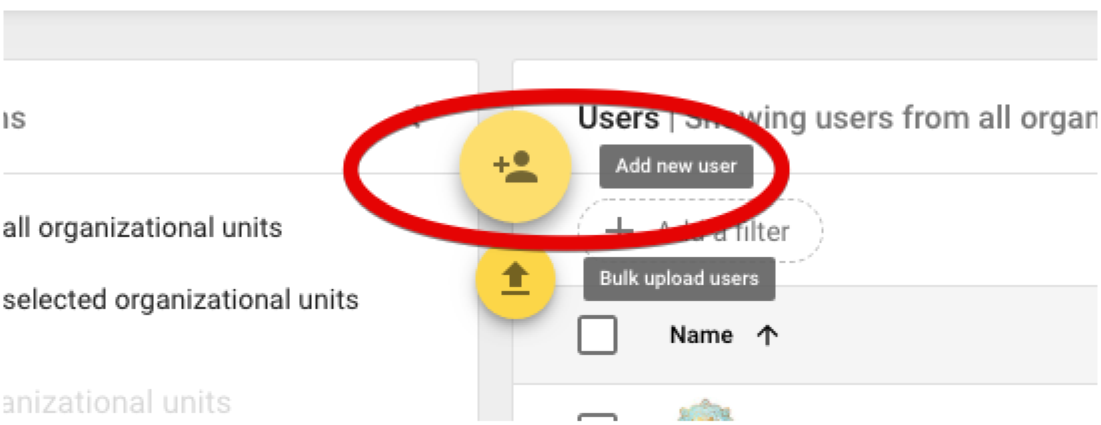
Must be created <u>before</u> the store phone call and walk-through.

1. Open paymoapp.com or copy and paste the URL into a new Chrome browser window or tab, and log in to your **Paymo** account.
2. Search for the **project** whose branded email you are going to create.
    a. Type the three-digit store number or store name in the top-left **Search** field in Paymo and select the <u>project</u>, not the client.
3. Click on the **Tasks** tab at the top of the page. One of the tasks under **Phase 1** will be **Branded Email**.
4. Open a new tab in your browser.
5. Log in to your official @dang-designs.com Google account, if you are not logged in already.
6. Open sheets.google.com or copy and paste it into the Chrome address bar.
7. Click on the **New CBD Store Owner Info (Responses)** workbook to open it.

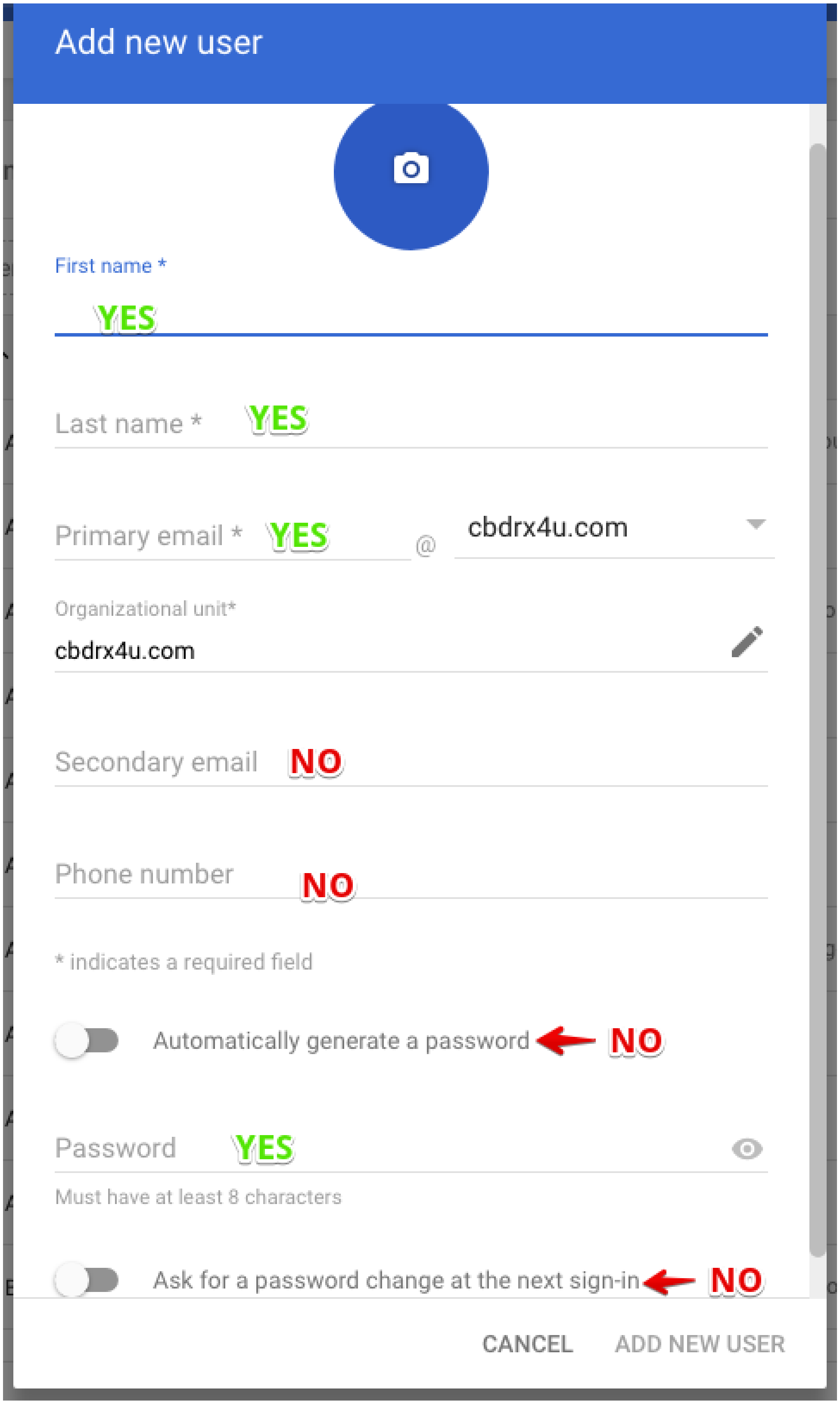
8. Click on your profile icon in the upper-right corner of the browser. Select the **CBD Admin** profile. (Note: This step is not demonstrated in the video. If you need help with it, please see **How to "Manage People" in Google Chrome** in the **Helpful Hints**, above.)
9. A new Google Chrome browser window will open.
10. Open admin.google.com or copy and paste it in the address bar and press return or enter on your keyboard.
11. Log in using the admin@cbdrx4u.com email and password. If you do not have the password, ask a colleague other than Dan for assistance.
12. Click **Users** on the left side of the **Google Admin Console** page.
13. Click on the yellow **+ add** button, located toward the top of the page.



1. Click **Add new user**.

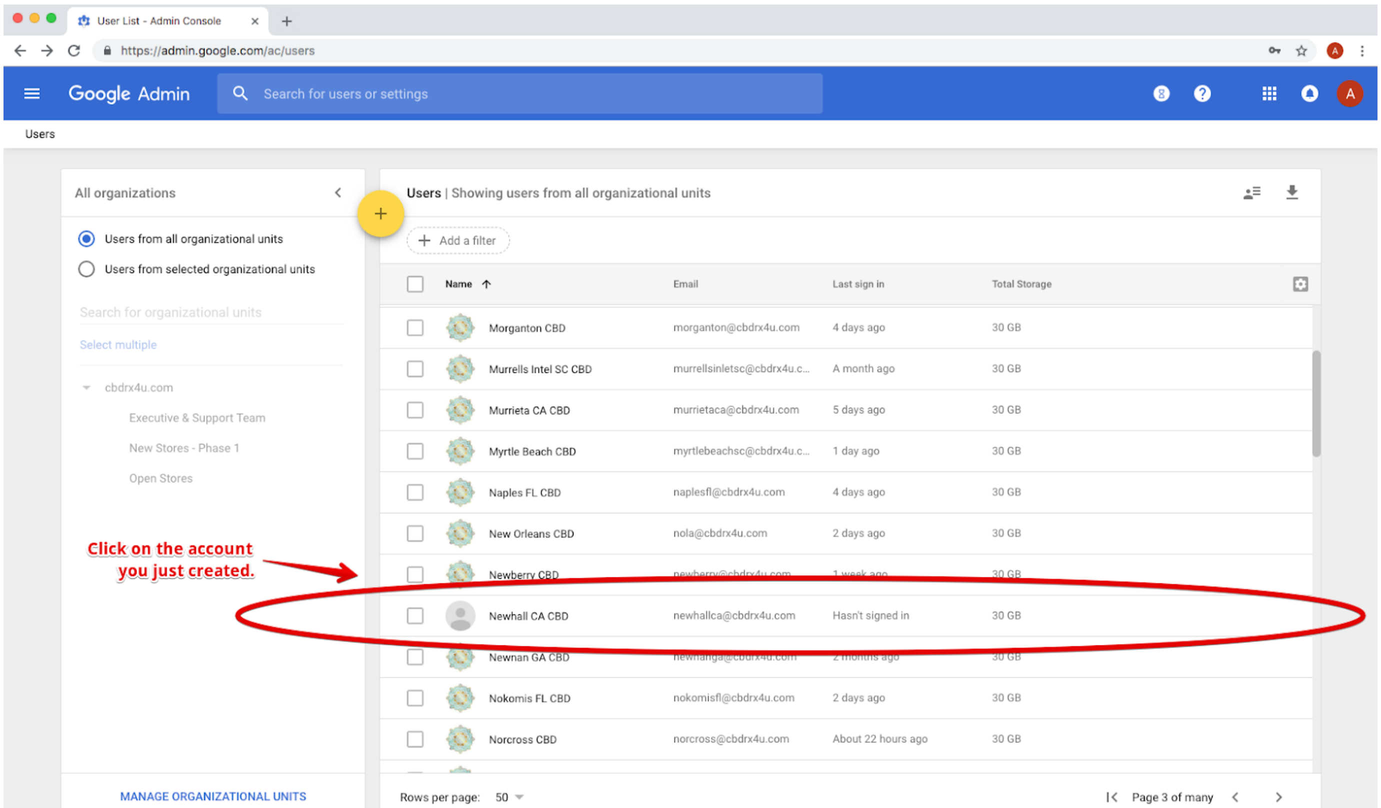


1. An **Add new user** window will appear on the screen. Complete the fields marked with a green YES in the screenshot below. (Note: These fields will not be marked on the live site.) The information needed to complete these fields is located in the **CBD Store** spreadsheet within the **New CBD Store Owner Info (Responses)** workbook.

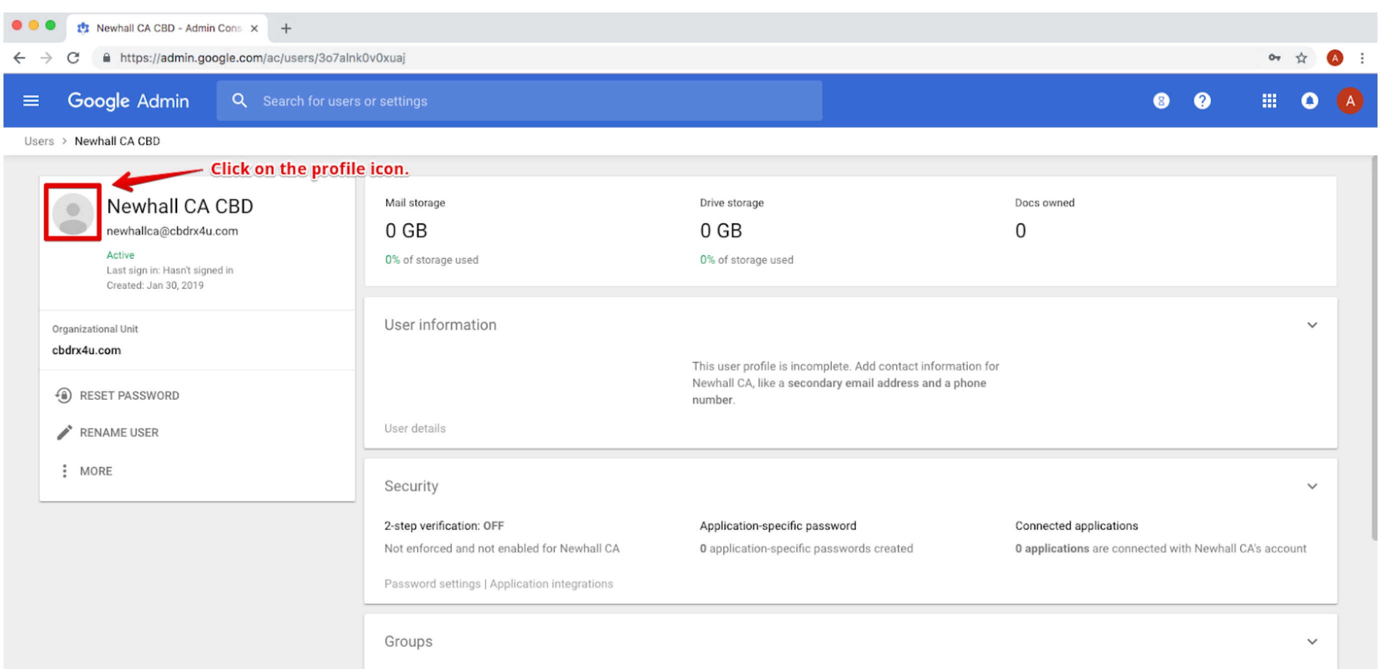


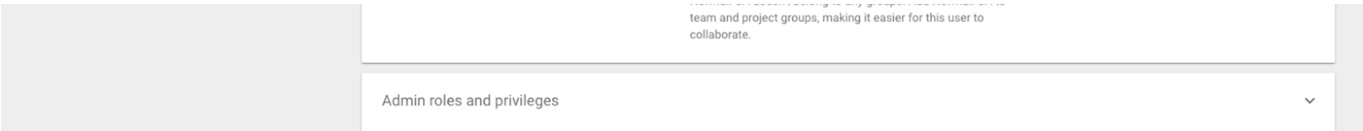
1. **First name**: This field consists of the entry from the **Location** column (with a capitalized first letter), followed by a space, followed by the state abbreviation (capitalized), with no comma in between. Example: "Corning NY". (Note: Sometimes the entries for a store's **Location** and **City** are different in the **CBD Store** spreadsheet. Always use the entry from the **Location** column when completing this field.)
2. **Last name**: This is the same for every store: "CBD" - all capitalized.
3. **Primary email:** This is the store's **Location** followed by their store number, all lowercase, and presented as one word. Example: "bouldercity0071".
   a. All email addresses will end in cbdrx4u.com except stores in Alabama. In the case of an Alabama store, click the down arrow next to "cbdrx4u.com" and change the email domain to "truecbd4u.com".

4. **Password**: We have a specific naming convention for creating passwords: the store's two-letter state abbreviation (lower case), followed by "CBD" (all capitalized), followed by the pound sign/hashtag (#), followed by the store's three-digit store number. For example, if Trussville, Alabama's store number is 198, "alCBD#198" would be their password.
5. Click the **ADD NEW USER** button, located at the bottom-right of the window.
6. A **New user added** window will appear. Click the **DONE** button in the lower left corner of the window to complete this step.
7. Using the scroll bar on the right side of the **Google Admin** page, scroll down until you come to the user account that you just created.
    a. If you don't see the account on a particular page, click the "next" arrow at the bottom-right of the window to access additional pages.
8. Click on the new user account to open the account's profile.

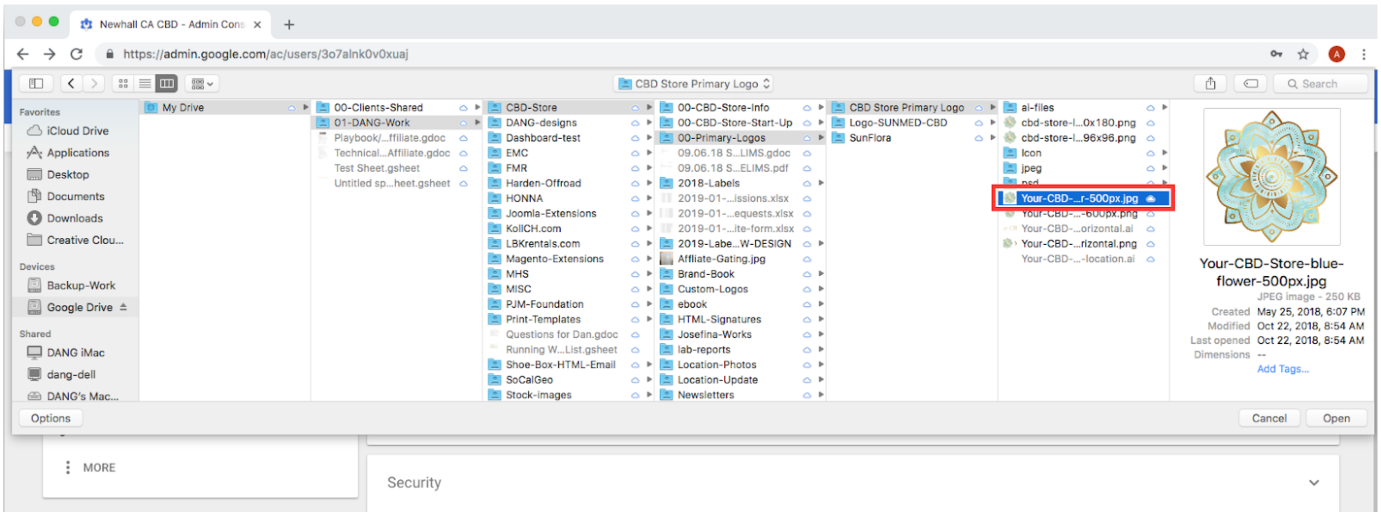


1. Click on the profile icon.

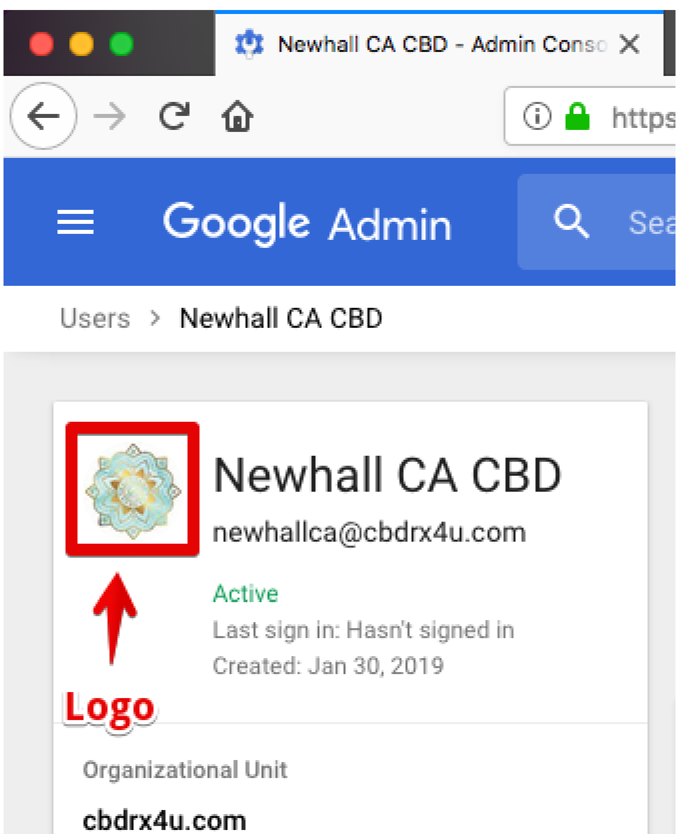


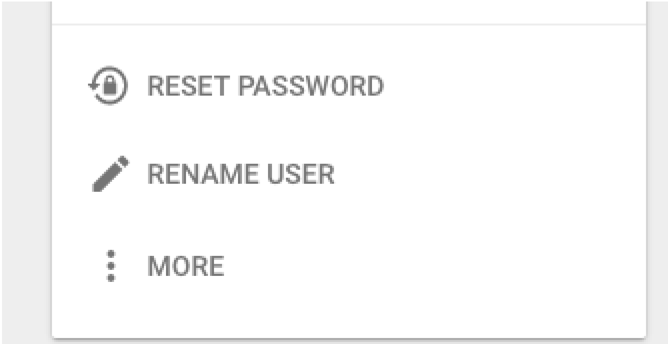


1. A small window will pop open and **Google Drive** should open automatically. If it doesn't open automatically, select Google Drive from the menu on the left.
2. Navigate through the following Google Drive folders to locate the primary Your CBD Store® logo: **My Drive 01-DANG-Work CBD-Store 00-Primary-Logos CBD Store Primary Logo**
    a. Select the **Your-CBD-Store-blue-flower-500px.jpg** file and click **Open**.

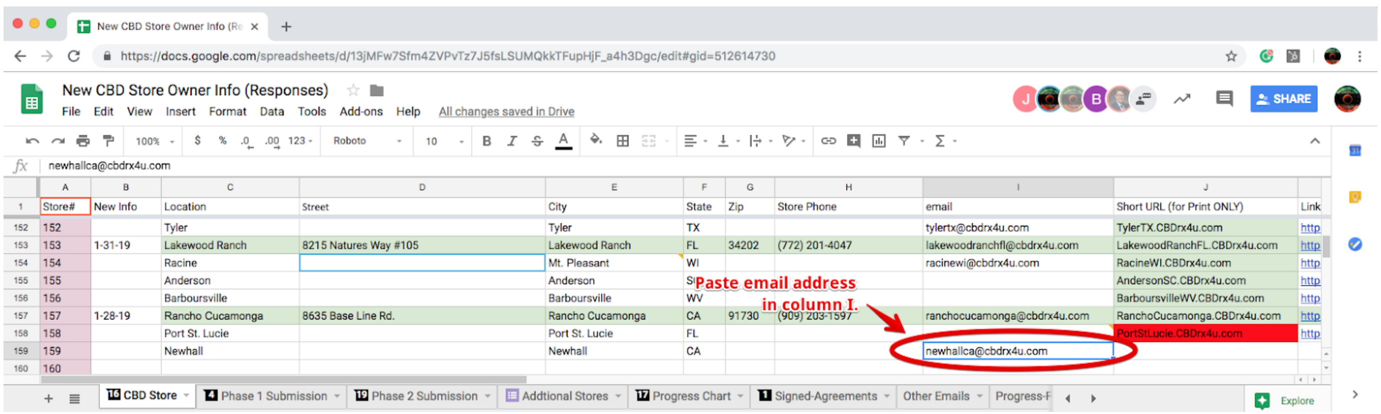


1. The Your CBD Store® logo should now appear in place of the blank profile picture for this user account.





1. Highlight and copy the email address you just created for this user account, located to the right of the logo.
2. Navigate back to the original Google Chrome browsing window where you are logged in to your official @dang-designs.com account.
    a. Return to the **New CBD Store Owner Info (Responses)** workbook and open the **CBD Store** spreadsheet.
    b. Scroll down to locate the store you're working with and paste this email address into the **Email** column.

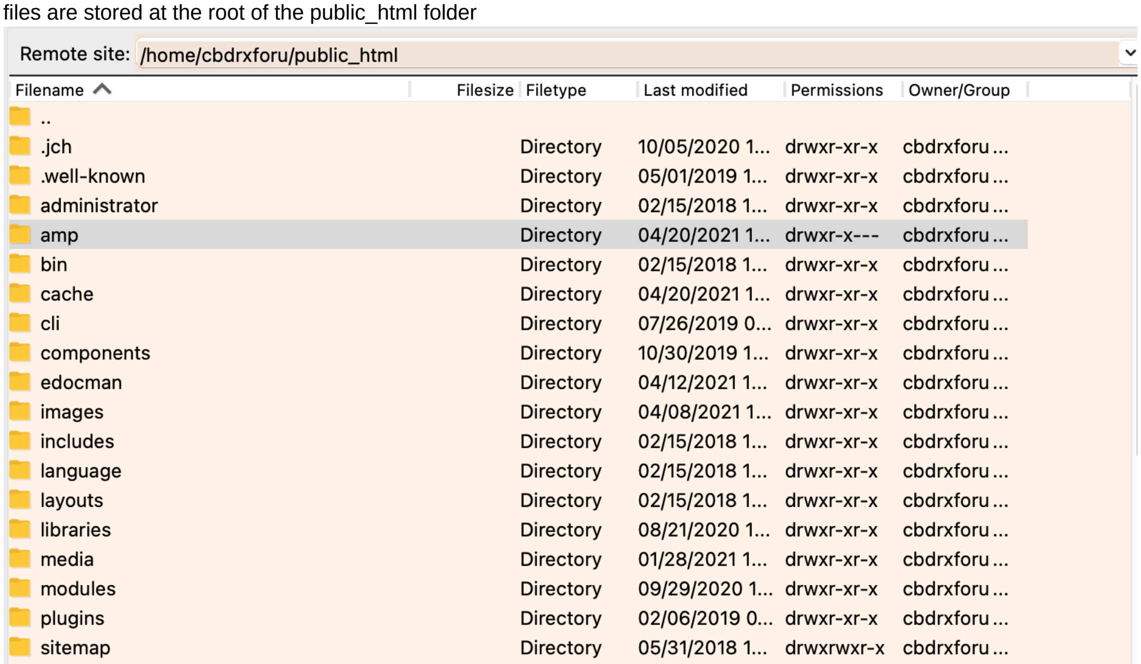


1. Return to the **Paymo** browser tab.
2. Move the task into "Review"
3. Make sure to add the email and password to CBD-Google/Owners-Usernames/Passwords

How To Redirect AMP Page For Closed Stores

This page will provide guidance on how to redirect traffic arriving at a closed store's AMP page (Mobile) so that the traffic is sent back to the find-us/state page.

1. **Get the name of the AMP file you have to update**
    a. All AMP files are stored at the root of the public_html folder
        i.

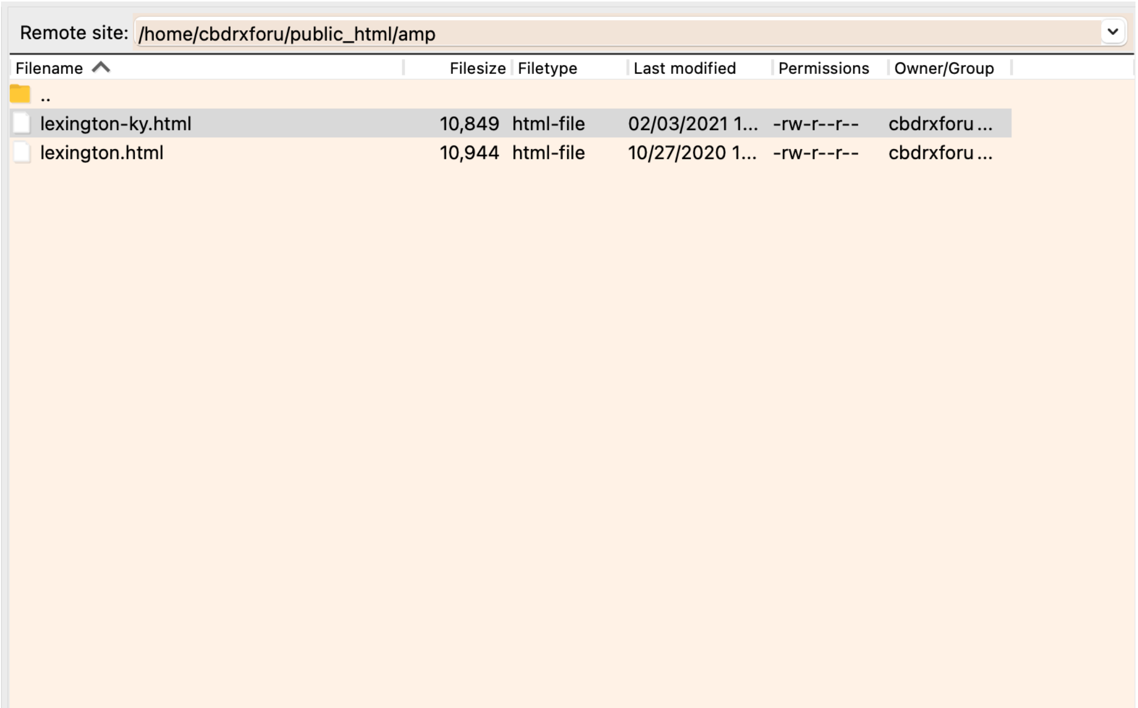


Remote site: /home/cbdrxforu/public_html

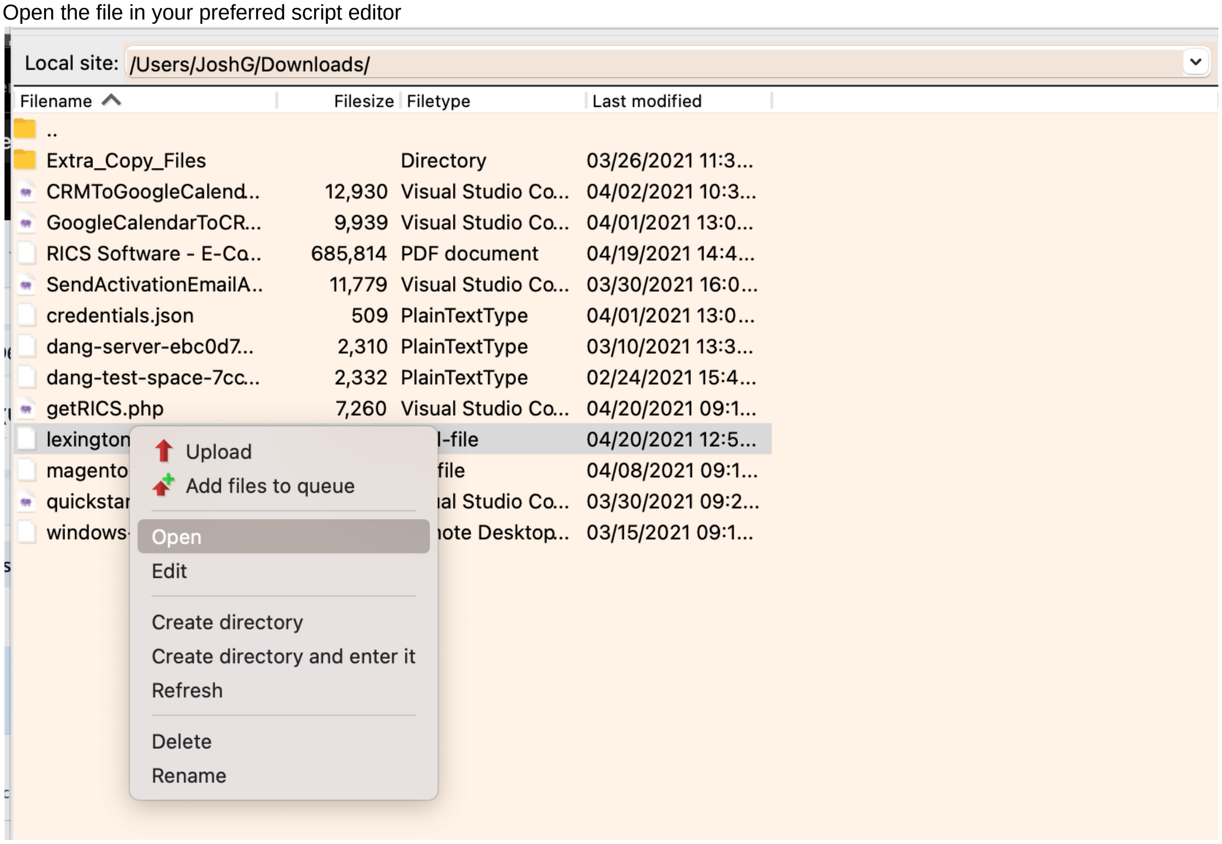
Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
.jch		Directory	10/05/2020 1...	drwxr-xr-x	cbdrxforu ...
.well-known		Directory	05/01/2019 1...	drwxr-xr-x	cbdrxforu ...
administrator		Directory	02/15/2018 1...	drwxr-xr-x	cbdrxforu ...
amp		Directory	04/20/2021 1...	drwx-x---	cbdrxforu ...
bin		Directory	02/15/2018 1...	drwxr-xr-x	cbdrxforu ...
cache		Directory	04/20/2021 1...	drwxr-xr-x	cbdrxforu ...
cli		Directory	07/26/2019 0...	drwxr-xr-x	cbdrxforu ...
components		Directory	10/30/2019 1...	drwxr-xr-x	cbdrxforu ...
edocman		Directory	04/12/2021 1...	drwxr-xr-x	cbdrxforu ...
images		Directory	04/08/2021 1...	drwxr-xr-x	cbdrxforu ...
includes		Directory	02/15/2018 1...	drwxr-xr-x	cbdrxforu ...
language		Directory	02/15/2018 1...	drwxr-xr-x	cbdrxforu ...
layouts		Directory	02/15/2018 1...	drwxr-xr-x	cbdrxforu ...
libraries		Directory	08/21/2020 1...	drwxr-xr-x	cbdrxforu ...
media		Directory	01/28/2021 1...	drwxr-xr-x	cbdrxforu ...
modules		Directory	09/29/2020 1...	drwxr-xr-x	cbdrxforu ...
plugins		Directory	02/06/2019 0...	drwxr-xr-x	cbdrxforu ...
sitemap		Directory	05/31/2018 1...	drwxrwxr-x	cbdrxforu ...
subdomains		Directory	04/20/2021 1...	drwxr-x---	cbdrxforu

ii. The names of the files in the folder are generally set as {cityname}-{state}.html

b. *For this example, we will use Lexington, KY
   i. AMP Page: https://cbdrx4u.com/amp/lexington-ky.html
   ii.

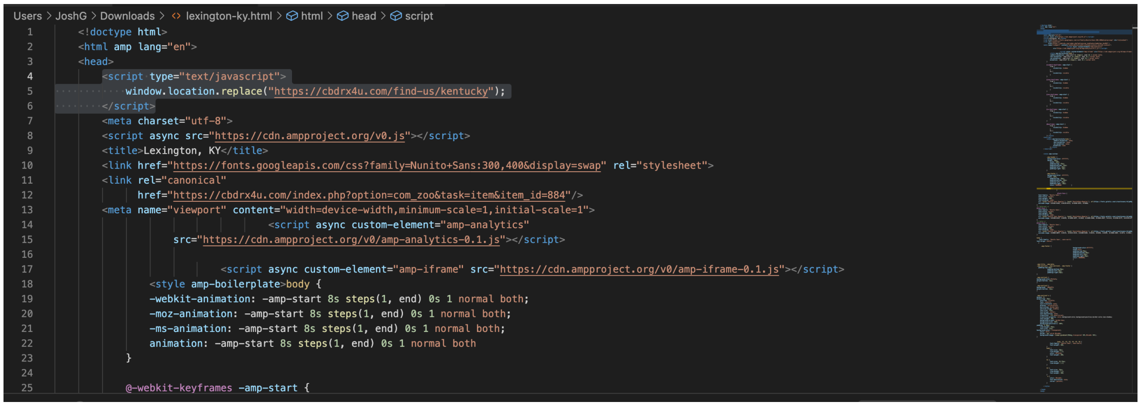


2. **Download the file from Filezilla to add the redirect script**
   a. *JavaScript is used here since the .htaccess file doesn't respond to the ruleset.
   b. Open the file in your preferred script editor
   c.



   d. Underneath the beginning of the <head> tag, add the following script:

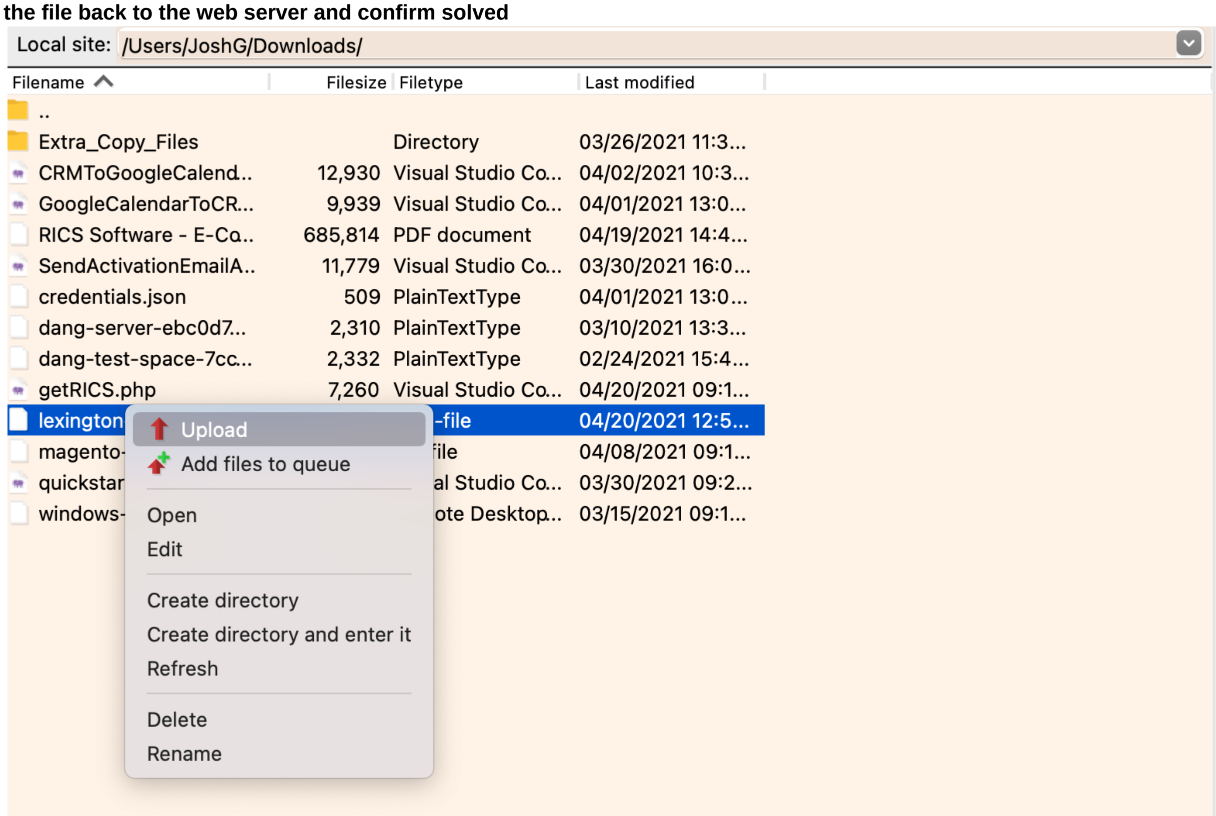
i.
```
<script type="text/javascript">
    window.location.replace("https://cbdrx4u.com/find-us
/kentucky");
</script>
```

ii. 

iii. **Make sure that you update the URL to the proper state's page! Best to pull the URL directly from the cbdrx4u website.**

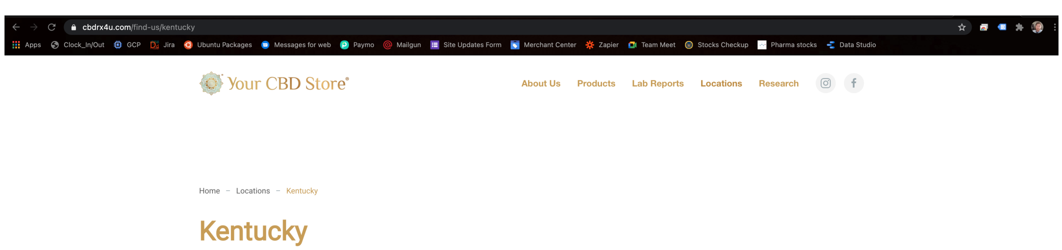
e. Save the file

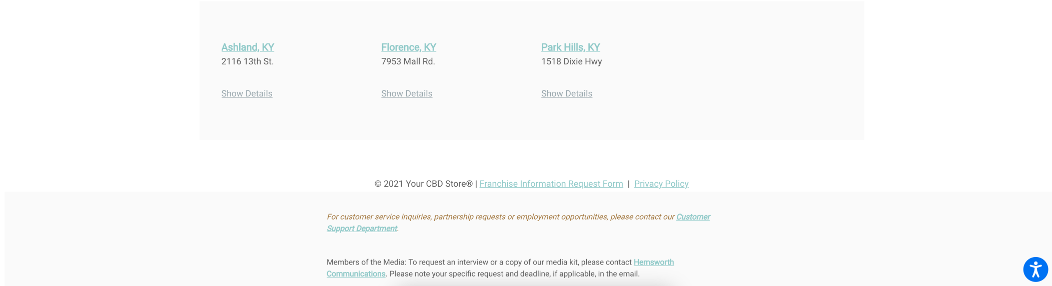
3. **Upload the file back to the web server and confirm solved**

a. 

b. Link to test: https://cbdrx4u.com/amp/lexington-ky.html

i. Results:

1. 



4. **You're Done!**

How To Set Up A 3PL Order Import Template

This page will host the process for setting up manual order imports for 3PL (http://secure-wms.com).

Example spreadsheet with instructions: https://docs.google.com/spreadsheets/d/1zYQbGrGA7t56tUkERNpZsVJbqme8qW2BxkOFq7YGQKg/edit#gid=38760320

1. **Enter all the required fields with data from Fishbowl/Magento**
    a. ReferenceNumber, ShipCarrier, ShipService, ShipTo Name, ShipToCompany, ShipToAddress1, ShipToCity, ShipToState, ShipToZip, ShipToCountry, SKU, Quantity
        i. ShipCarrier and ShipService data is found within the Sales Orders on Fishbowl
        ii. ShipToCountry must be in 'US' format (no (United States) portion is required
2. **Download the file as Tab-separated Values (.tsv)**
3. **Remove the headers from the page in a text editor**
    a. The entire row can be removed prior to download but it makes it harder to view after download
4. **Navigate to the (**https://secure-wms.com/smartui/?page=findorders **) site and select the green button 'Import'**
5. **Set the customer as 'Sunflora' which should default to the Fullfillment/Logistics warehouse**
6. **Select the checkbox that says 'View list of orders after file validation'**
7. **Press 'Select File' and choose your file previously downloaded and updated**
    a. Remember: the header row needs to be removed before submission or errors will occur.
8. **Press the 'Import' button after you products are showing in the list**
9. **Confirm the records came over correctly by navigating through the find orders by Reference Number**
10. **You're Done!**

How To Set Up A New Magento Website Domain

This page will reference the https://cs.sunflora.org site that was created recently to address orders that needed to be resent to customers due to being either damaged or less-than proper quality for the affiliates respective customers. Therefore, the example provided below has a few steps that were able to be skipped, such as updating the XML document to reference the use of a new theme, among several other steps for a brand new Magento website.

1. **Add the new domain to the CPanel for the site**
    a. Navigate to the CPanel for sunflora and go the icon listed as "Domains"
    b. On the top-right, select the option for "Create a New Domain"
    c. Add the new domain in the Domain field
        i. In this instance, we used cs.sunflora.org
    d. A Document Root location will then be displayed; remove the new domain name from the path so a new folder is not created and the path matches the other sites.
    e. Press Submit
2. **Update the site to use SSL**
    a. While still in the CPanel, navigate to SSL/TLS icon
    b. Select "Manage SSL sites" listed on the middle-right side
    c. Scroll to the new domain and confirm it looks similar to this:
        i. 

    d. If not, install the SSL using the options on the right-hand side.
3. **Create a new theme folder to use for the App and Skin directories**
    a. In this instance, we are reusing an existing theme so the new themes only need to be copied and pasted (while being renamed).
    b. Navigate to the file manager in CPanel
    c. First go to the directory listed below and copy the folder over with a new name for the new theme (in this case we used theme768-customer-service)
        i. public_html/app/design/frontend/tm_themes/
    d. Next go to the directory listed below and perform the same process as mentioned above
        i. public_html/skin/frontend/tm_themes/

4. **Create the website, store, and store view in Magento**
    a. Log into the Magento site as an admin user and navigate to System > Manage Stores
        i. Add the website
        ii. Add the store
        iii. Add the store view
    b. Navigate on the top menu to System > Design
        i. Update the theme folder used by the new website created
    c. Navigate to System > Configuration > Web
        i. Update the default store to your newly created website
        ii. Update the base URL from https://sunflora.org to https://cs.sunflora.org
    d. Add the new store to the index.php at the root of the file directory
        i. A new case block should be added that looks similar to this:

```
case 'cs.sunflora.org':
        $mageRunCode = 'customerservice';
        $mageRunType = 'website';
break;
```

    e. Navigate to System > Itoris Extensions > Store Login Control
        i. Update the new website to NOT allow registration and to require users to login
    f. Update the itoris if/else statement to include the new store to check for
        i. Found in app/design/frontend/base/default/template/itoris/storelogincontrol/login.phtml
        ii. The new else-if block should look something like this, but make sure to update verbiage throughout as needed:

```
else if($siteid ==11) {?>

<!--[if lt IE 9]>
<link rel="stylesheet" type="text/css" href="<?php
echo Mage::helper('core/js')->getJsUrl('itoris
/storelogincontrol/css/login_ie.css'); ?>" />
<![endif]-->
<div class="white_background" style="background-
image:url('/media/loginbgimage.jpg');
        background-repeat: no-repeat;
        background-size: cover; ">


        <div id="itoris_login_box" class="
login_form" style="opacity: 0.85;
 filter: alpha(opacity=85);">


                <div class="color-box" style="
height: 60px; padding: 10px"><h3>To Access your
Ordering Portal, please Log-in below</h3></div>
                <table width="100%" align="center"
style="background-color: #fff; border-left: 1px
solid #999; border-right: 1px solid #999">

                <tr>
                        <td align="center" valign="
bottom" colspan="2"><img src="/skin/frontend
/tm_themes/theme768-ycbds-b2b/images/YCBD-Logo-
Horizontal.png"  >
```

```
                        <h2>Customer Service
Ordering Portal</h2></td>

                </tr>
                <tr>

                        <td colspan="2" align="
center">
                        Welcome to your new home for
ordering for your store.  Please use the same login
you've used previously.  Be sure to Bookmark this
page.
                        </td>

                        </tr></table>
                <!-- <div class="border_t"></div>
                <div class="border_r"></div>
                <div class="border_b"></div>
                <div class="border_l"></div>
                <div class="corner_rt"></div>
                <div class="corner_lt"></div>
                <div class="corner_rb"></div>
                <div class="corner_lb"></div> -->
                <div id="itoris_login" class="box"
style="background-color: #fff">

                </div>
                <!-- <div id="store_switcher" class="
box"></div> -->
                <div class="color-box-bottom" style="
height: 80px; padding: 10px; background-color:
#ff0000">This portal is for Customer Service
Ordering only.  If you are not a Customer Service
Member, please click here: <a href="https://sunflora.
org">Sunflora</a></div>
        </div>
</div>
<?php }
```

iii. Make sure to update the site ID variable based on the core_website table number in the database as well

5. Add a new customer group

Code Pieces

This page hosts the groups of files stored from custom website coding pieces. They are separated based on their language used while prefixed with the name of the website for clarity's sake.

- GAS Projects
- Java Selenium Automation Projects
- PHP Projects
- JavaScript Projects
- WordPress Projects

GAS Projects

This will be the host for all Google Apps Script projects and the code related to them.

Convert GMT Column To EST

This page will host an array-converting script that takes GMT time/date values and converts them to EST.

The Code

```
function convertToEST(){
  var logsSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Missed_Calls")
  var testSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("testing")
  var gmtDates = logsSheet.getRange("J2:J").getValues()

  var array = []
  var gmtstring = ""
  var datestring = ""
  var period = ""

  var hour
  var minutes
  var year
  var month
  var day

  // For each value in the array, convert them to a date object.
  for(var i = 0; i < gmtDates.length; i++){

    gmtstring = gmtDates[i].toString()

    // get the time of the object and set the remainder for the date.
    if(gmtstring.indexOf("pm") != -1){
      hour = parseInt(gmtstring.substring(0,2)) +12
      minutes = parseInt(gmtstring.substring(3,6))
      gmtstring = gmtstring.substring(gmtstring.indexOf("pm")+3,
gmtstring.length)
    }
    else if(gmtstring.indexOf("am") != -1){
      hour = parseInt(gmtstring.substring(0,2))
      minutes = parseInt(gmtstring.substring(3,6))
      gmtstring = gmtstring.substring(gmtstring.indexOf("pm")+3,
gmtstring.length)
    }
```

```
    // Get the year, month, and day from the object.
    year = parseInt(gmtstring.substring(0, 4))
    month = parseInt(gmtstring.substring(5, 8)) - 1
    day = parseInt(gmtstring.substring(8, 11))

    // Set up the new date object
    var gmtdate = new Date(year, month, day, hour, minutes, 0, 0)
    gmtdate.setHours(gmtdate.getHours() - 5)

    // Set the period to add to the date.
    if(parseInt(gmtdate.getHours().toString()) >= 12){
      period = "PM"
    }
    else{
      period = "AM"
    }

    // Set the datestring to be displayed in a user friendly way.
    datestring = (gmtdate.getHours().toString() + ":" + gmtdate.
getUTCMinutes().toString() + period + " " + gmtdate.getFullYear().
toString() + "-" +
     (parseInt(gmtdate.getMonth().toString())+1) + "-" + gmtdate.
getDate().toString())

    array.push(datestring)
  }

  return array
}
```

Get Web Dev Measure Data From Lighthouse Client

This script is used on the DANG-Sites Google Sheet to obtain the information generated from measuring each individual site. This process is automated through a Java Selenium Project as well.

Setting Up Web Dev Website Testing Automation

```
/*************************************************************************
****
 * Created by Joshua Greenert on 7/30
/2020                                 *

*************************************************************************
***
 *                                            DESCRIPTION
 * This script will run to open a webpage, input the URL of the site,
press the enter key
 * and get the results from the search to store in each respective
area.  Found that the audit on
 * the site must be ran in order for it to register.
```

```
************************************************************************
**************************
 *                                              PARAMETERS
 * URL - The URL of the website to update the content for.

************************************************************************
**************************
 */
function getWebDevData(userurl) {

  // Set the URL object for testing
  var urlAddress = userurl

  var array = []

  // Set the URL to fetch and fetch it.  Then parse the JSON object and
set it as a string.
  var url = 'https:\/\/lighthouse-dot-webdotdevsite.appspot.com\/\/lh\
/reports?url=' +urlAddress;
  var response = UrlFetchApp.fetch(url);

  var obj = JSON.parse(response.getContentText())
  var str = JSON.stringify(obj, ['lhrSlim', 'score'])

  // cycle through the string and add the scores to the array.
  for(var i = 0; i < 4; i++){

    if(str.indexOf("score") != -1){
      var testscore = str.substring(str.indexOf("score\":")+7, str.
indexOf("}"))

      // check if score == 1, set to 100
      if(testscore == 1){
        testscore = 100.0
      }
      else{
        testscore = testscore * 100;
      }

      // Add the reference and update the string.
      array.push(testscore)
      str = str.substring(str.indexOf("}")+1, str.length)
    }
  }

  //Now put the data in the sheet
  return array
}
```

OnChange GMT To EST Row Conversion

Similar to the GMT to EST conversion array script, this script will take a date/time value and convert it to EST. However, this function is set with an override to the onChange function within Google Apps Script to generate the EST row when a new row is inserted (in this case, by the zap).

IMPORTANT **When using a function that creates triggers, a function MUST be created to delete the triggered being created or the list will reach its maximum forcing emails to be sent that the new trigger could not be created.**

The Code

```
/*
 * Created by Joshua Greenert on 1/15/2020
 *
 * This function will add a new row with the updated EST time from the
GMT time
 * presented within column J whenever a new row is inserted to the
sheet by zapier.
 */
function onChange(e){
  var sheet = SpreadsheetApp.getActive();
  ScriptApp.newTrigger("updateRowESTTime")
    .forSpreadsheet(sheet)
    .onChange()
    .create();
}

function updateRowESTTime(e) {
    var sh = SpreadsheetApp.getActiveSheet();

    // Once a new row is added, get the row and set the object to
    if(e.changeType === 'INSERT_ROW') {
      var row = sh.getActiveRange().getRow();
      var cellstring = "J" + row
      var gmtDates = sh.getRange(cellstring).getValue()

      var gmtstring = ""
      var datestring = ""
      var period = ""

      var hour
      var minutes
      var year
      var month
      var day

      gmtstring = gmtDates.toString()

      // get the time of the object and set the remainder for the date.
      if(gmtstring.indexOf("pm") != -1){
        hour = parseInt(gmtstring.substring(0,2)) +12
        minutes = parseInt(gmtstring.substring(3,6))
```

```
        gmtstring = gmtstring.substring(gmtstring.indexOf("pm")+3,
gmtstring.length)
      }
      else if(gmtstring.indexOf("am") != -1){
        hour = parseInt(gmtstring.substring(0,2))
        minutes = parseInt(gmtstring.substring(3,6))
        gmtstring = gmtstring.substring(gmtstring.indexOf("pm")+3,
gmtstring.length)
      }

      // Get the year, month, and day from the object.
      year = parseInt(gmtstring.substring(0, 4))
      month = parseInt(gmtstring.substring(5, 8)) - 1
      day = parseInt(gmtstring.substring(8, 11))

      // Set up the new date object
      var gmtdate = new Date(year, month, day, hour, minutes, 0, 0)
      gmtdate.setHours(gmtdate.getHours() - 5)

      // Set the period to add to the date.
      if(parseInt(gmtdate.getHours().toString()) >= 12){
        period = "PM"
      }
      else{
        period = "AM"
      }

      // Get the minutes and correct as needed.
      if(gmtdate.getUTCMinutes().toString().length == 1){
        minutes = "0" + gmtdate.getUTCMinutes().toString()
      }
      else{
        minutes = gmtdate.getUTCMinutes().toString()
      }

      // Set the datestring to be displayed in a user friendly way.
      datestring = (gmtdate.getHours().toString() + ":" + minutes +
period + " " + gmtdate.getFullYear().toString() + "-" +
      (parseInt(gmtdate.getMonth().toString())+1) + "-" + gmtdate.
getDate().toString())

      sh.getRange(row, 11).setValue(datestring)
  }

  deleteProjectTriggers()
}

function deleteProjectTriggers(){
    // do some stuff here then stop the trigger(s) by name
    var triggers = getProjectTriggersByName('updateRowESTTime');
```

```
    for (var i = 0; i < triggers.length; ++i)
        ScriptApp.deleteTrigger(triggers[i]);
}

function getProjectTriggersByName(name) {
    return ScriptApp.getProjectTriggers().filter(
        function(s) {return s.getHandlerFunction() === name;}
    );
}
```

Query Filter Build From Zapier Imports (Harden-Offroad)

This page holds the code used in automating the form data within harden-offroad. As Zapier is another product of different design, I will also be detailing steps to navigate that configuration in a separate location.

The Code

```
/*
 * Created by Joshua Greenert on 12/15/2020
 *
 * This program will create a list of registrants along with all of the
details requested
 * through the use of automation zapier actions.  The tables are pulled
into the Events,
 * Field_Values, and Registrants sheets and their ids are used to
reference their data.
 */

function QueryFilter(event_id) {

  // Get the contents from the cell.
  var registSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Registrants")
  var eventSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Events")
  var valuesSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Field_Values")

  var registrantsRange = registSheet.getRange("A2:BO").getValues()
  var valuesRange = valuesSheet.getRange("B2:D").getValues()
  var eventNameRange = eventSheet.getRange("A2:E6").getValues()

  var event_users_array = []
  var sortedArray = []

  var eventName = ""

  // Set the event name from the data
  for(var i = 0; i < eventNameRange.length; i++){
    if(eventNameRange[i][0] == event_id){
```

```
      eventName = eventNameRange[i][4].toString()
    }
  }

  // Use for loop for multi dimensional array to return values specific
to the user referenced.
  // [id, firstname, lastname, address, city, state, zip, phone, email,
totalamount, registerdate]
  for(var i = 0; i < registrantsRange.length; i++){
    for(var j = 1; j < 2; j++){
      if(registrantsRange[i][j] == event_id){
        event_users_array.push([registrantsRange[i][20], eventName,
registrantsRange[i][4], registrantsRange[i][5], registrantsRange[i][7],
registrantsRange[i][9], registrantsRange[i][10], registrantsRange[i]
[12].toString(),
                              registrantsRange[i][13].toString(),
registrantsRange[i][15], parseInt(registrantsRange[i][17]),
registrantsRange[i][0]])
      }
    }
  }

  // Sort the array by the reg date.
  sortedArray = ArrayLib.sort(event_users_array, 0, false)
  return sortedArray
}
```

Send Email When Plugins Expire

This page will host the code for setting up a plugin expiration notification email. A trigger is set for this to be a daily notification so that users will be reminded to address the situation until it is either updated, or it is stripped from the list to check for.

The Code

```
/*
 * Created on 6/22/2021 by Joshua Greenert
 *
 * This function will check the list of plugins to find ones that are
10 days away from
 * expiring to be renewed.  Then a list is compiled and an email is
sent out with all of
 * the sites needed to be updated.
 */

function sendEmailPlugins() {

  // Set the new ranges to get the data from the cells.
  var rangeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Plugins")
  var pluginData = rangeSheet.getRange("A5:G40").getValues()
```

```
  // Variables
  var pluginArray = []
  var msg = ""

  // Loop through the days left to find out what items need to be added
to the plugin array
  for(var i = 0; i < pluginData.length; i++){
    if(pluginData[i][6] < 10){
      pluginArray.push([pluginData[i][0], pluginData[i][1], pluginData
[i][2], pluginData[i][3], pluginData[i][6]])
    }
  }

  // Set the message to include the plugins and display them to the
user.
  if(pluginArray === undefined || pluginArray.length == 0) {
    Logger.log("No plugins have expired!")
  }
  else{

    msg = "<html><body><table style='width:25%; text-align:
center;'><tr><th>Plugin Name</th><th>Price</th> <th>CMS<
/th><th>Provider</th><th>Days Left</th></tr>"

    for(var i = 0; i < pluginArray.length; i++){
      msg += "<tr>"
      msg += "<td>" + pluginArray[i][0] + "</td>"
      msg += "<td>" + pluginArray[i][1] + "</td>"
      msg += "<td>" + pluginArray[i][2] + "</td>"
      msg += "<td>" + pluginArray[i][3] + "</td>"
      msg += "<td>" + pluginArray[i][4] + "</td>"
      msg += "</tr>"
    }

    msg += "</table></body></html>"

    MailApp.sendEmail({
    to: "dev@dang-designs.com",
    subject: "Plugins Expiring Soon:",
    htmlBody: msg,
    });
  }
}
```

Sunflora - Group Stores With Primary Parent

This page hosts the script used to consolidate stores with the primary parent grouped by Group ID (arbitrary). The end result produces their group id, customer id, and primary email associated with the customer id.

The Code

```
/*
 * Created by Joshua Greenert on 5/27/2021
 *
 * This function will create a list of ownergroups for each owner to
show a new row
 * for each of their respective child stores.  Primary store owners can
have two potential
 * emails to be associated due to the stores being closed which will be
shown as referenced
 * to the open store as a new row with the same ownergroupid.
 *
 * Fields needed: ownergroupid, customerid, customeremail
 */
function getOwnerGroups() {
  // Set the range sheet to use for all respective ranges.
  var rangeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("data")

  // Set the range columns.
  var ownergroupids = rangeSheet.getRange("A2:A").getValues()
  var ownercustomerid = rangeSheet.getRange("B2:B").getValues()
  var primarystore = rangeSheet.getRange("C2:C").getValues()
  var ownerstores = rangeSheet.getRange("D2:D").getValues()
  var ownerhistoricalstores = rangeSheet.getRange("E2:E").getValues()

  // Variables needed for operation.
  var grouparray = []

  var customerid = ""
  var customeremail = ""

  // For each row, get the information from the store and historical
store rows and set
  // them with the same ownergroupid.
  for(var i = 0; i < ownergroupids.length; i++){

    // first set the main store by setting an array of the stores if
there are multiple and set the
    // other arrays for future use.
    primarystoreArray = primarystore[i].toString().split(" ")
    primarystoreemail = ""
    secondprimarycustid = ""

    // Set to the .com so we can isolate the point where they should be
split; add the .com back in later.
    ownerstoresArray = ownerstores[i].toString().split(".com")
    ownerhistoricalstoresArray = ownerhistoricalstores[i].toString().
split(".com")
```

```
    if(primarystoreArray.length > 1){
      for(var j = 0; j < primarystoreArray.length; j++){
        if(j == 0){
          primarystoreemail = primarystoreArray[j].substring(0,
primarystoreArray[j].indexOf("(")-1)
          secondprimarycustid = primarystoreArray[j].substring
(primarystoreArray[j].indexOf("(") +1, primarystoreArray[j].indexOf
(")"))
          grouparray.push([ownergroupids[i].toString(), ownercustomerid
[i].toString(), primarystoreemail.toString()])
        }
        else{
          grouparray.push([ownergroupids[i].toString(),
secondprimarycustid.toString(), primarystoreArray[j].toString()])
        }
      }
    }
    else{
      grouparray.push([ownergroupids[i].toString(), ownercustomerid[i].
toString(), primarystore[i].toString()])
    }

    // Get the owner's stores and set them to the main owner's
information using for loop.
    for(var j = 0; j < ownerstoresArray.length; j++){
      if(j == 0){
        customerid = ownerstoresArray[j].toString().substring(1,
ownerstoresArray[j].toString().indexOf(")"))
      }
      else{
        customerid = ownerstoresArray[j].toString().substring(2,
ownerstoresArray[j].toString().indexOf(")"))
      }
      customeremail = ownerstoresArray[j].toString().substring
(ownerstoresArray[j].toString().indexOf(" ")+1,ownerstoresArray[j].
toString().length ) +".com"

      if(ownerstoresArray[j].toString() == ""){
        break
      }
      grouparray.push([ownergroupids[i].toString(), customerid,
customeremail])

    }

    // Get the owner's historical stores and set them to the main
owner's information using for loop.
    for(var k = 0; k < ownerhistoricalstoresArray.length; k++){
      if(k == 0){
        customerid = ownerhistoricalstoresArray[k].toString().substring
```

```
(1,ownerhistoricalstoresArray[k].toString().indexOf(")"))
      }
      else{
        customerid = ownerhistoricalstoresArray[k].toString().substring
(2,ownerhistoricalstoresArray[k].toString().indexOf(")"))
      }
      customeremail = ownerhistoricalstoresArray[k].toString().substring
(ownerhistoricalstoresArray[k].toString().indexOf(" ")+1,
ownerhistoricalstoresArray[k].toString().length )+".com"

      if(ownerhistoricalstoresArray[k].toString() == ""){
        break
      }
      grouparray.push([ownergroupids[i].toString(), customerid,
customeremail])

    }

  }

return grouparray
}
```

Watchful List Data Separator (Multiple Sites)

This script pulls the data provided from the watchful report into separate fields on the Reports and Definitions Google Sheet. This is useful for setting up reporting for clients where only their information is shown and segmented with what we want them to see.

```
/*
 * Created by Joshua Greenert on 9/16/2020
 *
 * This script will separate the site, date, and plugin update

                              PSEUDOCODE
 Find indexOf Activity Logs
        grab website and end at forward slash(/)
Start where forward slash ends
        grab date and end at indexOf m
Start where m ends
        grab update data where month begins
                month variable will need to be specified in 01, 10, 12
format

 */

function SiteChangeList(index1, index2) {
  var dataSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Multiple_Input")
```

```
  // Add the ranges to pull data from.
  var dataArrayRange = dataSheet.getRange("A1:A").getValues()
  var dataRange = []

  // Variables to hold the data.
  var siteRange = []
  var dateRange = []
  var pluginRange = []
  var finalArray = []

  var site = ""
  var date = ""
  var plugin = ""
  var updatestr = ""

  var plugstr = 0
  var datelen = 0
  var pluginlen = 0

  var month


  // Set all pieces within the given array to strings.
  for(var i = 0; i < dataArrayRange.length; i++){
    dataRange.push(dataArrayRange[i][0].toString())
  }

  // Set each site, date, and plugin through a for loop.
  for(var i = 0; i < dataRange.length; i++){

    // Set the site for each record, the date, and the plugin data.
    site = dataRange[i].substring(13, dataRange[i].indexOf("/"))

    /* This is to catch the 360twin issue and will need to be updated
every month that the report is ran.*/
    if(dataRange[i].indexOf("360twin") == -1){
      dataRange[i] = dataRange[i].substring(dataRange[i].indexOf
(index1), dataRange[i].length)
    }
    else{
      dataRange[i] = dataRange[i].substring(dataRange[i].indexOf
(index2), dataRange[i].length)
    }

    // For loop through each record after the first to ensure that the
data is grabbed.
    for(var j = 0; j < dataRange[i].length; j++){

      if((dataRange[i].indexOf("am") != -1) || (dataRange[i].indexOf
("pm") != -1)){
```

```
        date = dataRange[i].substring(0, dataRange[i].indexOf("m") +1)
        Logger.log(date)
        month = date.substring(0, 2)
        datelen = date.length
        date = new Date(Date.parse(date.substring(0, 10)))

        // Update the dataRange again to strip the data based on the
end of the date.
        dataRange[i] = dataRange[i].substring(datelen, dataRange[i].
length)

        if(dataRange[i].indexOf(month) == -1){
          plugin = (dataRange[i].indexOf("Page") != -1) ? dataRange[i].
substring(0, dataRange[i].indexOf("Page")) : dataRange[i].substring(0,
dataRange[i].length)
          pluginlen = plugin.length
        }
        else{
          plugin = dataRange[i].substring(0, dataRange[i].indexOf(month
+ "/"))
          pluginlen = plugin.length
        }
        dataRange[i] = dataRange[i].substring(pluginlen, dataRange[i].
length)

        // Separate the plugin from the modification.
        if(plugin.indexOf("updated") != -1){
          plugstr = plugin.substring(0, plugin.indexOf("has been
updated")-1)
          updatestr = plugin.substring(plugin.indexOf("updated"),
plugin.length)
          updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
        }
        else if(plugin.indexOf("installed") != -1){
          plugstr = plugin.substring(0, plugin.indexOf("has been
installed")-1)
          updatestr = plugin.substring(plugin.indexOf("installed"),
plugin.length)
          updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
        }
        else if(plugin.indexOf("modified") != -1){
          plugstr = plugin.substring(0, plugin.indexOf("has been
modified")-1)
          updatestr = plugin.substring(plugin.indexOf("modified"),
plugin.length)
          updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
        }
```

```
        else if(plugin.indexOf("deleted") != -1){
          plugstr = plugin.substring(0, plugin.indexOf("has been
deleted")-1)
          updatestr = plugin.substring(plugin.indexOf("deleted"),
plugin.length)
          updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
        }
        else if(plugin.indexOf("detected") != -1){
          continue
        }
        else if(plugin.indexOf("failed") != -1 || plugin.indexOf
("fausse") != -1){
          continue
        }
        else{
          plugstr = plugin
        }

        // Add the fields to a the final array.
        finalArray.push([date, site, plugstr, updatestr])
      }
      else{
        continue
      }
    }
  }
  return finalArray
}
```

Watchful List Data Separator (Singular Site)

This script separates the data provided by watchful report on a singular site. This will segment each field that is needed while, on the Sheets side of things, a query handles the order and listing.

```
/*
 * Created by Joshua Greenert on 9/16/2020
 *
 * This script will separate the site, date, and plugin update

                              PSEUDOCODE
 Find indexOf Activity Logs
        grab website and end at forward slash(/)
Start where forward slash ends
        grab date and end at indexOf m
Start where m ends
        grab update data where month begins
                month variable will need to be specified in 01, 10, 12
format
```

```
 */

function SiteChangeList(index1, index2) {
  var dataSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName
("Multiple_Input")

  // Add the ranges to pull data from.
  var dataArrayRange = dataSheet.getRange("A1:A").getValues()
  var dataRange = []

  // Variables to hold the data.
  var siteRange = []
  var dateRange = []
  var pluginRange = []
  var finalArray = []

  var site = ""
  var date = ""
  var plugin = ""
  var updatestr = ""

  var plugstr = 0
  var datelen = 0
  var pluginlen = 0

  var month


  // Set all pieces within the given array to strings.
  for(var i = 0; i < dataArrayRange.length; i++){
    dataRange.push(dataArrayRange[i][0].toString())
  }

  // Set each site, date, and plugin through a for loop.
  for(var i = 0; i < dataRange.length; i++){

    // Set the site for each record, the date, and the plugin data.
    site = dataRange[i].substring(13, dataRange[i].indexOf("/"))

    /* This is to catch the 360twin issue and will need to be updated
every month that the report is ran.*/
    if(dataRange[i].indexOf("360twin") == -1){
      dataRange[i] = dataRange[i].substring(dataRange[i].indexOf
(index1), dataRange[i].length)
    }
    else{
      dataRange[i] = dataRange[i].substring(dataRange[i].indexOf
(index2), dataRange[i].length)
    }
```

```
    // For loop through each record after the first to ensure that the
data is grabbed.
    for(var j = 0; j < dataRange[i].length; j++){

      if((dataRange[i].indexOf("am") != -1) || (dataRange[i].indexOf
("pm") != -1)){
        date = dataRange[i].substring(0, dataRange[i].indexOf("m") +1)
        Logger.log(date)
        month = date.substring(0, 2)
        datelen = date.length
        date = new Date(Date.parse(date.substring(0, 10)))

        // Update the dataRange again to strip the data based on the
end of the date.
        dataRange[i] = dataRange[i].substring(datelen, dataRange[i].
length)

        if(dataRange[i].indexOf(month) == -1){
          plugin = (dataRange[i].indexOf("Page") != -1) ? dataRange[i].
substring(0, dataRange[i].indexOf("Page")) : dataRange[i].substring(0,
dataRange[i].length)
          pluginlen = plugin.length
        }
        else{
          plugin = dataRange[i].substring(0, dataRange[i].indexOf(month
+ "/"))
          pluginlen = plugin.length
        }
        dataRange[i] = dataRange[i].substring(pluginlen, dataRange[i].
length)

        // Separate the plugin from the modification.
        if(plugin.indexOf("updated") != -1){
          plugstr = plugin.substring(0, plugin.indexOf("has been
updated")-1)
          updatestr = plugin.substring(plugin.indexOf("updated"),
plugin.length)
          updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
        }
        else if(plugin.indexOf("installed") != -1){
          plugstr = plugin.substring(0, plugin.indexOf("has been
installed")-1)
          updatestr = plugin.substring(plugin.indexOf("installed"),
plugin.length)
          updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
        }
        else if(plugin.indexOf("modified") != -1){
```

```
            plugstr = plugin.substring(0, plugin.indexOf("has been
modified")-1)
            updatestr = plugin.substring(plugin.indexOf("modified"),
plugin.length)
            updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
          }
          else if(plugin.indexOf("deleted") != -1){
            plugstr = plugin.substring(0, plugin.indexOf("has been
deleted")-1)
            updatestr = plugin.substring(plugin.indexOf("deleted"),
plugin.length)
            updatestr = updatestr.charAt(0).toUpperCase() + updatestr.
substring(1,updatestr.length)
          }
          else if(plugin.indexOf("detected") != -1){
            continue
          }
          else if(plugin.indexOf("failed") != -1 || plugin.indexOf
("fausse") != -1){
            continue
          }
          else{
            plugstr = plugin
          }

          // Add the fields to a the final array.
          finalArray.push([date, site, plugstr, updatestr])
        }
        else{
          continue
        }
      }
    }
    return finalArray
  }
```

Java Selenium Automation Projects

This page is being created to act as a folder for the Selenium automated projects are slowing growing too big to jump in and out of. I say that now but I know that it will grow much larger than expected.

Create A List of Write-A-Review Links For Stores

This page will host the code used to create a list of links where a user can write a review for the business (CBD specifically). After completing this process, there may be duplicates. Use Google Sheets, or alternative, to set a countif function to determine how many are in the list. If the default is presented when the code executes, then the duplicate will be the result of the item no longer being suggested on Google. Sort the entire results by the largest number to isolate the ones that didn't associate to verify.

The code

```
package challenge;

import java.awt.AWTException;
import java.awt.Robot;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import javax.swing.JFileChooser;
import javax.swing.filechooser.FileSystemView;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

/*
 * Created by Joshua Greenert on 12/08/2020
 *
 * This program will take all of the store locations for CBD to provide
to a
 * website that generates the review submission link.  These links will
be stored
 * on a file that can then be copied to the Google sheet.
 */
public class CreateBusinessReviewList {

        static WebDriver driver = null;
        static Robot r;

        public static void main(String[] args) throws IOException,
InterruptedException, AWTException {

                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> businesses = new ArrayList<String>();

                File businessListFile;
```

```
                // Go through the text file and store all of the
addresses in the list.
                try {

                        // Create a file chooser
                        final JFileChooser jc = new JFileChooser
(FileSystemView.getFileSystemView().getHomeDirectory());
                        jc.setDialogTitle("Select the file with the
websites");
                        jc.setFileSelectionMode(JFileChooser.
FILES_AND_DIRECTORIES);

                        // In response to a button click:
                        int returnVal = jc.showOpenDialog(null);

                        if (returnVal == JFileChooser.APPROVE_OPTION) {

                                // Set the file from the users
selection and update the path variable.
                                businessListFile = jc.getSelectedFile();

                                Scanner myReader = new Scanner
(businessListFile);

                                // While there is another url, take the
required pieces and add it to the lists.
                                while (myReader.hasNextLine()) {
                                        String data = myReader.
nextLine();

                                        businesses.add(data);
                                }

                                // For each loop to pass name, site,
and date to capture object.
                                for (int i = 0; i < businesses.size();
i++) {
                                        testWebsite(businesses.get(i));
                                }

                                myReader.close();
                        }
                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                }

        }

        // Initializes the driver.
```

```
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite(String business) throws
IOException, InterruptedException, AWTException {
                initDriver();

                // Set the site for the driver.
                driver.get("https://whitespark.ca/google-review-link-
generator/");

                // Find the textbox for the input and enter the input.
                WebElement businessInput = driver.findElement(By.id
("pac-input"));
                businessInput.sendKeys(business);
                // Wait for 2 seconds to recognize options.
                TimeUnit.SECONDS.sleep(2);
                businessInput.sendKeys(Keys.ARROW_DOWN);
                businessInput.sendKeys(Keys.ENTER);

                // Wait for 2 seconds to recognize options.
                TimeUnit.SECONDS.sleep(2);

                // Get the review link and save it as a variable.
                WebElement reviewInput = driver.findElement(By.xpath("
//input[@id='gp_link']"));
                String reviewText = reviewInput.getAttribute("value");

                // Set each item within a string for a new row in a
text file.
                try(FileWriter fw = new FileWriter("../../Documents
/Github/webpage_screenshots/reviewListFinal.txt", true);
                    BufferedWriter bw = new BufferedWriter(fw);
                    PrintWriter out = new PrintWriter(bw))
                {

                    out.println(reviewText);

                } catch (IOException e) {
                        e.printStackTrace();
```

```
                }


                // Check the title of the page
                System.out.println("The link for " +reviewText+" has
been added!");


                destroy();
        }

        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }
}
```

List All Website Expiration Dates

This page will host the code that is set to automate the process of collecting the expiration dates of websites and pushing the list to GitHub through JGit.

The Code

```
package selenium;


import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URISyntaxException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;

import org.eclipse.jgit.api.Git;
import org.eclipse.jgit.api.PushCommand;
import org.eclipse.jgit.api.errors.GitAPIException;
import org.eclipse.jgit.transport.UsernamePasswordCredentialsProvider;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;


/*
```

```
 * Created by Joshua Greenert on 12/17/2020
 *
 * This script will capture the information about each website to store
a list of expiration dates.
 */

public class WhoIsExpirationDate {

        static WebDriver driver = null;

        public static void main(String[] args) throws IOException,
InterruptedException, URISyntaxException, GitAPIException {


                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> sites = new ArrayList<String>();

                File file;
                File deleteFile = new File("..//..//Documents//Github
//webpage_screenshots//siteExpiryDate.txt");

                // Delete the file that is used to hold the gtmetrix
scores
        if(deleteFile.delete())
        {
            System.out.println("File deleted successfully");
        }
        else
        {
            System.out.println("Failed to delete the file");
        }

                // Go through the text file and store all of the
addresses in the list.
                try {

                        // Set the file from the users selection and
update the path variable.
                        file = new File("..//..//Documents//GitHub
//webpage_screenshots/fullwebsitelist.txt");

                        Scanner myReader = new Scanner(file);

                        // While there is another url, take the
required pieces and add it to the lists.
                        while (myReader.hasNextLine()) {
                                String data = myReader.nextLine();

                                sites.add(data);
```

```
                        }


                        // For each loop to pass name, site, and date
to capture object.
                        for (int i = 0; i < sites.size(); i++) {
                                testWebsite(sites.get(i));
                        }

                        myReader.close();

                        // Push commit
                        sendToGit();

                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                }

        }

        // Initializes the driver.
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite(String site) throws
InterruptedException, IOException {
                initDriver();

                String resultText = "";

                // Set the site for the driver.
                driver.get("https://godaddy.com/whois");

                // Look up the input field and enter the request.
                WebElement searchField = driver.findElement(By.name
("domain"));
                searchField.sendKeys(site);
                searchField.sendKeys(Keys.RETURN);
```

```
                // Wait for 5 seconds to continue.
                TimeUnit.SECONDS.sleep(5);

                // Find the text from the results.
                WebElement results = driver.findElement(By.id("whois-
data"));
                resultText = results.getText();

                // Separate the text from the rest.
                if(resultText.indexOf("Registrar Registration
Expiration Date") != -1 &&
                                resultText.indexOf("Registry Expiry
Date") != -1) {
                        resultText = resultText.substring(resultText.
indexOf("Registry Expiry Date") + 22,
                                        resultText.indexOf("Registry
Expiry Date") + 32);
                }
                else if (resultText.indexOf("Registrar Registration
Expiration Date") != -1){
                        resultText = resultText.substring(resultText.
indexOf("Registrar Registration Expiration Date") + 40,
                                        resultText.indexOf("Registrar
Registration Expiration Date") + 50);
                }
                else {
                        resultText = resultText.substring(resultText.
indexOf("Registry Expiry Date") + 22,
                                        resultText.indexOf("Registry
Expiry Date") + 32);
                }

                // Set each item within a string for a new row in a
text file.
                try(FileWriter fw = new FileWriter("..//..//Documents
//Github//webpage_screenshots//siteExpiryDate.txt", true);
                    BufferedWriter bw = new BufferedWriter(fw);
                    PrintWriter out = new PrintWriter(bw))
                {

                    out.println(resultText);

                } catch (IOException e) {
                        e.printStackTrace();
                }


                destroy();
        }
```

```
        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }

        // Create a method to call that will send files to GitHub.
        public static void sendToGit() throws IOException,
URISyntaxException, GitAPIException {

                // Open the path selected by user.
                File repo = new File("..//..//Documents//Github
//webpage_screenshots");
                Git git = Git.open(repo);

                // Commit all items in the path.
                git.commit().setAll(true).setMessage("Automated push
from Java Program").call();

                // push to remote:
            PushCommand pushCommand = git.push();
            pushCommand.setCredentialsProvider(new
UsernamePasswordCredentialsProvider
("d8d10b373a560141099d6fc893755fa246e77a5c", ""));
            pushCommand.call();

            System.out.println("Pushed to GitHub!");
        }
}
```

Setting Up GTMetrix Website Testing Automation

This page will serve as a host for the steps and code required to test each website on GTMetrix through use of Java and the Selenium driver. Once the website test is completed on GTMetrix, the application will capture the PageSpeed, YSlow, Fully Loaded Time, Total Page Size, and requests. Each item will be written to a text file in a similar pattern shown below:

[website_name],PageSpeed,YSlow,Fully_Loaded_Time,Total_Page Size,requests

These values allow the user to see a full picture of what data they are getting from the GTMetrix site and use a simple/standard delimiter (comma). Therefore, when pasting to Google Sheets, the values can be split with ease.

NOTE: This is not the final step to setting up this script for full automation. The next step involves setting up the Sheets API to allow the data to be posted to the spreadsheet directly, rather than requiring the manual copy/pasting from the user.

Here is the code to process this for Mac

```
package challenge;

import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
```

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import javax.swing.JFileChooser;
import javax.swing.filechooser.FileSystemView;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

/*
 * Created by Joshua Greenert on 10/13/2020
 *
 * This program will pull up the gtmetrix site and submit a web page
hyperlink.
 * After waiting until the page has loaded (unsure about time frame)
the elements will be captured
 * from the pagespeed, yslow, fully loaded, size request, suggestion 1,
suggestion 2, and suggestion 3
 * fields.
 */
public class GTMetrixTesting {

        static WebDriver driver = null;
        static Robot r;

        static String pagespeed = "";
        static String yslow = "";
        static String loadtime = "";
        static String pagesize = "";
        static String requests = "";
        static String performance = "";
        static String details = "";

        public static void main(String[] args) throws IOException,
InterruptedException, AWTException {

                // Create List object to store URLS and file object to
store user-file.
                        ArrayList<String> sites = new
ArrayList<String>();
                        ArrayList<String> names = new
ArrayList<String>();

                        File deleteFile = new File("../../Documents
/Github/webpage_screenshots/gtmetrixdata.txt");
                        File siteFile;
```

```
                            String webname = "";
                            String site = "";
                            String name = "url";
                            String username = "[USERNAME]";
                            String password = "[PASSWORD]";

                            // Delete the file that is used to hold the 
gtmetrix scores
                if(deleteFile.delete())
                {
                    System.out.println("File deleted successfully");
                }
                else
                {
                    System.out.println("Failed to delete the file");
                }

                            // Go through the text file and store all of 
the addresses in the list.
                            try {

                                    // Create a file chooser
                                    final JFileChooser jc = new JFileChooser
(FileSystemView.getFileSystemView().getHomeDirectory());
                                    jc.setDialogTitle("Select the file with 
the websites");
                                    jc.setFileSelectionMode(JFileChooser.
FILES_AND_DIRECTORIES);

                                    // In response to a button click:
                                    int returnVal = jc.showOpenDialog(null);

                                    if (returnVal == JFileChooser.
APPROVE_OPTION) {

                                            // Set the file from the users 
selection and update the path variable.
                                            siteFile = jc.getSelectedFile();

                                            Scanner myReader = new Scanner
(siteFile);

                                            // While there is another url, 
take the required pieces and add it to the lists.
                                            while (myReader.hasNextLine()) {
                                                    String data = myReader.
nextLine();

                                                    // Check for the https
```

```
portion of the string and print that line
                                                    if (data.indexOf
("https") != -1) {
                                                            if(data.indexOf
(".com") != -1) {
                                                                    site =
data.substring(data.indexOf("//") + 2, data.indexOf(".com")+4);
                                                                    site =
"https://" + site;
                                                                    sites.
add(site);
                                                            }
                                                            else if(data.
indexOf(".net") != -1) {
                                                                    site =
data.substring(data.indexOf("//") + 2, data.indexOf(".net")+4);
                                                                    site =
"https://" + site;
                                                                    sites.
add(site);
                                                            }
                                                            else if(data.
indexOf(".org") != -1) {
                                                                    site =
data.substring(data.indexOf("//") + 2, data.indexOf(".org")+4);
                                                                    site =
"https://" + site;
                                                                    sites.
add(site);
                                                            }
                                                            else if(data.
indexOf(".us") !=  -1) {
                                                                    site =
data.substring(data.indexOf("//") + 2, data.indexOf(".us")+3);
                                                                    site =
"https://" + site;
                                                                    sites.
add(site);
                                                            }
                                                            else {

continue;
                                                            }

                                                            // If the
string has a www, then get the length up to last period; remove the
                                                            // 'www.'
                                                            if (data.indexOf
("www") == -1) {
                                                                    webname
```

```
= data.substring(data.indexOf("//") + 2, data.indexOf(".", data.indexOf
(".")));
                                                                names.
add(webname);
                                                        } else {
                                                                webname
= data.substring(data.indexOf("//") + 2, data.lastIndexOf("."));
                                                                webname
= webname.substring(4, webname.length());
                                                                names.
add(webname);
                                                        }
                                                }
                                        }

                                        // For each loop to pass name,
site, and date to capture object.
                                        for (int i = 0; i < sites.
size(); i++) {
                                                testWebsite(sites.get
(i), name, username, password, names.get(i));
                                        }

                                        myReader.close();
                                }
                        } catch (FileNotFoundException e) {
                                e.printStackTrace();
                }

        }

        // Initializes the driver.
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite(String site, String name, String
user, String pass, String webname) throws IOException,
InterruptedException, AWTException {
                initDriver();
```

```
                // Reset the variables for each value.
                pagespeed = "";
                yslow = "";

                // Set the site for the driver.
                driver.get("https://gtmetrix.com/");

                // Find the login button and authenticate.
                WebElement loginButton = driver.findElement(By.className
("js-auth-widget-link"));
                loginButton.click();
                WebElement username = driver.findElement(By.id("li-
email"));
                username.sendKeys(user);
                WebElement password = driver.findElement(By.id("li-
password"));
                password.sendKeys(pass);

                // Submit the credentials through robot and wait for
page to load up.
                r = new Robot();
                r.keyPress(KeyEvent.VK_ENTER);
                r.keyRelease(KeyEvent.VK_ENTER);
                r.keyPress(KeyEvent.VK_ENTER);
                r.keyRelease(KeyEvent.VK_ENTER);
                TimeUnit.SECONDS.sleep(5);

                // Find the URL field and enter the name of the site.
                WebElement element = driver.findElement(By.name(name));
                element.sendKeys(site);

                // Create a new web element for the button and press
submit.
                WebElement submitButton = driver.findElement(By.
className("analyze-form-button"));
                submitButton.click();

                // Wait for 25 seconds to continue.
                TimeUnit.SECONDS.sleep(50);

                // After the testing is complete, grab the fields
needed.
                WebElement performscores = driver.findElement(By.
className("report-scores"));
                WebElement pagedetails = driver.findElement(By.className
("report-page-details"));

                performance = performscores.getText();
                details = (String)pagedetails.getText();
```

```
                int count = 0;
                // Use string operations to separate the scores out.
                for(int i = 0; i < performance.length(); i++) {
                        if(Character.isDigit(performance.charAt(i))) {
                                if(count < 2) {
                                        pagespeed += performance.charAt
(i);
                                        count++;
                                }
                                else if(count >= 2) {
                                        yslow += performance.charAt(i);
                                        count++;
                                }
                        }
                }

                count = 0;
                for(int i = 0; i < details.length(); i++) {

                        if(count ==3) {
                                break;
                        }

                        if(Character.isDigit(details.charAt(i))) {
                                if(count < 1) {
                                        details = details.substring(i,
details.length());
                                        loadtime = details.substring(0,
details.indexOf("s"));
                                        count++;
                                        i = details.indexOf("s")+1;
                                }
                                else if(count == 1) {
                                        details = details.substring(i,
details.length());

                                        if(details.indexOf("MB") != -1)
{
                                                pagesize = details.
substring(0, details.indexOf("MB")+2);
                                                count++;
                                                i = details.indexOf
("MB")+2;
                                        }
                                        else {
                                                pagesize = details.
substring(0, details.indexOf("KB")+2);
                                                count++;
                                                i = details.indexOf
("KB")+2;
```

```
                                        }
                                }
                                else if(count == 2) {
                                        details = details.substring(i,
details.length());
                                        requests = details.substring(0,
details.length());
                                        count++;
                                        break;
                                }
                        }
                }

                details = webname + "," + pagespeed + "," + yslow + ","
+ loadtime + "," + pagesize + "," + requests + "\n";

                // Set each item within a string for a new row in a
text file.
                try(FileWriter fw = new FileWriter("../../Documents
/Github/webpage_screenshots/gtmetrixdata.txt", true);
                    BufferedWriter bw = new BufferedWriter(fw);
                    PrintWriter out = new PrintWriter(bw))
                {

                    out.print(details);

                } catch (IOException e) {
                        e.printStackTrace();
                }


                // Check the title of the page
                System.out.println("The site " +site+" has been
tested!");


                destroy();
        }

        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }
}
```

References	
URI	Comments
https://stackoverflow.com/questions/23804123/selenium-gettext	Possible solution for getting the text from the class/id on a website.

Setting Up Pingdom Website Testing Automation

This page will show the code needed to automate testing a list of websites for tools.pingdom.com. Unfortunately, there are pieces that the selenium driver cannot find, regardless of time/effort put forward such as the toggle selection for testing region. This tends to add more time to the testing tool, and pushes the scope of the expectation further than anticipated.

Code for Mac

```
package challenge;

import java.awt.AWTException;
import java.awt.Robot;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class PingdomTesting {

        static WebDriver driver = null;
        static Robot r;

        static String details;
        static String performance;
        static String loadtime;
        static String requests;
        static String pagesize;

        public static void main(String[] args) throws IOException,
InterruptedException, AWTException {

                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> sites = new ArrayList<String>();
                ArrayList<String> names = new ArrayList<String>();

                File deleteFile = new File("../../Documents/Github
/webpage_screenshots/pingdomdata.txt");
                File siteFile;

                String webname = "";
                String site = "";
                String name = "urlInput";
```

```
                // Delete the file that is used to hold the gtmetrix
scores
        if(deleteFile.delete()) {
            System.out.println("File deleted successfully");
        }
        else{
            System.out.println("Failed to delete the file");
        }

        siteFile = new File("../../Documents/Github/webpage_screenshots
/webtesting.rtf");

        Scanner myReader = new Scanner(siteFile);

                // While there is another url, take the required pieces
and add it to the lists.
                while (myReader.hasNextLine()) {
                        String data = myReader.nextLine();

                        // Check for the https portion of the string
and print that line
                        if (data.indexOf("https") != -1) {
                                if(data.indexOf(".com") != -1) {
                                        site = data.substring(data.
indexOf("//") + 2, data.indexOf(".com")+4);
                                        site = "https://" + site;
                                        sites.add(site);
                                }
                                else if(data.indexOf(".net") != -1) {
                                        site = data.substring(data.
indexOf("//") + 2, data.indexOf(".net")+4);
                                        site = "https://" + site;
                                        sites.add(site);
                                }
                                else if(data.indexOf(".org") != -1) {
                                        site = data.substring(data.
indexOf("//") + 2, data.indexOf(".org")+4);
                                        site = "https://" + site;
                                        sites.add(site);
                                }
                                else if(data.indexOf(".us") !=  -1) {
                                        site = data.substring(data.
indexOf("//") + 2, data.indexOf(".us")+3);
                                        site = "https://" + site;
                                        sites.add(site);
                                }
                                else {
                                        continue;
                                }
```

```
                                // If the string has a www, then get
the length up to last period; remove the
                                // 'www.'
                                if (data.indexOf("www") == -1) {
                                        webname = data.substring(data.
indexOf("//") + 2, data.indexOf(".", data.indexOf(".")));
                                        names.add(webname);
                                } else {
                                        webname = data.substring(data.
indexOf("//") + 2, data.lastIndexOf("."));
                                        webname = webname.substring(4,
webname.length());
                                        names.add(webname);
                                }
                        }
                }

        // For each loop to pass name, site, and date to capture object.
                for (int i = 0; i < sites.size(); i++) {
                        testWebsite(sites.get(i), name, names.get(i));
                }

                myReader.close();
        }

        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }

        // Initializes the driver.
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite(String site, String name, String
webname) throws IOException, InterruptedException, AWTException {
                initDriver();

                // Set the site for the driver.
```

```
                driver.get("https://tools.pingdom.com/");

                // Find the URL field and enter the name of the site.
                WebElement element = driver.findElement(By.id(name));
                element.sendKeys(site);

                // Create a new web element for the button and press
submit.
                WebElement submitButton = driver.findElement(By.xpath("
//input[@value='START TEST']"));
                submitButton.click();

                // Wait for seconds to continue.
                TimeUnit.SECONDS.sleep(50);

                // After the testing is complete, grab the fields
needed.
                WebElement performscore = driver.findElement(By.name
("Performance grade"));
                WebElement pagesizescore = driver.findElement(By.name
("Page size"));
                WebElement loadtimescore = driver.findElement(By.name
("Load time"));
                WebElement requestscore = driver.findElement(By.name
("Requests"));

                // Set items as text objects.  NOTE: any 0s are the
result of the page not loading before the element was captured.
                performance = performscore.getText();
                pagesize = pagesizescore.getText();
                loadtime = loadtimescore.getText();
                requests = requestscore.getText();

                // Cut the pieces down to numbers.
                performance = performance.replaceAll("[^\\d.]", "");
                loadtime = loadtime.replaceAll("[^\\d.]", "");
                requests = requests.replaceAll("[^\\d.]", "");

                // Pagesize is of course a special lady.
                pagesize = pagesize.indexOf("MB") == -1 ? pagesize.
substring(pagesize.indexOf("ze")+2, pagesize.indexOf("KB")+2) :
pagesize.substring(pagesize.indexOf("ze")+2, pagesize.indexOf("MB")+2);
                pagesize = pagesize.replaceAll("\n", "");

                details = webname + "," + performance + "," + loadtime
+ "," + pagesize + "," + requests +"\n";

                // Set each item within a string for a new row in a
text file.
                try(FileWriter fw = new FileWriter("../../Documents
```

```
/Github/webpage_screenshots/pingdomdata.txt", true);
                    BufferedWriter bw = new BufferedWriter(fw);
                    PrintWriter out = new PrintWriter(bw))
                {

                    out.print(details);

                } catch (IOException e) {
                        e.printStackTrace();
                }

                // Check the title of the page
                System.out.println("The site " +site+" has been
tested!");

                destroy();
        }

}
```

Setting Up Screenshots of All Web Pages

This page will be the repository for the notes and successes throughout the process of setting up a tool manually that goes for $100+ online. The overall goal is to have the script run through a text file which holds all web addresses to capture, capture a screenshot of each instance, save the screenshot to a folder, upload the entire folder to a GitHub Repository, then clean up the folder by deleting the screenshots just taken.

Pseudocode

https://docs.google.com/document/d/1uNrBo6Fv3mMcNtj6cRWhQqytX-XehEbFgnAEZij1e64/edit

Requirements

- **Eclipse**
- **Dependency Jar files**
  - **client-combined jar**
  - **commons.io.jar**
  - **byte-buddy jar**
  - **commons-exec jar**
  - **guava jar**
  - **okhttp jar**
  - **okio jar**
  - **org.eclipse.jgit**

Beginning Piece That Confirms Plausibility

```
package challenge;

import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.Point;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class WebPageScreenShotTaker {

	static int screenshotNum=0;
	static WebDriver driver=null;
	public static void main(String[] args) throws IOException {

		initDriver();
		WebPageScreenShotTaker ws=new WebPageScreenShotTaker();
		ws.capture("https://www.google.com");
		ws.capture("https://facebook.com");
		destroy();

	}
	public static void initDriver()
	{
		System.setProperty("webdriver.chrome.driver",
				"driver/chromedriver.exe");
		driver=new ChromeDriver();
		driver.manage().window().setPosition(new Point(-2000, 0));
	}
	public void capture(String site) throws IOException
	{
		screenshotNum++;
		driver.get(site);
		File scrFile = ((TakesScreenshot)driver).getScreenshotAs
(OutputType.FILE);
		FileUtils.copyFile(scrFile, new File("site"+screenshotNum+".
png"));
		System.out.println("Took Screenshot for "+site+" and saved as
"+"site"+screenshotNum+".png");
	}
	public static void destroy()
	{
		driver.quit();
	}
}
```

The initial goal was to have the files automatically send to GitHub as the last step. However, due to issues in fighting with the library (or Eclipse-specific issues) the code listed below puts everything into the GitHub folder selected and then commits and pushes the folder to the GitHub folder selected.

For Mac

```
package challenge;
/*
 *  Created by Joshua Greenert on 10/5/2020
 *
 *  This script will read from a text file to create a list of links to
 *  take screenshots of.  The screenshots will be saved in a folder to
```

```
then
 *  be pushed to GitHub.
 */

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.net.URISyntaxException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import javax.imageio.ImageIO;
import javax.swing.JFileChooser;
import javax.swing.filechooser.FileSystemView;
import org.apache.commons.io.FileUtils;
import org.eclipse.jgit.api.Git;
import org.eclipse.jgit.api.PushCommand;
import org.eclipse.jgit.api.errors.GitAPIException;
import org.eclipse.jgit.transport.UsernamePasswordCredentialsProvider;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import ru.yandex.qatools.ashot.AShot;
import ru.yandex.qatools.ashot.Screenshot;
import ru.yandex.qatools.ashot.shooting.ShootingStrategies;

public class WebPageScreenShotTaker {

        static WebDriver driver = null;

        public static void main(String[] args) throws IOException {

                // Start up the browser window and create screenshot
object.
                WebPageScreenShotTaker ws = new
WebPageScreenShotTaker();

                // Create List object to store URLS
                ArrayList<String> sites = new ArrayList<String>();
                ArrayList<String> names = new ArrayList<String>();

                String name = "";
                String site = "";
                String path = "";

                File file;
```

```
                SimpleDateFormat format = new SimpleDateFormat("MM-dd-
yyyy");
                String dateString = format.format(new Date());

                // Go through the text file and store all of the
addresses in the list.
                try {

                        // Create a file chooser
                        final JFileChooser jc = new JFileChooser
(FileSystemView.getFileSystemView().getHomeDirectory());
                        jc.setDialogTitle("Select the website_map file
from the Github Repo");
                        jc.setFileSelectionMode(JFileChooser.
FILES_AND_DIRECTORIES);

                        // In response to a button click:
                        int returnVal = jc.showOpenDialog(null);

                        if(returnVal == JFileChooser.APPROVE_OPTION) {

                                // Set the file from the users
selection and update the path variable.
                                file = jc.getSelectedFile();
                                path = file.getAbsolutePath();
                                path = path.substring(0, path.length()
- 15);

                                Scanner myReader = new Scanner(file);

                                // While there is another url, take the
required pieces and add it to the lists.
                                while (myReader.hasNextLine()) {
                                        String data = myReader.
nextLine();

                                        // Check for the https portion
of the string and print that line
                                        if (data.indexOf("https") !=
-1) {
                                                site = data.substring
(data.indexOf("//") + 2, data.length());
                                                site = "https://" +
site;
                                                sites.add(site);

                                                // If the string has a
www, then get the length up to last period; remove the 'www.'
                                                if(data.indexOf("www")
== -1){
```

```
                                                        name = data.
substring(data.indexOf("//") + 2, data.indexOf(".", data.indexOf(".")));
                                                        names.add(name);
                                                }
                                                else {
                                                        name = data.
substring(data.indexOf("//") + 2, data.lastIndexOf("."));
                                                        name = name.
substring(4, name.length());

                                                        names.add(name);
                                                }
                                        }
                                }

                                // For each loop to pass name, site,
and date to capture object.
                                for (int i = 0; i < sites.size(); i++) {
                                        ws.capture(sites.get(i), names.
get(i), dateString, path);
                                }

                                myReader.close();
                        }
                }
                catch (FileNotFoundException e) {
                        e.printStackTrace();
                }

                try {
                        sendToGit(path);
                } catch (IOException | URISyntaxException |
GitAPIException e) {
                        e.printStackTrace();
                }

        }

        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();

        }

        public void capture(String site, String name, String date,
String path) throws IOException {
                initDriver();
```

```
                // Set up path and increase screenshot number.
                path = path + name + "/" + date.substring(date.length()
- 4, date.length()) + "/" + date.substring(0, 2) + "/";

                driver.get(site);

                // Remove all slashes from the site name
                site = site.replaceAll("/", "|");

                File scrFile = ((TakesScreenshot) driver).
getScreenshotAs(OutputType.FILE);
                FileUtils.copyFile(scrFile, new File(path + site + ".
png"));

                Screenshot screenshot = new AShot().shootingStrategy
(ShootingStrategies.viewportPasting(1000))
                                .takeScreenshot(driver);
                ImageIO.write(screenshot.getImage(), "PNG", new File
(path + site + ".png"));

                System.out.println("Took Screenshot for " + site + "
and saved as " + site + ".png");

                destroy();
        }

        public static void destroy() {
                driver.quit();
        }

        // Create a method to call that will send files to GitHub.
        public static void sendToGit(String path) throws IOException,
URISyntaxException, GitAPIException {

                // Open the path selected by user.
                File repo = new File(path);
                Git git = Git.open(repo);

                // Commit all items in the path.
                git.commit().setAll(true).setMessage("Automated push
from Java Program").call();

                // push to remote:
            PushCommand pushCommand = git.push();
            pushCommand.setCredentialsProvider(new
UsernamePasswordCredentialsProvider([USERNAME], [PASSWORD]));
            pushCommand.call();

        }
```

```
}
```

For Windows

```
package selenium;

/*
 *  Created by Joshua Greenert on 10/5/2020
 *
 *  This script will read from a text file to create a list of links to
 *  take screenshots of.  The screenshots will be saved in a folder to
then
 *  be pushed to GitHub.
 */

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.net.URISyntaxException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import javax.imageio.ImageIO;
import org.apache.commons.io.FileUtils;
import org.eclipse.jgit.api.Git;
import org.eclipse.jgit.api.PushCommand;
import org.eclipse.jgit.api.errors.GitAPIException;
import org.eclipse.jgit.transport.UsernamePasswordCredentialsProvider;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import ru.yandex.qatools.ashot.AShot;
import ru.yandex.qatools.ashot.Screenshot;
import ru.yandex.qatools.ashot.shooting.ShootingStrategies;

public class WebPageScreenShotTaker {

        static WebDriver driver = null;
        static int num = 1;

        public static void main(String[] args) throws IOException {

                // Start up the browser window and create screenshot
object.
                WebPageScreenShotTaker ws = new
WebPageScreenShotTaker();
```

```
				// Create List object to store URLS
				ArrayList<String> sites = new ArrayList<String>();
				ArrayList<String> names = new ArrayList<String>();

				String name = "";
				String site = "";
				String path = "";

				File file;

				SimpleDateFormat format = new SimpleDateFormat("MM-dd-
yyyy");
				String dateString = format.format(new Date());

				// Go through the text file and store all of the
addresses in the list.
				try {
					// Set the file from the users selection and
update the path variable.
					file = new File("..//..//Documents//GitHub
//webpage_screenshots//website_map.txt");
					path = file.getAbsolutePath();
					path = path.substring(0, path.length() - 15);

					Scanner myReader = new Scanner(file);

					// While there is another url, take the
required pieces and add it to the lists.
					while (myReader.hasNextLine()) {
						String data = myReader.nextLine();

						// Check for the https portion of the
string and print that line
						if (data.indexOf("https") != -1) {
							site = data.substring(data.
indexOf("//") + 2, data.length());
							site = "https://" + site;
							sites.add(site);

							// If the string has a www,
then get the length up to last period; remove the 'www.'
							if(data.indexOf("www") == -1){
								name = data.substring
(data.indexOf("//") + 2, data.indexOf(".", data.indexOf(".")));
								names.add(name);
							}
							else {
								name = data.substring
(data.indexOf("//") + 2, data.lastIndexOf("."));
```

```
                                                name = name.substring
(4, name.length());
                                                names.add(name);
                                        }
                                }
                        }

                        // For each loop to pass name, site, and date
to capture object.
                        for (int i = 0; i < sites.size(); i++) {
                                ws.capture(sites.get(i), names.get(i),
dateString, path);
                        }

                        myReader.close();
                }
                catch (FileNotFoundException e) {
                        e.printStackTrace();
                }

                try {
                        sendToGit(path);
                } catch (IOException | URISyntaxException |
GitAPIException e) {
                        e.printStackTrace();
                }

        }

        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();

        }

        public void capture(String site, String name, String date,
String path) throws IOException {
                initDriver();

                // Set up path and increase screenshot number.
                path = path + name + "/" + date.substring(date.length()
- 4, date.length()) + "/" + date.substring(0, 2) + "/";

                driver.get(site);

                // Remove all slashes from the site name
```

```
                site = site.replaceAll("/", "-");
                site = site.replaceAll(":", "_");

                // check if the file exists and delete it.
                File deleteFile = new File(path + site + num +".png");
                if(deleteFile.delete()) {
                        System.out.println("File deleted");
                }

                File scrFile = ((TakesScreenshot) driver).
getScreenshotAs(OutputType.FILE);
                FileUtils.copyFile(scrFile, new File(path + site + num
+".png"));

                Screenshot screenshot = new AShot().shootingStrategy
(ShootingStrategies.viewportPasting(1000))
                                .takeScreenshot(driver);
                ImageIO.write(screenshot.getImage(), "PNG", new File
(path + site + num + ".png"));

                System.out.println("Took Screenshot for " + site + "
and saved as " + site + ".png");
                num++;

                destroy();
        }

        public static void destroy() {
                driver.quit();
        }

        // Create a method to call that will send files to GitHub.
        public static void sendToGit(String path) throws IOException,
URISyntaxException, GitAPIException {

                // Open the path selected by user.
                File repo = new File(path);
                Git git = Git.open(repo);

                // Commit all items in the path.
                git.commit().setAll(true).setMessage("Automated push
from Java Program").call();

                // push to remote:
            PushCommand pushCommand = git.push();
            pushCommand.setCredentialsProvider(new
UsernamePasswordCredentialsProvider
("d8d10b373a560141099d6fc893755fa246e77a5c", ""));
            pushCommand.call();
```

```
            // State that the process was completed.
            System.out.println("Sent to GitHub!");

        }
}
```

The next step is to create an executable that can be ran anywhere by simply having the singular file (no extra jars, eclipse, etc.). Since we are using Mac computers, the 'Wine' application is potentially needed to run a standard executable file. Further, JDK for Java 11 is essential as well.

In order to create the jar file, that will later be converted to an executable, use the following steps listed below:

1. Create a folder to hold the contents that are going to be in the jar file.
2. Add all applicable jar files that were needed for execution.
3. Add driver folder needed for execution of screenshot jar
4. Add manifest file either through command prompt or the use of standard texteditor
    a. manifest.mf should include the following:
        i. Manifest-Version: 1.0
        ii. Main-Class: folder.classname (challenge.WebPageScreenShotTaker)
5. Create the jar using the code listed below

    a.
    ```
    jar cvf TicTacToe.jar *
    ```

References	
URI	**Comments**
https://commons.apache.org/proper/commons-io/download_io.cgi	commons-io.jar file needed for FileUtils variable that allows copy and pasting.
http://www.java2s.com/Code/Jar/s/Downloadseleniumjavajar.htm	selenium-java.jar file needed for web interactions like screenshots.
https://cooltrickshome.blogspot.com/2016/11/take-screenshot-of-any-website-using.html	Example of user creating Maven project that allows process.
https://github.com/csanuragjain/recorder/tree/master/WebsiteScreenshot	GitHub of the previous example listed.
https://chromedriver.storage.googleapis.com/index.html?path=85.0.4183.87/	Driver needed for execution of script. Store in folder called 'driver'
https://jar-download.com/download-handling.php	Link for AShot which proposes full screen of webpage for screenshots.
https://golb.hplar.ch/2019/08/git-with-jgit.html	JGit example with add and commit
https://mkyong.com/swing/java-swing-jfilechooser-example/	Example that shows how to create a window for user file selections.

Setting Up Web Dev Website Testing Automation

This page will host information regarding how to automate website testing using Java and Selenium — and its associated counterparts — to ensure that steps are completed without additional oversight. Truthfully, this process will work as a step towards determining if other uses of this set up will be applicable. Fingers crossed!

Code that should work; limited on query output available at the moment. Plan to test tomorrow. Also need to add wait time based on time taken by web.dev tool to run per site (most likely about 15 - 20 seconds).

For Mac

```
package challenge;

import java.io.File;
```

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import javax.swing.JFileChooser;
import javax.swing.filechooser.FileSystemView;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;


/*
 * Created by Joshua Greenert on 10/12/2020
 *
 * The goal of this script is to take a list, provided by the user, of
websites and
 * cycle through the list by testing each one on the web.dev/measure
website.
 */

public class TestWebsites {

        static WebDriver driver = null;

        public static void main(String[] args) throws IOException,
InterruptedException {


                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> sites = new ArrayList<String>();
                File file;

                String site = "";
                String name = "url";

                // Go through the text file and store all of the
addresses in the list.
                try {

                        // Create a file chooser
                        final JFileChooser jc = new JFileChooser
(FileSystemView.getFileSystemView().getHomeDirectory());
                        jc.setDialogTitle("Select the file with the
websites");
                        jc.setFileSelectionMode(JFileChooser.
FILES_AND_DIRECTORIES);
```

```
                        // In response to a button click:
                        int returnVal = jc.showOpenDialog(null);

                        if (returnVal == JFileChooser.APPROVE_OPTION) {

                                // Set the file from the users
selection and update the path variable.
                                file = jc.getSelectedFile();

                                Scanner myReader = new Scanner(file);

                                // While there is another url, take the
required pieces and add it to the lists.
                                while (myReader.hasNextLine()) {
                                        String data = myReader.
nextLine();

                                        // Check for the https portion
of the string and print that line
                                        if (data.indexOf("https") !=
-1) {
                                                if(data.indexOf(".com")
!= -1) {
                                                        site = data.
substring(data.indexOf("//") + 2, data.indexOf(".com")+4);
                                                        site =
"https://" + site;

                                                        sites.add(site);
                                                }
                                                else if(data.indexOf(".
net") != -1) {
                                                        site = data.
substring(data.indexOf("//") + 2, data.indexOf(".net")+4);
                                                        site =
"https://" + site;

                                                        sites.add(site);
                                                }
                                                else if(data.indexOf(".
org") != -1) {
                                                        site = data.
substring(data.indexOf("//") + 2, data.indexOf(".org")+4);
                                                        site =
"https://" + site;

                                                        sites.add(site);
                                                }
                                                else if(data.indexOf(".
us") !=  -1) {
                                                        site = data.
substring(data.indexOf("//") + 2, data.indexOf(".us")+4);
                                                        site =
```

```
"https://" + site;
                                                        sites.add(site);
                                                }
                                                else {
                                                        continue;
                                                }
                                        }
                                }

                                // For each loop to pass name, site,
and date to capture object.
                                for (int i = 0; i < sites.size(); i++) {
                                        testWebsite(sites.get(i), name);
                                }

                                myReader.close();
                        }
                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                }

        }

        // Initializes the driver.
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite(String site, String name) throws
IOException, InterruptedException {
                initDriver();

                // Set the site for the driver.
                driver.get("https://web.dev/measure/");

                // Find the text input element by its name
                WebElement element = driver.findElement(By.name(name));

                // Enter something to search for
                element.sendKeys(site);
```

```
                // Create a new web element for the button and press
submit.
                //WebElement submitButton = driver.findElement(By.xpath
("//button[contains(.,’Run Audit’)]"));
                WebElement submitButton = driver.findElement(By.id("run-
lh-button"));
                submitButton.click();

                // Wait for 25 seconds to continue.
                TimeUnit.SECONDS.sleep(25);

                // Now submit the form. WebDriver will find the form
for us from the element
                //element.submit();

                // Check the title of the page
                System.out.println("The site " +site+" has been
submitted!");


                destroy();
        }

        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }
}
```

For Windows

```
package selenium;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;


/*
 * Created by Joshua Greenert on 10/12/2020
```

```
 *
 * The goal of this script is to take a list, provided by the user, of
websites and
 * cycle through the list by testing each one on the web.dev/measure
website.
 */

public class WebDevTesting {

        static WebDriver driver = null;

        public static void main(String[] args) throws IOException,
InterruptedException {


                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> sites = new ArrayList<String>();
                File file;

                String site = "";
                String name = "url";

                // Go through the text file and store all of the
addresses in the list.
                try {

                        // Set the file from the users selection and
update the path variable.
                        file = new File("..//..//Documents//GitHub
//webpage_screenshots//webtesting.rtf");

                        Scanner myReader = new Scanner(file);

                        // While there is another url, take the
required pieces and add it to the lists.
                        while (myReader.hasNextLine()) {
                                String data = myReader.nextLine();

                                // Check for the https portion of the
string and print that line
                                if (data.indexOf("https") != -1) {
                                        if(data.indexOf(".com") != -1) {
                                                System.out.println
("Reached");
                                                site = data.substring
(data.indexOf("//") + 2, data.indexOf(".com")+4);
                                                site = "https://" +
site;
                                                sites.add(site);
```

```
                                        }
                                        else if(data.indexOf(".net") !=
-1) {
                                                site = data.substring
(data.indexOf("//") + 2, data.indexOf(".net")+4);
                                                site = "https://" +
site;

                                                sites.add(site);
                                        }
                                        else if(data.indexOf(".org") !=
-1) {
                                                site = data.substring
(data.indexOf("//") + 2, data.indexOf(".org")+4);
                                                site = "https://" +
site;

                                                sites.add(site);
                                        }
                                        else if(data.indexOf(".us") !=
-1) {
                                                site = data.substring
(data.indexOf("//") + 2, data.indexOf(".us")+3);
                                                site = "https://" +
site;

                                                sites.add(site);
                                        }
                                        else {
                                                continue;
                                        }
                                }
                        }


                        // For each loop to pass name, site, and date
to capture object.
                        for (int i = 0; i < sites.size(); i++) {
                                testWebsite(sites.get(i), name);
                        }

                        myReader.close();

                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                }

        }

        // Initializes the driver.
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
```

```
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite(String site, String name) throws
IOException, InterruptedException {
                initDriver();

                // Set the site for the driver.
                driver.get("https://web.dev/measure/");

                // Find the text input element by its name
                WebElement element = driver.findElement(By.name(name));

                // Enter something to search for
                element.sendKeys(site);

                // Create a new web element for the button and press
submit.
                //WebElement submitButton = driver.findElement(By.xpath
("//button[contains(.,'Run Audit')]"));
                WebElement submitButton = driver.findElement(By.id("run-
lh-button"));
                submitButton.click();

                // Wait for 25 seconds to continue.
                TimeUnit.SECONDS.sleep(25);

                // Now submit the form. WebDriver will find the form
for us from the element
                //element.submit();

                // Check the title of the page
                System.out.println("The site " +site+" has been
submitted!");


                destroy();
        }

        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }
```

```
}
```

References	
Link	**Comments**
https://seleniumjava.com/2015/09/13/how-does-selenium-webdriver-work/	Selenium example on how to make post requests by the name of the ID element (JavaScript).
https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/By.html#method.summary	Selenium By class with method options listed and simple explanations.

Test All Subdomains For Issues Or Non-matching References

This page hosts the script created to test all subdomains for cbdrx4u to check for inconsistencies. The process of doing this would, otherwise, take at least an hour or two for all 700 subdomains within.

Code for Mac

```
package challenge;

import java.awt.AWTException;
import java.awt.Robot;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

/*
 * Created by Joshua Greenert 10/23/2020
 *
 * This script will test our list of subdomains to check for errors.
 */
public class TestSubdomains {

        static WebDriver driver = null;
        Robot r;

        public static void main(String[] args) throws IOException,
InterruptedException, AWTException {

                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> domains = new ArrayList<String>();
                ArrayList<String> urls = new ArrayList<String>();

                File domainfile;
```

```
                File urlfile;

                String currentLine;

                domainfile = new File("../../Documents/Github
/webpage_screenshots/subdomains.txt");
                urlfile = new File("../../Documents/Github
/webpage_screenshots/redirects.txt");

                FileReader domainfr = new FileReader(domainfile);
        BufferedReader domainbr = new BufferedReader(domainfr);
        FileReader urlfr = new FileReader(urlfile);
        BufferedReader urlbr = new BufferedReader(urlfr);

                // Add every domain to the list.
                while ((currentLine = domainbr.readLine()) != null) {
                        String data = currentLine;

                        if(data.indexOf("cbdrx4u") != -1) {
                                if(data.indexOf("strokec2") != -1) {
                                        data = data.substring(data.
indexOf("strokec2") + 9, data.length());
                                }

                                data = data.replace
("\\f0\\fs29\\fsmilli14667 ", "");
                                data = "https://" + data;
                                domains.add(data);
                        }
                }

                // Add every url to the list.
                while ((currentLine = urlbr.readLine()) != null) {
                        String data = currentLine;

                        if(data.indexOf("cbdrx4u") != -1 || data.indexOf
("truecbd") != -1) {
                                if(data.indexOf("strokec2") != -1) {
                                        data = data.substring(data.
indexOf("strokec2") + 9, data.length());
                                }

                                data = data.replace
("\\f0\\fs29\\fsmilli14667 ", "");

                                urls.add(data);
                        }

                }
```

```
		// For each loop to pass name, site, and date to
capture object.
		for (int i = 0; i < domains.size(); i++) {
			testWebsite(domains.get(i), urls.get(i));
		}


		domainbr.close();
		urlbr.close();
	}

	private static void testWebsite(String domain, String url)
throws IOException, InterruptedException, AWTException {
		initDriver();

		// Set the site for the driver.
		driver.get(domain);

		// Wait for seconds to continue.
		TimeUnit.SECONDS.sleep(5);

		String redirection = driver.getCurrentUrl();

		if(!(url.equals(redirection))){
			System.out.println("URL: " + url +
"\nREDIRECTION: " + redirection + "\n");
		}

		destroy();

	}

	// Kills the driver so that the webpage is closed.
	public static void destroy() {
		driver.quit();
	}

	// Initializes the driver.
	public static void initDriver() throws IOException {
		System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
		driver = new ChromeDriver();
		driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
		driver.manage().window().maximize();
	}
}
```

Code for Windows

```
package selenium;

import java.awt.AWTException;
import java.awt.Robot;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

/*
 * Created by Joshua Greenert 10/23/2020
 *
 * This script will test our list of subdomains to check for errors.
 */
public class TestSubdomains {

        static WebDriver driver = null;
        static ArrayList<String> differences = new ArrayList<String>();
        Robot r;

        public static void main(String[] args) throws IOException,
InterruptedException, AWTException {

                // Create List object to store URLS and file object to
store user-file.
                ArrayList<String> domains = new ArrayList<String>();
                ArrayList<String> urls = new ArrayList<String>();

                File domainfile;
                File urlfile;

                String currentLine;

                domainfile = new File("..//..//Documents//Github
//webpage_screenshots//subdomains.txt");
                urlfile = new File("..//..//Documents//Github
//webpage_screenshots//redirects.txt");

                FileReader domainfr = new FileReader(domainfile);
        BufferedReader domainbr = new BufferedReader(domainfr);
        FileReader urlfr = new FileReader(urlfile);
        BufferedReader urlbr = new BufferedReader(urlfr);

                // Add every domain to the list.
```

```
                while ((currentLine = domainbr.readLine()) != null) {
                        String data = currentLine;

                        if(data.indexOf("cbdrx4u") != -1) {
                                if(data.indexOf("strokec2") != -1) {
                                        data = data.substring(data.
indexOf("strokec2") + 9, data.length());
                                }

                                data = data.replace
("\\f0\\fs29\\fsmilli14667 ", "");
                                data = "https://" + data;
                                domains.add(data);
                        }
                }

                // Add every url to the list.
                while ((currentLine = urlbr.readLine()) != null) {
                        String data = currentLine;

                        if(data.indexOf("cbdrx4u") != -1 || data.indexOf
("truecbd") != -1) {
                                if(data.indexOf("strokec2") != -1) {
                                        data = data.substring(data.
indexOf("strokec2") + 9, data.length());
                                }

                                data = data.replace
("\\f0\\fs29\\fsmilli14667 ", "");

                                urls.add(data);
                        }

                }

                // For each loop to pass name, site, and date to
capture object.
                for (int i = 0; i < domains.size(); i++) {
                        testWebsite(domains.get(i), urls.get(i));
                }

                // Print all of the list differences.
                for(int i = 0; i < differences.size(); i++) {
                        System.out.println(differences.get(i));
                }

                domainbr.close();
                urlbr.close();
        }
```

```
	private static void testWebsite(String domain, String url)
throws IOException, InterruptedException, AWTException {
		initDriver();

		// Set the site for the driver.
		driver.get(domain);

		// Wait for seconds to continue.
		TimeUnit.SECONDS.sleep(5);

		String redirection = driver.getCurrentUrl();

		if(!(url.equals(redirection))){
			System.out.println("URL: " + url +
"\nREDIRECTION: " + redirection + "\n");
			differences.add("URL: " + url + "\nREDIRECTION:
" + redirection + "\n");
		}

		destroy();

	}

	// Kills the driver so that the webpage is closed.
	public static void destroy() {
		driver.quit();
	}

	// Initializes the driver.
	public static void initDriver() throws IOException {
		System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
		driver = new ChromeDriver();
		driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
		driver.manage().window().maximize();
	}
}
```

Website A HREF List Testing

This script will pull all references of links (whether scripts, hrefs, or src) to print a comprehensive list.

The Code

```
package challenge;


import java.awt.AWTException;
```

```
import java.awt.Robot;
import java.io.IOException;
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

/*
 * Created by Joshua Greenert on 12/08/2020
 *
 * This program will pull up a website and collect a list of valid links
 * that it finds.  This will be done to test whether our clocking
process is accurate or not
 * but may still not be the best method for checking.
 */
public class Website_Email_Test {

        static WebDriver driver = null;
        static Robot r;

        public static void main(String[] args) throws IOException,
InterruptedException, AWTException {

                // Call the function to test the website.
                testWebsite();

        }

        // Initializes the driver.
        public static void initDriver() throws IOException {
                System.setProperty("webdriver.chrome.driver", "driver
/chromedriver.exe");
                driver = new ChromeDriver();
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.
SECONDS);
                driver.manage().window().maximize();
                //driver.manage().window().setPosition(new Point(-2000,
0));   Add this in when completed
        }

        // Creates the driver object, sends the parameter, waits, then
destroys the driver.
        public static void testWebsite() throws IOException,
InterruptedException, AWTException {
                initDriver();

                // Set the site for the driver.
                driver.get("https://truecbd4u.com/find-us/alabama");
```

```
                List<WebElement> list=driver.findElements(By.xpath("//*
[@href or @src]"));

               for(WebElement e : list){
                   String link = e.getAttribute("href");
                   if(null==link)
                       link=e.getAttribute("src");
                   System.out.println(e.getTagName() + " = "  + link);
               }

                destroy();
        }

        // Kills the driver so that the webpage is closed.
        public static void destroy() {
                driver.quit();
        }
}
```

PHP Projects

This page will be the host for all children meeting PHP project specifications. Mainly, this is being created as an alternative repository for code samples to use for future reference.

- 360twin - XML URL Creation For Google Search Sitemap
- Billing.dang-designs - Call Logs Widget
- Billing.dang-designs - ForceEveryoneToLogin
- Billing.dang-designs - GoogleCalendarToCRM
- Billing.dang-designs - CRMToGoogleCalendar
- Billing.dang-designs - _Self Opening Link Hook
- Billing.dang-designs - Send Monthly Report Emails
- Billing.dang-designs - Update CRM with RingCentral Call Logs/Missed Calls
- Capitalcityshoes - Add Customization To Item Description (Ariba)
- Capitalcityshoes - Cookies
- Capitalcityshoes - DMS Users Update Store ID
- Capitalcityshoes - Import Users View & Credit Update
- Capitalcityshoes - Insert Inventory Missing Rows
- Capitalcityshoes - Send Activation Email Automation
- CBDRX4U - Create Folders and .htaccess Files With List
- CBDRX4U - Update All HTML Files To Replace TM with R
- CBDRX4U - Update Zoo Item Links In Database
- Mywatersidesouth - Forgot Username
- Mywatersidesouth - Forgot Password
- Mywatersidesouth - Reset Password
- SunFlora - Setting Up Cloud Storage Bucket From PHP Script
- SunFlora - 3PL API Import

360twin - XML URL Creation For Google Search Sitemap

This page will host the code used to create additional bike URLs not indexed on Google's Search Console. The intent of the script was to have the bikes' shop pages for each bike indexed since they are not being done so on the website currently; however, it was later discovered that the bikes were listed on each page by the URL of "..shop/?make=XXXmodel=XXXyear=XXX" which doesn't include the host domain at the beginning.

The Code

```
<?php
/*
```

```
 * Created by Joshua Greenert on 3/9/2021
 *
 * This script will create xml statements from the list of bikes to add
to the sitemap for 360twin.
 * The process starts by pulling all items into an array, looping
through the array to add the lines
 * to a new document, then using that new document to copy and paste to
the sitemap.
 */

 // Make the array to hold the URLS.
 $urls = array();

 // read the contents of the file to store them
 $urlsfile = fopen("allbikes.txt", "r") or die("unable to open file");

 // Add the contents to the array
 while(!feof($urlsfile)){
     array_push($urls, fgets($urlsfile));
 }

 fclose($urlsfile);

 for($i = 0; $i < count($urls); $i++){

    // Correct newline character.
    $url = str_replace("\n", "",$urls[$i]);
    $xmlstr = "";
    $xmlstr .= "<url>\n\t<loc>$url</loc>\n";
    $xmlstr .= "\t<lastmod>2020-06-29T19:50:13+00:00</lastmod>\n\t";
    $xmlstr .= "<changefreq>daily</changefreq>\n\t<priority>0.5000<
/priority>\n</url>\n";

    file_put_contents("sitemapdata.txt", $xmlstr, FILE_APPEND);
 }

?>
```

Billing.dang-designs - Call Logs Widget

This page will host the code and the pieces assigned to the Call Logs Widget set up on the billing.dang-designs site. The widget script provided below creates the instance of a widget with links that lead the user to a page that shows the data from the database.

Widget Directory - home/modules/widgets
Call Logs script - home/taygray
Missed Calls script - home/taygray

Call Logs Widget Script

```
<?php
```

```
add_hook('AdminHomeWidgets', 1, function() {
    return new CallLogsWidget();
});

class CallLogsWidget extends \WHMCS\Module\AbstractWidget
{
    protected $title = 'RingCentral Call Data';
    protected $description = '';
    protected $weight = 150;
    protected $columns = 1;
    protected $cache = false;
    protected $cacheExpiry = 120;
    protected $requiredPermission = '';

    public function getData()
    {
        return array();
    }

    public function generateOutput($data)
    {
        return <<<EOF
<div class="widget-content-padded" style="text-align: center;">
    <a href="https://billing.dang-designs.com/taygray/call_logs.php"
>Call Logs Page</a>
</div>
<div class="widget-content-padded" style="text-align: center;">
    <a href="https://billing.dang-designs.com/taygray/missed_calls.php"
>Missed Calls Page</a>
</div>
EOF;
    }
}
```

Call Logs Script

```
<?php
/*
 * Created by Joshua Greenert on 12/17/2020
 *
 * This script will provide the logs stored in the database for
RingCentral call
 * information to the view of the widget added to the home screen.
 */
include('../configuration.php');

// Set variables for connection
$servername = "localhost";
```

```
$database = $db_name;
$username = $db_username;
$password = $db_password;

// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// Create the statements to pull the logs.
$query_call_logs = "SELECT * FROM tblringcentral_call_logs order by
startTime desc limit 150";

?>
<!DOCTYPE html>
<html>
<head>
    <style>
    body {
      margin: 0;
    }

    /* Style the header */
    .header {
      background-color: #3D8DBC;
      color: white;
      padding: 20px;
      text-align: center;
    }
    /* Style the top navigation bar */
    .topnav {
      overflow: hidden;
      background-color: #333;
    }

    /* Style the topnav links */
    .topnav a {
      float: left;
      display: block;
      color: #f2f2f2;
      text-align: center;
      padding: 14px 16px;
      text-decoration: none;
    }

    /* Change color on hover */
    .topnav a:hover {
```

```
      background-color: #ddd;
      color: black;
    }
    table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
      padding: 8px;
    }
    th {
        color: white;
        background-color: orange;
        font-weight: bold;
        text-align: center;
    }
    td {
        text-align: center;
    }
    </style>
</head>
<body>

<div class="header">
  <h1>RingCentral Call Logs</h1>
</div>
<div class="topnav">
  <a href="https://billing.dang-designs.com/taygray/index.php">Home</a>
</div>

<table>
  <tr>
    <th>startTime</th>
    <th>fromPhoneNumber</th>
    <th>toPhoneNumber</th>
    <th>direction</th>
    <th>fromName</th>
    <th>toName</th>
    <th>fromExtensionNumber</th>
    <th>toExtensionNumber</th>
    <th>duration</th>
    <th>call_id</th>
  </tr>
<?php
$result1 = $conn->query($query_call_logs);

if($result1->num_rows > 0){
    while($row = $result1->fetch_assoc()){
        echo "<tr>";
        echo "<td>" . $row['startTime'] . "</td>";
        echo "<td>" . $row['fromPhoneNumber'] . "</td>";
        echo "<td>" . $row['toPhoneNumber'] . "</td>";
```

```
        echo "<td>" . $row['direction'] . "</td>";
        echo "<td>" . $row['fromName'] . "</td>";
        echo "<td>" . $row['toName'] . "</td>";
        echo "<td>" . $row['fromExtensionNumber'] . "</td>";
        echo "<td>" . $row['toExtensionNumber'] . "</td>";
        echo "<td>" . $row['duration'] . "</td>";
        echo "<td>" . $row['call_id'] . "</td>";
        echo "</tr>";
    }
}
?>
</table>
</body>
</html>
<?php
mysqli_close($conn);
?>
```

Missed Calls Script

```
<?php
/*
 * Created by Joshua Greenert on 12/17/2020
 *
 * This script will provide the logs stored in the database for
RingCentral call
 * information to the view of the widget added to the home screen.
 */
include('../configuration.php');

// Set variables for connection
$servername = "localhost";
$database = $db_name;
$username = $db_username;
$password = $db_password;

// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// Create the statements to pull the logs.
$query_missed_calls = "SELECT * FROM tblringcentral_missed_calls order
by startTime desc limit 150";
```

```
?>
<!DOCTYPE html>
<html>
<head>
    <style>
    body {
      margin: 0;
    }

    /* Style the header */
    .header {
      background-color: #3D8DBC;
      color: white;
      padding: 20px;
      text-align: center;
    }
    /* Style the top navigation bar */
    .topnav {
      overflow: hidden;
      background-color: #333;
    }

    /* Style the topnav links */
    .topnav a {
      float: left;
      display: block;
      color: #f2f2f2;
      text-align: center;
      padding: 14px 16px;
      text-decoration: none;
    }

    /* Change color on hover */
    .topnav a:hover {
      background-color: #ddd;
      color: black;
    }
    table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
      padding: 8px;
    }
    th {
        color: white;
        background-color: orange;
        font-weight: bold;
        text-align: center;
    }
    td {
        text-align: center;
```

```
    }
    </style>
</head>
<body>

<div class="header">
  <h1>RingCentral Missed Calls</h1>
</div>
<div class="topnav">
  <a href="https://billing.dang-designs.com/taygray/index.php">Home</a>
</div>

<table>
  <tr>
    <th>startTime</th>
    <th>fromPhoneNumber</th>
    <th>toPhoneNumber</th>
    <th>direction</th>
    <th>fromName</th>
    <th>toName</th>
    <th>fromExtensionNumber</th>
    <th>toExtensionNumber</th>
    <th>duration</th>
    <th>call_id</th>
  </tr>
<?php
$result1 = $conn->query($query_missed_calls);

if($result1->num_rows > 0){
    while($row = $result1->fetch_assoc()){
        echo "<tr>";
        echo "<td>" . $row['startTime'] . "</td>";
        echo "<td>" . $row['fromPhoneNumber'] . "</td>";
        echo "<td>" . $row['toPhoneNumber'] . "</td>";
        echo "<td>" . $row['direction'] . "</td>";
        echo "<td>" . $row['fromName'] . "</td>";
        echo "<td>" . $row['toName'] . "</td>";
        echo "<td>" . $row['fromExtensionNumber'] . "</td>";
        echo "<td>" . $row['toExtensionNumber'] . "</td>";
        echo "<td>" . $row['duration'] . "</td>";
        echo "<td>" . $row['call_id'] . "</td>";
        echo "</tr>";
    }
}
?>
</table>
</body>
</html>
<?php
mysqli_close($conn);
```

```
?>
```

Billing.dang-designs - ForceEveryoneToLogin

After the update for WHMCS, the forceeveryonetologin.php script is no longer actively working properly. To correct this, we removed all pages listed within the in_array call and replaced them with “index” (index.php). Additionally, to check that the user is at the right location, since the index page is used for most links but concatenated, we send any request from a non-logged in user to the Login page.

Located in: public_html/includes/hooks/

The Code

```
<?php

if (!defined("WHMCS"))
   die("This file cannot be accessed directly");

function hook_ForceEveryoneToLogin($vars) {

   $clientID = intval($_SESSION['uid']);
   $adminID = intval($_SESSION['adminid']);

   // Create the link for the incoming URL.
    $link = "https";
    $link .= "://";
    $link .= $_SERVER['HTTP_HOST'];
    $link .= $_SERVER['REQUEST_URI'];

   if ($adminID===0){
       if (!in_array($vars['filename'], array("index", "pwreset",
"contact", "submitticket")) && $clientID===0){
            // Check if the link is accurate.
            if($link != "https://billing.dang-designs.com/index.php?rp=
/login" && $link != "https://billing.dang-designs.com/index.php?rp=
/password/reset" &&
                $link != "https://billing.dang-designs.com/index.php?rp=
/password/reset/change" && $link != "https://billing.dang-designs.com
/contact.php" &&
                $link != "https://billing.dang-designs.com/register.
php" && $link != "https://billing.dang-designs.com/allpolicies.html"){
                header("Location: login.php");
                exit;
            }
       }
       else if($clientID===0){

           // Check if the link is accurate.
           if($link != "https://billing.dang-designs.com/index.php?rp=
/login" && $link != "https://billing.dang-designs.com/index.php?rp=
/password/reset" &&
                $link != "https://billing.dang-designs.com/index.php?rp=
```

```
/password/reset/change" && $link != "https://billing.dang-designs.com
/contact.php" &&
                $link != "https://billing.dang-designs.com/register.
php" && $link != "https://billing.dang-designs.com/allpolicies.html"){
                header("Location: login.php");
                exit;
            }
        }
    }

}
add_hook("ClientAreaPage", 1, "hook_ForceEveryoneToLogin");
```

Billing.dang-designs - GoogleCalendarToCRM

This page will host the script for the support Google account that ensures when a new event is added in the appropriate way, the event is then created in the CRM. The process uses an oauth credential set from Google to function properly and the same credentials cannot be used for two different files (one set of oauth credentials per file).

The Code

```
<?php
/*
 * Created by Joshua Greenert on 3/29/2021
 *
 * This program will fetch all upcoming events from the support Google
Calendar, check them against the table
 * in CRM for billing.dang-designs, and insert the ones that don't
exist in the proper format.
 *
 * TABLES: tblcalendar, crm_followups, crm_resources
 */
require '../../../vendor/autoload.php';

// Define the STDIN to supress errors.
define('STDIN',fopen("php://stdin","r"));
include('../../configuration.php');

// Create connection
$conn = new mysqli($db_host, $db_username, $db_password, $db_name);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Variables to use.
$userid;
$username;
$email;
$meetingtype;
$type_id;
```

```
$admin_id = 14;

/**
 * Returns an authorized API client.
 * @return Google_Client the authorized client object
 */
function getClient($authCode)
{
    $client = new Google_Client();
    $client->setApplicationName('Google Calendar API PHP Quickstart');
    $client->setRedirectUri('http://' . $_SERVER['HTTP_HOST'] . '
/GoogleCalendarToCRM.php');
    $client->setScopes(Google_Service_Calendar::CALENDAR);
    $client->setAuthConfig('credentials.json');
    $client->setAccessType('offline');
    $client->setPrompt('select_account consent');

    // Load previously authorized token from a file, if it exists.
    // The file token.json stores the user's access and refresh tokens,
and is
    // created automatically when the authorization flow completes for
the first
    // time.
    $tokenPath = 'token.json';
    if (file_exists($tokenPath)) {
        $accessToken = json_decode(file_get_contents($tokenPath), true);
        $client->setAccessToken($accessToken);
    }

    // If there is no previous token or it's expired.
    if ($client->isAccessTokenExpired()) {
        // Refresh the token if possible, else fetch a new one.
        if ($client->getRefreshToken()) {
            $client->fetchAccessTokenWithRefreshToken($client-
>getRefreshToken());
        } else {

            // Check if the auth code exists already and has been sent
to the browser for this call.
            if($authCode == "" && !isset($_GET['code'])){
                // Request authorization from the user.
                $authUrl = $client->createAuthUrl();
                printf("Open the following link in your browser:\n%
s\n", $authUrl);
                print 'Enter verification code: ';
                $authCode = trim(fgets(STDIN));
            }

            // Exchange authorization code for an access token.
            $accessToken = $client->fetchAccessTokenWithAuthCode
```

```
($authCode);
            $client->setAccessToken($accessToken);

            // Check to see if there was an error.
            if (array_key_exists('error', $accessToken)) {
                throw new Exception(join(', ', $accessToken));
            }
        }
        // Save the token to a file.
        if (!file_exists(dirname($tokenPath))) {
            mkdir(dirname($tokenPath), 0700, true);
        }
        file_put_contents($tokenPath, json_encode($client-
>getAccessToken()));
    }
    return $client;
}

// Check the post for information
if(!isset($_GET['code'])){

    // Set the auth code to nothing to begin the execution.
    $authCode = "";

    // Get the API client and construct the service object.
    $client = getClient($authCode);
    $service = new Google_Service_Calendar($client);

    // Print the next 10 events on the user's calendar.
    $calendarId = 'primary';
    $optParams = array(
    'maxResults' => 100,
    'orderBy' => 'startTime',
    'singleEvents' => true,
    'timeMin' => date('c'),
    );
    $results = $service->events->listEvents($calendarId, $optParams);
    $events = $results->getItems();

    if (empty($events)) {
        print "No upcoming events found.\n";
    } else {
        print "Upcoming events:\n";
        foreach ($events as $event) {
            $start = $event->start->dateTime;
            if (empty($start)) {
                $start = $event->start->date;
            }

            // Set the call or meeting; default is call
```

```
            $meetingtype = (strpos(strtolower($event->getSummary()),
"meeting") != false) ? "meeting" : "call";
            $type_id = ($meetingtype == "call") ? 2 : 1;

            // Strip the date and time from the event object. MAY NOT
BE NEEDED
            $date = substr($start, 0, strpos($start, "T"));
            $time = substr($start, strpos($start, "T") + 1, 8);
            $eventdatemod = $date . " " . $time;

            // Set the start time numeric parse string that goes into
the calendar table.
            // Get the event details needed for the transfer.  date("Y-
m-d H:i:s");
            $gcalendarid = strval($event->getId());
            $gcalendartitle = $event->getSummary();
            $gdescription = $event->getDescription();
            $gstarttime = $eventdatemod;
            $crmstarttime = date("Y-m-d H:i:s");
            $tblstarttime = strtotime($time);

            // Get the email listed in the description using regex.
            preg_match_all("/[\._a-zA-Z0-9-+]+@[\._a-zA-Z0-9-]+/i",
$gdescription, $matches);
            foreach($matches as $match){
                foreach($match as $mat){
                    $email = strval($mat);
                }
            }

            // Now use the connection to check the table for the
calendar id, if it's not found enter it.
            $querycalendarids = "Select * from tblcalendar where
calendar_id is not null and calendar_id = '$gcalendarid'";
            $resultcalendarcheck = $conn->query($querycalendarids);

            // We expect this to be null, so that the else statement is
what inserts the new event.
            if($resultcalendarcheck->num_rows > 0){
                // There was a record with that id; don't enter it
again.
                continue;
            }
            else{

                // Check the users email and find the name to insert at
a later step; need name and last name put together and user's id.
                $queryuseremails = "Select id, name, lastname from
crm_resources where email = '$email'";
                $resultusernamecheck = $conn->query($queryuseremails);
```

```
                if($resultusernamecheck->num_rows > 0){
                    while($row = $resultusernamecheck->fetch_assoc()){

                        $userid = $row['id'];
                        $username = $row['name'] . " " . $row
['lastname'];
                        $title = "CRM - " . strval($username);

                        // Insert into the crm follow ups table to use
the id later to insert into the description table.
                        $insertcrmtable = "Insert into crm_followups
(resource_id, type_id, admin_id, description, `date`, created_at,
updated_at, deleted_at)
                            values ($userid, $type_id, $admin_id,
'$gdescription', '$gstarttime', '$crmstarttime', '$crmstarttime',
NULL)";
                        if($conn->query($insertcrmtable) === TRUE){
                            //echo "new record added";

                            // Need to set up piece to add into the
desc field for URL.
                            $querynewCRM = "Select id from
crm_followups where created_at = '$crmstarttime'";
                            $resultnewCRM = $conn->query($querynewCRM);

                            if($resultnewCRM->num_rows > 0){
                                while($row2 = $resultnewCRM-
>fetch_assoc()){

                                    // Get the id and set the url to
place into the description field.
                                    $crm_id = $row2['id'];
                                    $crm_url = " -( https://billing.
dang-designs.com/taygray/crm.php#!/contacts/$userid/followup/$crm_id )";
                                    $gdescription .= $crm_url;

                                    $insertcalendartable = "Insert into
tblcalendar (title, `desc`, start, end, allday, adminid, recurid,
calendar_id) values
                                        ('$title', '$gdescription',
'$tblstarttime', 0, 1, $admin_id, 0, '$gcalendarid')";
                                    if($conn->query
($insertcalendartable) === TRUE){
                                        echo "new record added";
                                    }
                                    else {
                                        echo "Error: " . $sql . "<br>"
. $conn->error;
                                    }
```

```
                            }
                        }
                    }
                    else {
                        echo "Error: " . $sql . "<br>" . $conn-
>error;
                    }
                }
            }
        }
    }
}

    // Don't forget to close the connection.
    $conn->close();
}
else{

    // Get the auth code and store it for the authorization execution.
    $authCode = $_GET['code'];

    // Get the API client and construct the service object.
    $client = getClient($authCode);
    $service = new Google_Service_Calendar($client);

    print "Account Link Successful!\n";
    // Print the next 10 events on the user's calendar.
    // Left in for proof of life after initialization.
    $calendarId = 'primary';
    $optParams = array(
    'maxResults' => 10,
    'orderBy' => 'startTime',
    'singleEvents' => true,
    'timeMin' => date('c'),
    );
    $results = $service->events->listEvents($calendarId, $optParams);
    $events = $results->getItems();

    if (empty($events)) {
        print "No upcoming events found.\n";
    } else {
        print "Upcoming events:\n";
        foreach ($events as $event) {
            $start = $event->start->dateTime;
            if (empty($start)) {
                $start = $event->start->date;
            }
            printf("%s (%s)\n", $event->getSummary(), $start);
        }
    }
```

```
}

?>
```

Billing.dang-designs - CRMToGoogleCalendar

This page holds the script created to pull new follow up events with clients over to the support Google account's Google Calendar to create an event. It operates by looking for records created within the past 5 minutes so that actions are not taken on the previous records that were not recorded.

The Code

```
<?php
/*
 * Created by Joshua Greenert on 3/30/2021
 *
 * This script will pull data from the CRM_followups table if the time
of the created checked is
 * within 5 minutes of it's execution.   The resources will be pulled
from the tables as long as
 * the fields for the calendar_id field is confirmed as null, placed
into an array to insert into
 * the event, and pushed to the calendar.  Then, the calendar will be
called again to get the id;
 *
 * NOTE: This calendar has to be fast enough after the event has been
sent to it to not incur a
 *       second event creation by the first script.
 *
 * TABLES: crm_followups, tblcalendar, crm_resources
 */
require '../../../vendor/autoload.php';
// Define the STDIN to supress errors.
define('STDIN',fopen("php://stdin","r"));
include('../../configuration.php');

// Create connection
$conn = new mysqli($db_host, $db_username, $db_password, $db_name);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Google Calendar Function call
function getClient()
{
    $client = new Google_Client();
    $client->setApplicationName('Google Calendar API New Event');
    $client->setRedirectUri('http://' . $_SERVER['HTTP_HOST'] . '
/CRMToGoogleCalendar.php');
    $client->setScopes(Google_Service_Calendar::CALENDAR);
```

```
    $client->setAuthConfig('credentials.json');
    $client->setAccessType('offline');
    $client->setPrompt('select_account consent');

    // Load previously authorized token from a file, if it exists.
    // The file token.json stores the user's access and refresh tokens,
and is
    // created automatically when the authorization flow completes for
the first
    // time.
    $tokenPath = 'token.json';
    if (file_exists($tokenPath)) {
        $accessToken = json_decode(file_get_contents($tokenPath), true);
        $client->setAccessToken($accessToken);
    }

    // If there is no previous token or it's expired.
    if ($client->isAccessTokenExpired()) {
        // Refresh the token if possible, else fetch a new one.
        if ($client->getRefreshToken()) {
            $client->fetchAccessTokenWithRefreshToken($client-
>getRefreshToken());
        } else {

            // Check if the auth code exists already and has been sent
to the browser for this call.
            if($authCode == "" && !isset($_GET['code'])){
                // Request authorization from the user.
                $authUrl = $client->createAuthUrl();
                printf("Open the following link in your browser:\n%
s\n", $authUrl);
                print 'Enter verification code: ';
                $authCode = trim(fgets(STDIN));
            }

            // Exchange authorization code for an access token.
            $accessToken = $client->fetchAccessTokenWithAuthCode
($authCode);
            $client->setAccessToken($accessToken);

            // Check to see if there was an error.
            if (array_key_exists('error', $accessToken)) {
                throw new Exception(join(', ', $accessToken));
            }
        }
        // Save the token to a file.
        if (!file_exists(dirname($tokenPath))) {
            mkdir(dirname($tokenPath), 0700, true);
        }
        file_put_contents($tokenPath, json_encode($client-
```

```
>getAccessToken()));
    }
    return $client;
}

// Check the post for information
if(!isset($_GET['code'])){

    // Pull data from crm table that has been created within 5 minutes
    $getRecentQuery = "SELECT * FROM crm_followups where created_at >
(NOW() - INTERVAL 245 MINUTE) and deleted_at is NULL limit 1";
    $resultGetRecent = $conn->query($getRecentQuery);

    if($resultGetRecent->num_rows > 0){
        while($row = $resultGetRecent->fetch_assoc()){

            // Add in the variables needed for later.
            $crm_id = $row['id'];
            $resource_id = $row['resource_id'];
            $type = $row['type_id'] == 2 ? "Call" : "Meeting";
            $description = $row['description'];
            $event_date = $row['date'];

            $firstname;
            $lastname;
            $email;
            $starttime;
            $event_title;
            $timezone;
            $datetime;

            // Use the resource id to capture to get user's name, last
name, and email from resources table.
            $getUserDataQuery = "SELECT name, email, lastname FROM
crm_resources where id = $resource_id";
            $resultGetUserData = $conn->query($getUserDataQuery);

            if($resultGetUserData->num_rows > 0){
                while($row2 = $resultGetUserData->fetch_assoc()){

                    // Set the user data variables and the description
variable to look for.
                    $firstname = $row2['name'];
                    $lastname = $row2['lastname'];
                    $email = $row2['email'];

                    // Use description values to pull query and check
calendar_id field.
                    $getStartTimeQuery = "SELECT start FROM tblcalendar
where `desc` like '%$description%' and calendar_id is null";
```

```
                    $resultGetStartTime = $conn->query
($getStartTimeQuery);

                    if($resultGetStartTime->num_rows > 0){
                        while($row3 = $resultGetStartTime-
>fetch_assoc()){

                            // Convert start time back to numeric time.
                            $starttime = date('H:i:s', $row3['start']);
                            $endtime = date('H:i:s', (strtotime
($starttime) + 60*60));

                            // Set event summary title, timezone.
                            $event_title = 'CRM - EVENT (' .$type. ' -
' .$firstname. ' ' .$lastname. ')';
                            $timezone = 'America/New_York';

                            // Set the datetime object to be
dynamically entered.
                            $datetime = str_replace(" ", "T",
$event_date);
                            $endtime = substr($datetime, 0, 10) . "T".
$endtime;

                            // Get the API client and construct the
service object.
                            $client = getClient($authCode);
                            $service = new Google_Service_Calendar
($client);
                            $calendarId = 'primary';

                            print "Account Link Successful!\n";
                            // Print the next 10 events on the user's
calendar.
                            // Left in for proof of life after
initialization.
                            $optParams = array(
                            'maxResults' => 10,
                            'orderBy' => 'startTime',
                            'singleEvents' => true,
                            'timeMin' => date('c'),
                            );

                            $results = $service->events->listEvents
($calendarId, $optParams);
                            $events = $results->getItems();

                            try{
                                $event = new
Google_Service_Calendar_Event(array(
```

```
                                'summary' => $event_title,
                                'description' =>  $description,
                                'start' => array(
                                'dateTime' => $datetime,
                                'timeZone' => $timezone,
                                ),
                                'end' => array(
                                'dateTime' => $endtime,
                                'timeZone' => $timezone,
                                ),
                                'reminders' => array(
                                'useDefault' => TRUE,
                                ),
                            ));

                            $event = $service->events->insert
($calendarId, $event);
                            printf('Event created: %s\n', $event-
>htmlLink);
                        }
                        catch(Exception $e){
                            echo 'Caught exception: ',  $e-
>getMessage(), "\n";
                        }

                        // Update the event in CRM with the new
Google Calendar Event id
                        // Get the API client and construct the
service object.
                        $client = getClient();
                        $service = new Google_Service_Calendar
($client);


                        print "Account Link Successful!\n";
                        // Print the next 10 events on the user's
calendar.
                        // Left in for proof of life after
initialization.
                        $optParams = array(
                        'maxResults' => 10,
                        'orderBy' => 'startTime',
                        'singleEvents' => true,
                        'timeMin' => date('c'),
                        );
                        $results = $service->events->listEvents
($calendarId, $optParams);
                        $events = $results->getItems();

                        if (empty($events)) {
                            print "No upcoming events found.\n";
```

```
                            } else {
                                print "Upcoming events:\n";
                                foreach ($events as $event) {
                                    $start = $event->start->dateTime;
                                    if (empty($start)) {
                                        $start = $event->start->date;
                                    }

                                    // Get the calendarid and set it
                                    $gcalendarid = strval($event-
>getId());

                                    // Confirm that the record exists
to update the calendar ID to.
                                    $getIDCalendarIDInsert = "Select id
from tblcalendar where `desc` like '%$description%' and calendar_id is
null and '$gcalendarid' not in (select calendar_id from tblcalendar
where calendar_id is not null) ";
                                    $resultGetID = $conn->query
($getIDCalendarIDInsert);

                                    if($resultGetID->num_rows > 0){
                                        while($row4 = $resultGetID-
>fetch_assoc()){

                                            $recordID = $row4['id'];

                                            // Place the google
calendar id in the CRM table row.
                                            $insertGoogleID = "Update
tblcalendar set calendar_id = '$gcalendarid' where id = $recordID";
                                            if($conn->query
($insertGoogleID) === TRUE){
                                                echo "Google Calendar
ID updated!";
                                            }
                                            else {
                                                echo "Error: " . $sql .
"<br>" . $conn->error;
                                            }
                                        }
                                    }
                                    else{ echo "Description doesn't
match, calendar ID is already entered, or calendar_id field is not
null!"; }

                                }
                            }
```

```
                        }
                    }
                    else{
                        echo "No descriptions match or calendar_id
fields are not null " .$description;
                        continue;
                    }
                }
            }
            else{ echo "User not found or query was empty!"; }
        }
    }
    else{ echo "No recent records exist!"; }
}
else{

    // Get the auth code and store it for the authorization execution.
    $authCode = $_GET['code'];

    // Get the API client and construct the service object.
    $client = getClient();
    $service = new Google_Service_Calendar($client);

    print "Account Link Successful!\n";
    // Print the next 10 events on the user's calendar.
    // Left in for proof of life after initialization.
    $calendarId = 'primary';
    $optParams = array(
    'maxResults' => 10,
    'orderBy' => 'startTime',
    'singleEvents' => true,
    'timeMin' => date('c'),
    );
    $results = $service->events->listEvents($calendarId, $optParams);
    $events = $results->getItems();

    if (empty($events)) {
        print "No upcoming events found.\n";
    } else {
        print "Upcoming events:\n";
        foreach ($events as $event) {
            $start = $event->start->dateTime;
            if (empty($start)) {
                $start = $event->start->date;
            }
            printf("%s (%s)\n", $event->getSummary(), $start);
        }
    }
}
```

```
// Don't forget to close the connection.
$conn->close();
?>
```

Billing.dang-designs - _Self Opening Link Hook

The scripts listed below ensure that the admin users, when using the CRM application, only have the links opened within the same window. No separate tabs should be made when using the redirects internally. Since there are two specific locations to add this to, both scripts will be listed here for simplicity.

Code For the Client Area

```
<?php

# Login As Owner Open In New Tab Hook
# Written by brian!
// Adjusted by Joshua Greenert on 12/21/2020.  Instead of the hook operating
// in the way that it was designed, it will be set to open in the same page.

function CRMSameTabClientArea($vars) {

        if ($vars['filename'] == 'clientssummary' && isset($_GET["userid"])) {
                return '<script>$("a.btn.btn-primary.btn-small.btn-xs").prop("target","_self");</script>';
        }
}
add_hook("AdminAreaFooterOutput", 1, "CRMSameTabClientArea");
```

Billing.dang-designs - Send Monthly Report Emails

This page hosts two pieces that are used to automate the report-sending process for monthly reports. The first piece is the text file that hosts the email template to use when sending the email out to clients. The second piece contains the code that, when activated, captures the content in a variable, performs a query for user report files that meet the date requirements, loops through the list to rename the PDF in the dangbills/attachments folder location as well as in the database, then sends the email with the attachment.

The Text File

```
<html>
<head>
  <style type="text/css">
      .ExternalClass,.ExternalClass div,.ExternalClass font,.ExternalClass p,.ExternalClass span,.ExternalClass td,h1,img{line-height:100%}h1,h2{display:block;font-family:Helvetica;font-style:normal;font-weight:700}#outlook a{padding:0}.ExternalClass,.ReadMsgBody{width:100%}a,blockquote,body,li,p,table,td{-webkit-text-size-adjust:100%;-ms-text-size-adjust:100%}table,td{mso-table-lspace:0;mso-table-rspace:0}img
```

```
{-ms-interpolation-mode:bicubic;border:0;height:auto;outline:0;text-
decoration:none}table{border-collapse:collapse!important}#bodyCell,
#bodyTable,body{height:100%!important;margin:0;padding:0;width:100%!
important}#bodyCell{padding:20px;}#templateContainer{width:600px;border:
1px solid #ddd;background-color:#fff}#bodyTable,body{background-color:
#FAFAFA}h1{color:#202020!important;font-size:26px;letter-spacing:normal;
text-align:left;margin:0 0 10px}h2{color:#404040!important;font-size:
20px;line-height:100%;letter-spacing:normal;text-align:left;margin:0 0
10px}h3,h4{display:block;font-style:italic;font-weight:400;letter-
spacing:normal;text-align:left;margin:0 0 10px;font-family:Helvetica;
line-height:100%}h3{color:#606060!important;font-size:16px}h4{color:
grey!important;font-size:14px}.headerContent{background-color:#f8f8f8;
border-bottom:1px solid #ddd;color:#505050;font-family:Helvetica;font-
size:20px;font-weight:700;line-height:100%;text-align:left;vertical-
align:middle;padding:0}.bodyContent,.footerContent{font-family:
Helvetica;line-height:150%;text-align:left;}.footerContent{text-align:
center}.bodyContent pre{padding:15px;background-color:#444;color:
#f8f8f8;border:0}.bodyContent pre code{white-space:pre;word-break:
normal;word-wrap:normal}.bodyContent table{margin:10px 0;background-
color:#fff;border:1px solid #ddd}.bodyContent table th{padding:4px 10px;
background-color:#f8f8f8;border:1px solid #ddd;font-weight:700;text-
align:center}.bodyContent table td{padding:3px 8px;border:1px solid
#ddd}.table-responsive{border:0}.bodyContent a{word-break:break-all}.
headerContent a .yshortcuts,.headerContent a:link,.headerContent a:
visited{color:#1f5d8c;font-weight:400;text-decoration:underline}
#headerImage{height:auto;max-width:600px;padding:0px}#templateBody
{background-color:#fff}.bodyContent{color:#505050;font-size:14px;
padding:20px}.bodyContent a .yshortcuts,.bodyContent a:link,.
bodyContent a:visited{color:#1f5d8c;font-weight:400;text-decoration:
underline}.bodyContent a:hover{text-decoration:none}.bodyContent img
{display:inline;height:auto;max-width:560px}.footerContent{color:grey;
font-size:12px;padding:20px}.footerContent a .yshortcuts,.footerContent
a span,.footerContent a:link,.footerContent a:visited{color:#606060;
font-weight:400;text-decoration:underline}@media only screen and (max-
width:640px){h1,h2,h3,h4{line-height:100%!important}#templateContainer
{max-width:600px!important;width:100%!important}#templateContainer,body
{width:100%!important}a,blockquote,body,li,p,table,td{-webkit-text-size-
adjust:none!important}body{min-width:100%!important}#bodyCell{padding:
10px!important}h1{font-size:24px!important}h2{font-size:20px!important}
h3{font-size:18px!important}h4{font-size:16px!important}
#templatePreheader{display:none!important}.headerContent{font-size:20px!
important;line-height:125%!important}.footerContent{font-size:14px!
important;line-height:115%!important}.footerContent a{display:block!
important}.hide-mobile{display:none;}}
  </style>
</head>
<body>
<center>
            <table align="center" border="0" cellpadding="0"
cellspacing="0" height="100%" width="100%" id="bodyTable">
```

```
                <tr>
                    <td align="center" valign="top" id="bodyCell">
                        <table border="0" cellpadding="0" cellspacing="
0" id="templateContainer">
                            <tr>
                                <td align="center" valign="top">
                                    <table border="0" cellpadding="0"
cellspacing="0" width="100%" id="templateHeader">
                                        <tr>
                                            <td align="left" valign="
top" class="headerContent">
                                                <a href="billing.dang-
designs.com">
                                                    <img src="
https://ci5.googleusercontent.com/proxy
/7drEPRo6gQbWQ3zxSfT8hDR7CH3BnkaqQYQK-
RYRj2HgVZYZZr1Ek0OsNkAH6scaQmPeX_jWvGHJVa3Nobs-005wNEjoYDapFjp6_rk=s0-d-
e1-ft#https://dang-designs.com/images/logos/DANG-Logo-2000px.jpg"
style="max-width:600px" id="headerImage" alt="DANG designs" />
                                            </td>
                                        </tr>
                                    </table>
                                </td>
                            </tr>
                            <tr>
                                <td align="center" valign="top">
                                    <table border="0" cellpadding="0"
cellspacing="0" width="100%" id="templateBody">
                                        <tr>
                                            <td valign="top" class="
bodyContent">
<p><br />Hello,</p>
<div class="gmail_default"><span style="font-family: arial, sans-serif;"
>You have a new report available from DANG designs!<br />The report is
attached and is also available for download in your <a href="
billing.dang-designs.com" target="_blank" rel="nofollow" data-
saferedirecturl=”https://www.google.com/url?q=https://billing.dang-
designs.
com&source=gmail&ust=1624548988820000&usg=AFQjCNHK2ub5twuyhcF0VSG3w3igf4
Zc3Q”>Client Billing Account Profile</a>.<br /></span></div>
<div class="gmail_default"><span style="font-family: arial, sans-serif;"
>As always, our team is here to answer questions and help you with
your online presence. </span></div>
<p><span style="font-family: arial, sans-serif;">Call us at: 813-418-
7592 or email <a href="mailto:service@dang-designs.com" target="
_blank">service@dang-designs.com</a> with any questions. <
/span></p>
<div class="gmail_default"><span style="font-family: arial, sans-serif;"
> </span></div>
<div class="gmail_default"><span style="font-family: arial, sans-serif;"
```

```
>The DANG designs team</span></div>

            </td>
                                        </tr>
                                    </table>
                                </td>
                            </tr>
                            <tr>
                                <td align="center" valign="top">
                                    <table border="0" cellpadding="0"
cellspacing="0" width="100%" id="templateFooter">
                                        <tr>
                                            <td valign="top" class="
footerContent">
                                                 <a href="https://dang-
designs.com">visit our website</a>
                                                <span class="hide-
mobile"> | </span>
                                                <a href="
https://billing.dang-designs.com">log in to your account</a>
                                                 <br />
                                                Copyright © DANG
designs, Inc., All rights reserved.
                                            </td>
                                        </tr>
                                    </table>
                                </td>
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
        </center>
</body>
</html>
```

The Code

```
<?php
/*
 * Created by Joshua Greenert on 6/23/2021
 *
 * This function will query the database to capture the most recent
filename submitted for a user,
 * then search the attachments directory for the file to attach to the
email to send out from the
 * email templated created at
 * https://docs.google.com/document/d/1aiZR_N-y1-L3bwGgJkOpA6R2ZX-
```

```
zo2IS1LntWx6gtJA/edit
 *
 */

// Define the STDIN to supress errors.
define('STDIN',fopen("php://stdin","r"));
include('../../configuration.php');

// Create connection
$conn = new mysqli($db_host, $db_username, $db_password, $db_name);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Function used to create and send email.  Required params are email,
file path, and filename
function sendMonthlyReportEmail($email, $filepath, $filename, $bodytext)
{

    $file = $filepath;
    $subject = 'New Report From DANG designs';
    $from_name = 'DANG designs, Inc.';
    $from_mail = 'dang@dang-designs.com';

    $content = file_get_contents( $file);
    $content = chunk_split(base64_encode($content));
    $uid = md5(uniqid(time()));
    $name = basename($file);

    // header
    $header = "From: <".$from_mail.">\r\n";
    $header .= "Reply-To: ".$from_mail."\r\n";
    $header .= "MIME-Version: 1.0\r\n";
    $header .= "Content-Type: multipart/mixed; boundary=\"".$uid."\"
\r\n\r\n";

    // message & attachment
    $nmessage = "--".$uid."\r\n";
    $headers  = 'MIME-Version: 1.0' . "\r\n";
    $nmessage .= "Content-type:text/html; charset=iso-8859-1\r\n";
    $nmessage .= "Content-Transfer-Encoding: 7bit\r\n\r\n";
    $nmessage .= $bodytext."\r\n\r\n";
    $nmessage .= "--".$uid."\r\n";
    $nmessage .= "Content-Type: application/octet-stream;
name=\"".$filename."\"\r\n";
    $nmessage .= "Content-Transfer-Encoding: base64\r\n";
    $nmessage .= "Content-Disposition: attachment;
filename=\"".$filename."\"\r\n\r\n";
    $nmessage .= $content."\r\n\r\n";
    $nmessage .= "--".$uid."--";
```

```
    if (mail($email, $subject, $nmessage, $header)) {
        echo "This was true";
    } else {
        echo "mail send ... ERROR!";
        print_r( error_get_last() );
    }
}

// Set the body of the email since that piece will always be the same
once set.
$bodytext = file_get_contents(strval('emailTemplate.txt'));

// Set the variables to use throughout the process.
$todayYear = date("Y");
$todayMonth = date("m");

// Set the SQL query for the connection.
$query_tbljoin_attachments = "SELECT tblclients.id as 'client_id',
tblclients.email, tblclientsfiles.id, filename, dateadded FROM
tblclientsfiles
    join tblclients on tblclients.id = tblclientsfiles.userid
    where tblclients.id = 113
    ORDER BY tblclientsfiles.dateadded desc";

// Connect to the table with the query and navigate through the records.
$resultgetclientattachments = $conn->query($query_tbljoin_attachments);

if($resultgetclientattachments->num_rows > 0){
    while($row = $resultgetclientattachments->fetch_assoc()){

        // Get the date object from the query and compare it with the
current month and year.
        $addedDate = $row['dateadded'];
        $addedDateYear = date('Y', strtotime($addedDate));
        $addedDateMonth = date('m', strtotime($addedDate));

        if($addedDateYear == $todayYear && $addedDateMonth ==
$todayMonth){

            // Set the items necessary to ensure that the email can be
sent.
            $id = $row['id'];
            $email = $row['email'];
            $filename = $row['filename'];
            $filepath = "../../../attachments/" . $filename;

            // Rename the file to a more friendly name for users, then
update the database to have the correct file.
            $newFilename = substr($filename, strpos($filename, "_") +1,
```

```
strlen($filename));
            $newFilename = substr($newFilename, 0, strpos($newFilename,
".")). "_" .$todayMonth. "_" .$todayYear. ".pdf";
            $newFilepath = "../../../attachments/" . $newFilename;

            rename($filepath, $newFilepath);

            // Update the database record to match
            $query_update_filename = "update tblclientsfiles set
filename = '$newFilename' where id = $id";

            if ($conn->query($query_update_filename) === TRUE) {
                echo "Filename updated successfully!";
            } else {
                echo "ERROR: " . $conn->error;
            }

            // Send the email with the newly named attachment.
            sendMonthlyReportEmail($email, $newFilepath, $newFilename,
$bodytext);
        }
        else{
            break;
        }
    }
}
?>
```

Billing.dang-designs - Update CRM with RingCentral Call Logs/Missed Calls

This page will host the script that is stored on billing.dang-designs.com that updates the CRM call logs for users that call in. The file for this script is located in public_html/taygray/callLogs/RCcallLogs.php

For future reference, ensure that the proper quotes are used (i.e. "" instead of " when necessary).

The code

```
<?php
/*
 * Created by Joshua Greenert on 1/6/2021
 *
 * This script will check the database for records that have a value of
NULL in the
 * crmLog column for the call logs and missed calls stored.  Once the
files have been
 * checked, they will be checked against the CRM_Resources table to get
the id.  Then,
 * they will be stored in the CRM_Logs table to ensure the events are
visible within the
 * CRM application.
```

```
 *
 * TABLES: crm_resources, crm_logs, tblringcentral_call_logs,
tblringcentral_missed_calls
 */

require_once('../../configuration.php');

// Make the database connection.
$dbhost = "localhost";
$dbuser = $db_username;
$dbpass = $db_password;
$db = $db_name;

$conn = new mysqli($dbhost, $dbuser, $dbpass,$db) or die("Connect
failed: %s\n". $conn -> error);

// Create the queries for the call logs
$query_call_logs = "SELECT * FROM tblringcentral_call_logs where crmLog
is null";
$query_missed_calls = "SELECT * FROM tblringcentral_missed_calls where
crmLog is null";
$query_get_client_phones = "Select * from crm_resources where phone is
not null and phone != ''";
$query_get_id_count = "select max(id) as 'count' from crm_logs";

// Created global variables here.
$resourcesidArray = array();
$resourcesphoneArray = array();
$phone = "";
$idcount;
$direction;
$toname;
$today = date("Y-m-d H:i:s");

/*********************GET CLIENT IDS AND PHONE
NUMBERS************************/
// Get the ids from the crm_resources table by pulling a list of numbers
// The numbers will need to be adjusted to remove any special
characters.
$resultgetclientphones = $conn->query($query_get_client_phones);

if($resultgetclientphones->num_rows > 0){
    while($row = $resultgetclientphones->fetch_assoc()){

        // Update the phone numbers to only be numbers.
        $phone = preg_replace('/[^0-9]/', '', strval($row['phone']));

        if(substr($phone, 0, 1) == "1"){
            $phone = substr($phone, 1, strlen($phone));
        }
```

```
        // Add items to the array.
        array_push($resourcesidArray, $row['id'] );
        array_push($resourcesphoneArray, $phone);
    }
}

/********************GET CALL LOGS INBOUND
/OUTBOUND************************/
// Get the records from the database.
$resultgetcalllogs = $conn->query($query_call_logs);

// While there are rows available, check the outbound/inbound and get
the numbers.
// IF Inbound -> GET fromPhoneNumber, IF Outbound -> GET toPhoneNumber
if($resultgetcalllogs->num_rows > 0){
    while($row = $resultgetcalllogs->fetch_assoc()){

        if($row['direction'] == 'Inbound'){

            $direction = strval($row['direction']);
            $toname = strval($row['toName']);
            $unchangedphone = $row['fromPhoneNumber'];

            // Update the phone numbers to only be numbers.
            $phone = preg_replace('/[^0-9]/', '', strval($row
['fromPhoneNumber']));
            if(substr($phone, 0, 1) == "1"){
                $phone = substr($phone, 1, strlen($phone));
            }

            for($i = 0; $i < count($resourcesphoneArray); $i++){

                if($phone == "" || !is_numeric($phone)){
                    $unchangedphone = "No Data";
                    break;
                }

                if($resourcesphoneArray[$i] == $phone){
                    // Get the new id to set for the primary key
                    $resultidcount = $conn->query($query_get_id_count);
                    if($resultidcount->num_rows > 0){
                        while($rw = $resultidcount->fetch_assoc()){
                            $idcount = $rw['count'] + 1;
                        }
                    }
                    $id = $resourcesidArray[$i];

                    $insertquery = "Insert into crm_logs (id,
resource_id, admin_id, event, date, message) values ($idcount, $id, 14,
```

```
'Called In', '$today', 'Call recorded from $direction to $toname')";
                    mysqli_query( $conn, $insertquery);
                    $updatequery = 'Update tblringcentral_call_logs set
crmLog = "DONE" where fromPhoneNumber = "' .$unchangedphone . '"';
                    mysqli_query( $conn, $updatequery);
                }
            }
        }
        else if($row['direction'] == 'Outbound'){

            $direction = strval($row['direction']);
            $toname = strval($row['fromName']);
            $unchangedphone = $row['toPhoneNumber'];

            // Update the phone numbers to only be numbers.
            $phone = preg_replace('/[^0-9]/', '', strval($row
['toPhoneNumber']));
            if(substr($phone, 0, 1) == "1"){
                $phone = substr($phone, 1, strlen($phone));
            }

            for($i = 0; $i < count($resourcesphoneArray); $i++){

                if($phone == "" || !is_numeric($phone)){
                    $unchangedphone = "No Data";
                    break;
                }

                if($resourcesphoneArray[$i] == $phone){
                    // Get the new id to set for the primary key
                    $resultidcount = $conn->query($query_get_id_count);
                    if($resultidcount->num_rows > 0){
                        while($rw = $resultidcount->fetch_assoc()){
                            $idcount = $rw['count'] + 1;
                        }
                    }
                    $id = $resourcesidArray[$i];

                    $insertquery = "Insert into crm_logs (id,
resource_id, admin_id, event, date, message) values ($idcount, $id, 14,
'Called In', '$today', 'Call recorded from $direction by $toname')";
                    mysqli_query( $conn, $insertquery);

                    $updatequery = 'Update tblringcentral_call_logs set
crmLog = "DONE" where toPhoneNumber = "' .$unchangedphone . '"';
                    mysqli_query( $conn, $updatequery);
                }
            }
        }
```

```
        // If the call has not been updated or recorded in the crm_logs
due to not
        // having a number associated, then update the table to remove
that from future
        // queries.
        if($row['direction'] == 'Inbound'){
            $updatequery = 'Update tblringcentral_call_logs set crmLog
= "DONE" where fromPhoneNumber = "' .$unchangedphone . '"';
            mysqli_query( $conn, $updatequery);
        }
        else{
            $updatequery = 'Update tblringcentral_call_logs set crmLog
= "DONE" where toPhoneNumber = "' .$unchangedphone . '"';
            mysqli_query( $conn, $updatequery);
        }
    }
}

/********************GET MISSED CALLS INBOUND************************/
$resultgetmissedcalls = $conn->query($query_missed_calls);

// Get the missed calls and store them as missed calls in the CRM logs
if($resultgetmissedcalls->num_rows > 0){
    while($row = $resultgetmissedcalls->fetch_assoc()){

        $direction = strval($row['direction']);
        $toname = strval($row['toName']);
        $unchangedphone = strval($row['fromPhoneNumber']);

        // Update the phone numbers to only be numbers.
        $phone = preg_replace('/[^0-9]/', '', strval($row
['fromPhoneNumber']));
        if(substr($phone, 0, 1) == "1"){
            $phone = substr($phone, 1, strlen($phone));
        }

        for($i = 0; $i < count($resourcesphoneArray); $i++){

            if($phone == "" || !is_numeric($phone)){
                $unchangedphone = "No Data";
                break;
            }

            if($resourcesphoneArray[$i] == $phone){
                // Get the new id to set for the primary key
                $resultidcount = $conn->query($query_get_id_count);
                if($resultidcount->num_rows > 0){
                    while($rw = $resultidcount->fetch_assoc()){
                        $idcount = $rw['count'] + 1;
                    }
```

```
                }
                $id = $resourcesidArray[$i];

                $insertquery = "Insert into crm_logs (id, resource_id,
admin_id, event, date, message) values ($idcount, $id, 14, 'Missed
Call', '$today', 'Call missed from $unchangedphone to $toname')";
                mysqli_query( $conn, $insertquery);

                $updatequery = 'Update tblringcentral_missed_calls set
crmLog = "DONE" where fromPhoneNumber = "' . $unchangedphone .'"';
                mysqli_query( $conn, $updatequery);

            }
        }

        // If the call has not been updated or recorded in the crm_logs
due to not
        // having a number associated, then update the table to remove
that from future
        // queries.
        $updatequery = 'Update tblringcentral_missed_calls set crmLog =
"DONE" where fromPhoneNumber = "' . $unchangedphone .'"';
        mysqli_query( $conn, $updatequery);
    }
}
mysqli_close($conn);
?>
```

Capitalcityshoes - Add Customization To Item Description (Ariba)

This page will host the function used for Capitalcityshoes to send the customization info over through Ariba. The pieces that are passed over are initially placed in hidden input items that are posted when the user selects to post their punchout order. This information is being placed in the instance that future adjustments need to be made to punchout processes.

This page can be found in the files under: /Users/JoshG/Documents/Github/Shop.CapitalCityShoes.com/app/design/frontend/default/theme542/template/checkout/onepage/review/info.phtml

The Code

```
<?php
/**
 * Magento
 *
 * NOTICE OF LICENSE
 *
 * This source file is subject to the Academic Free License (AFL 3.0)
 * that is bundled with this package in the file LICENSE_AFL.txt.
 * It is also available through the world-wide-web at this URL:
 * http://opensource.org/licenses/afl-3.0.php
 * If you did not receive a copy of the license and are unable to
 * obtain it through the world-wide-web, please send an email
```

```
 * to license@magento.com so we can send you a copy immediately.
 *
 * DISCLAIMER
 *
 * Do not edit or add to this file if you wish to upgrade Magento to
newer
 * versions in the future. If you wish to customize Magento for your
 * needs please refer to http://www.magento.com for more information.
 *
 * @category    design
 * @package     base_default
 * @copyright   Copyright (c) 2006-2015 X.commerce, Inc. (http://www.
magento.com)
 * @license     http://opensource.org/licenses/afl-3.0.php  Academic
Free License (AFL 3.0)
 */
include('var/variables/variables.php');

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    echo "Connection Failed";
}
?>
<?php echo $this->getChildHtml('items_before'); ?>

<div id="checkout-review-table-wrapper">
 <form id="punchout-basket-form" action="<?php echo $_COOKIE
["callbackURL"];?>" method="post">
    <table class="data-table" id="checkout-review-table">
        <?php if ($this->helper('tax')->displayCartBothPrices()):
$colspan = $rowspan = 2; else: $colspan = $rowspan = 1; endif; ?>
        <col />
        <col width="1" />
        <col width="1" />
        <col width="1" />
        <?php if ($this->helper('tax')->displayCartBothPrices()): ?>
        <col width="1" />
        <col width="1" />
        <?php endif; ?>
        <thead>
            <tr>
                     <th rowspan="<?php echo $rowspan ?>"><?php echo
$this->__('Product Name') ?></th>
                <th colspan="<?php echo $colspan ?>" class="a-center"><?
php echo $this->__('Price') ?></th>
                <th rowspan="<?php echo $rowspan ?>" class="a-center"><?
php echo $this->__('Qty') ?></th>
```

```
                <th colspan="<?php echo $colspan ?>" class="a-center"><?
php echo $this->__('Subtotal') ?></th>
            </tr>
            <?php if ($this->helper('tax')->displayCartBothPrices()): ?>
                <tr>
                    <th class="a-right"><?php echo $this->helper('tax')-
>getIncExcTaxLabel(false) ?></th>
                    <th><?php echo $this->helper('tax')-
>getIncExcTaxLabel(true) ?></th>
                    <th class="a-right"><?php echo $this->helper('tax')-
>getIncExcTaxLabel(false) ?></th>
                    <th><?php echo $this->helper('tax')-
>getIncExcTaxLabel(true) ?></th>
                </tr>
            <?php endif; ?>
        </thead>
        <?php echo $this->getChildHtml('totals'); ?>
        <tbody>

    <?php //set counter variable // updated Description >> Get Name
        $counter = 1;
        $duplicateSkuArray = array();
        $duplicateFound = "";

        foreach($this->getItems() as $_item):
        {  $product = Mage::getModel('catalog/product')->load($_item-
>getProduct()->getId());
                $manu = $product->getAttributeText('manufacturer');
                $comcode = $product->getData('commodity_code');
                $itemid = $product->getData('item_no');
        $color = $product->getAttributeText('color');
        $prodtype =  $product->getTypeId();

        try{
            if($prodtype === "simple"){
                $simplename = strval($_item->getName());
            }
            else{
                $simplename = strval($_item->getOptionByCode
('simple_product')->getProduct()->getName());
            }
        }catch(Exception $e){
            Mage::Log('Caught exception: ',  $e->getMessage(), "\n");
        }

        // Search database to get options not explicitly listed; get
quote id and sku first for each product.
        $session = Mage::getSingleton('checkout/session');
        $quote_id = $session->getQuoteId();
        $sku = $_item->getSku();
```

```
        // Check for later as needed.
        $duplicateFound = "false";

        // Check if the item is a simple or configurable product for
the query.
        if($prodtype === "simple"){
            $getItemIdQuery = "SELECT item_id FROM
sales_flat_quote_item where quote_id = $quote_id and sku = $sku limit
1";
        }
        else{
            $getItemIdQuery = "SELECT item_id, sku FROM
sales_flat_quote_item where quote_id = $quote_id and sku = $sku and
parent_item_id is not null";
        }

        $resultGetItemId = $conn->query($getItemIdQuery);

        if($resultGetItemId->num_rows == 1){
            while($row = $resultGetItemId->fetch_assoc()){
                $productItemId = $row['item_id'];

                // Get the options for the products
                $getProductOptionsQuery = "SELECT
sales_flat_quote_item_option.code, sales_flat_quote_item_option.value
FROM sales_flat_quote_item_option where item_id = $productItemId
                    and code like '%option%' and code != 'option_ids'";

                $resultGetItemOptions = $conn->query
($getProductOptionsQuery);

                if($resultGetItemOptions->num_rows > 0){
                    $simplename .= " CUSTOMIZATION:";

                    while($row2 = $resultGetItemOptions->fetch_assoc()){
                        $productCode = $row2['code'];
                        $productValue = $row2['value'];

                        if($productCode === "option_10068"){
                            $simplename .= " NAME STRIP: " .
$productValue;
                        }
                        else{
                            // For each value, search the table to find
it's associated value.  21417 = FDACS Main Logo
                            $getOptionTitle = "SELECT title FROM
catalog_product_option_type_title  where option_type_id =
$productValue";
                            $resultGetOptionTitle = $conn->query
```

```
($getOptionTitle);

                            if($resultGetOptionTitle->num_rows > 0){
                                while($row3 = $resultGetOptionTitle-
>fetch_assoc()){
                                    $title = strval($row3['title']);
                                    $simplename .= " " . $title;
                                }
                            }
                        }
                    }
                }
            }
        }
        else if($resultGetItemId->num_rows > 1){
            if(empty($duplicateSkuArray)){
                $duplicateCount = 0;

                while($row = $resultGetItemId->fetch_assoc()){
                    $productItemId = $row['item_id'];
                    $productItemSku = $row['sku'];

                    if($duplicateCount > 0){
                        array_push($duplicateSkuArray, [$productItemId,
$productItemSku]);
                    }
                    else{
                        // Get the options for the products
                        $getProductOptionsQuery = "SELECT
sales_flat_quote_item_option.code, sales_flat_quote_item_option.value
FROM sales_flat_quote_item_option where item_id = $productItemId
                        and code like '%option%' and code !=
'option_ids'";

                        $resultGetItemOptions = $conn->query
($getProductOptionsQuery);

                        if($resultGetItemOptions->num_rows > 0){
                            $simplename .= " CUSTOMIZATION:";

                            while($row2 = $resultGetItemOptions-
>fetch_assoc()){
                                $productCode = $row2['code'];
                                $productValue = $row2['value'];

                                if($productCode === "option_10068"){
                                    $simplename .= " NAME STRIP: " .
$productValue;
                                }
                                else{
```

```
                                    // For each value, search the table
to find it's associated value.  21417 = FDACS Main Logo
                                    $getOptionTitle = "SELECT title
FROM catalog_product_option_type_title  where option_type_id =
$productValue";
                                    $resultGetOptionTitle = $conn->query
($getOptionTitle);

                                    if($resultGetOptionTitle->num_rows
> 0){
                                        while($row3 =
$resultGetOptionTitle->fetch_assoc()){
                                            $title = strval($row3
['title']);
                                            $simplename .= " " . $title;
                                        }
                                    }
                                }
                            }
                        }

                        // Increment the duplicate counter.
                        $duplicateCount++;
                    }


                }
            }
            else{

                // Look through the array to find the matching sku for
the product and set the item_id variable.
                for($i = 0; $i < count($duplicateSkuArray); $i++){
                    if($sku === $duplicateSkuArray[$i][1]){
                        $productItemId = $duplicateSkuArray[$i][0];
                        unset($duplicateSkuArray[$i]);

                        // recatigorize the array so that future
iterations will be able to acknowledge the difference.
                        $duplicateSkuArray = array_values
($duplicateSkuArray);
                        $duplicateFound = "true";
                        break;
                    }
                }

                // If the duplicate was found, then apply the next item
in the array; otherwise, go through the next
                // set of items and determine what needs to be added to
the array for future iterations.
```

```
                if($duplicateFound === "true"){
                    // Get the options for the products
                    $getProductOptionsQuery = "SELECT
sales_flat_quote_item_option.code, sales_flat_quote_item_option.value
FROM sales_flat_quote_item_option where item_id = $productItemId
                    and code like '%option%' and code != 'option_ids'";

                    $resultGetItemOptions = $conn->query
($getProductOptionsQuery);

                    if($resultGetItemOptions->num_rows > 0){
                        $simplename .= " CUSTOMIZATION:";

                        while($row2 = $resultGetItemOptions-
>fetch_assoc()){
                            $productCode = $row2['code'];
                            $productValue = $row2['value'];

                            if($productCode === "option_10068"){
                                $simplename .= " NAME STRIP: " .
$productValue;
                            }
                            else{
                                // For each value, search the table to
find it's associated value.  21417 = FDACS Main Logo
                                $getOptionTitle = "SELECT title FROM
catalog_product_option_type_title  where option_type_id =
$productValue";
                                $resultGetOptionTitle = $conn->query
($getOptionTitle);

                                if($resultGetOptionTitle->num_rows > 0){
                                    while($row3 = $resultGetOptionTitle-
>fetch_assoc()){
                                        $title = strval($row3['title']);
                                        $simplename .= " " . $title;
                                    }
                                }
                            }
                        }
                    }
                }
                else{
                    $duplicateCount = 0;

                    while($row = $resultGetItemId->fetch_assoc()){
                        $productItemId = $row['item_id'];
                        $productItemSku = $row['sku'];

                        if($duplicateCount > 0){
```

```
                              array_push($duplicateSkuArray,
[$productItemId, $productItemSku]);
                        }
                        else{
                            // Get the options for the products
                            $getProductOptionsQuery = "SELECT
sales_flat_quote_item_option.code, sales_flat_quote_item_option.value
FROM sales_flat_quote_item_option where item_id = $productItemId
                            and code like '%option%' and code !=
'option_ids'";

                            $resultGetItemOptions = $conn->query
($getProductOptionsQuery);

                            if($resultGetItemOptions->num_rows > 0){
                                $simplename .= " CUSTOMIZATION:";

                                while($row2 = $resultGetItemOptions-
>fetch_assoc()){
                                    $productCode = $row2['code'];
                                    $productValue = $row2['value'];

                                    if($productCode === "option_10068"){
                                        $simplename .= " NAME STRIP: "
. $productValue;
                                    }
                                    else{
                                        // For each value, search the
table to find it's associated value.  21417 = FDACS Main Logo
                                        $getOptionTitle = "SELECT title
FROM catalog_product_option_type_title  where option_type_id =
$productValue";
                                        $resultGetOptionTitle = $conn-
>query($getOptionTitle);

                                        if($resultGetOptionTitle-
>num_rows > 0){
                                            while($row3 =
$resultGetOptionTitle->fetch_assoc()){
                                                $title = strval($row3
['title']);
                                                $simplename .= " " .
$title;
                                            }
                                        }
                                    }
                                }
                            }

                            // Increment the duplicate counter.
```

```
                            $duplicateCount++;
                        }
                    }
                }
            }
        }

        // Gets the config product color - static only
        /* Updates from 7/29 - change to get simple product name
         * Updated on 7/23/2021 by Joshua Greenert
        */
    }

        ?>
       <?php echo $this->getItemHtml($_item) ?>

                <input type="hidden" name="items[<?php echo $counter?>].
itemCode" value="<?php echo $_item->getSku() ?>">
                <input type="hidden" name="items[<?php echo $counter?>].
itemQuantity" value="<?php echo $_item->getQty() ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
unitPrice" value="<?php echo $_item->getCalculationPrice() ?>"/>
                <input type="hidden" id="item_description<?php echo
$counter?>" name="items[<?php echo $counter?>].itemDescription" value="
<?php echo $simplename; ?> - <?php echo $product->getSku() ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
descriptionLanguage" value="en"/>
                <input type="hidden" name="items[<?php echo $counter?>].
currencyName" value="USD"/>
                <input type="hidden" name="items[<?php echo $counter?>].
unitOfMeasure" value="EA"/>
                <input type="hidden" name="items[<?php echo $counter?>].
unspsc" value="<?php echo $comcode; ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
manufacturerName" value="<?php echo $manu; ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
manufacturerPartID" value="<?php echo $itemid; ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
supplierBasketLineNumber" value="<?php echo $counter?>"/>

                <?php
        $counter++; //increment counter by 1 on every pass
        ?>
        <?php endforeach ?>

    </table>
    <input name="punchout" type="submit" title="Post Punchout" value="
Post Punchout" label="Post to Punchout" onclick="review.save();" id="
punchout">
        </form>
```

```
</div>
<?php echo $this->getChildHtml('items_after'); ?>
<script type="text/javascript">
//<![CDATA[
    decorateTable('checkout-review-table');
    truncateOptions();
//]]>
</script>
<div id="checkout-review-submit">
    <?php echo $this->getChildHtml('agreements') ?>
    <div class="buttons-set" id="review-buttons-container">
        <p class="f-left"><?php echo $this->__('Need to make a
change?') ?> <a href="<?php echo $this->getUrl('checkout/cart') ?>"><?
php echo $this->__('Edit Your Cart') ?></a></p>
        <?php echo $this->getChildHtml('button') ?>
        <span class="please-wait" id="review-please-wait" style="
display:none;">
            <img src="<?php echo $this->getSkinUrl('images/opc-ajax-
loader.gif') ?>" alt="<?php echo Mage::helper('core')->quoteEscape
($this->__('Submitting order information...')) ?>" title="<?php echo
Mage::helper('core')->quoteEscape($this->__('Submitting order
information...')) ?>" class="v-middle" /> <?php echo $this->__
('Submitting order information...') ?>
        </span>
    </div>
    <script type="text/javascript">
    //<![CDATA[
        review = new Review('<?php echo $this->getUrl('checkout/onepage
/saveOrder', array('form_key' => Mage::getSingleton('core/session')-
>getFormKey())) ?>', '<?php echo $this->getUrl('checkout/onepage
/success') ?>', $('checkout-agreements'));
    //]]>
    </script>
</div>
```

Capitalcityshoes - Cookies

This page does not conform to the normal approach of supplying a script and having solutions. However, the pieces that are left here may be necessary in the future while working with cookies for alternative solutions. Below I will place the process for creating a cookie with JavaScript and PHP so that they are documented for future use cases.

The Code: JavaScript

```
<script>
                var allCookies = document.cookie;
                var formkeyValue = '';
                var callbackURLValue = '';
                var flag = false;

                document.cookie = 'PunchoutFormkey' +"=; Max-Age=-
```

```
99999999;";

                // Get the cookie desired.
                var cookieArray = allCookies.split(';');
                for(var i = 0; i < cookieArray.length; i++){
                        cookiename = cookieArray[i].split('=')[0];
                        if(cookiename.includes("PunchoutFormkey")){
                                formkeyValue = cookieArray[i].split('=')
[1];
                        }

                        if(cookiename.includes("callbackURL")){
                                callbackURLValue = cookieArray[i].split
('=')[1];
                                flag = true;
                        }
                }

                // Get the value from the formkey
                if(flag){
                        var strlength = callbackURLValue.length - 12;
                        var strindex = callbackURLValue.indexOf("tid")
+ 6
                        var newcallbackURLValue = callbackURLValue.
substring(strindex, strlength);

                        //Mage.Cookies.set('PunchoutFormkey', "", -1);

                        var cookieName = 'PunchoutFormkey';
                        var cookieValue = newcallbackURLValue;
                        var myDate = new Date();
                        myDate.setMonth(myDate.getMonth() + 12);
                        document.cookie = cookieName +"=" + cookieValue
+ ";expires=" + myDate
                                                            + ";domain=.
capitalcityshoes.com;path=/";

                }

                </script>
```

The Code: PHP

```
// Get the cookie formid and send it over the new URL.
        if(isset($_COOKIE['callbackURL'])){
                $site = "https://stateoffloridagear.capitalcityshoes.
com";
```

```
                $callbackCookie = $_COOKIE['callbackURL'];

                // Get the formid
                $strposition = strpos($callbackCookie, 'tid');
                $strlength = strlen($callbackCookie) - $strposition -
12;
                $cookieFormID = substr($callbackCookie, $strposition +
4, $strlength-4);

                // Check if the punchoutformkey cookie is set already,
reset it if it is.
                if(isset($_COOKIE['PunchoutFormkey']) && isset($_COOKIE
['callbackURL'])){
                        if($_COOKIE['PunchoutFormkey'] !==
$cookieFormID){
                                unset($_COOKIE['PunchoutFormkey']);
                                setcookie('PunchoutFormkey',
$cookieFormID, time() + 3600, "/", '.capitalcityshoes.com');
                                Header('Location: '.$_SERVER
['PHP_SELF']);
                                Exit();
                        }
                }
                else{
                        setcookie('PunchoutFormkey', $cookieFormID,
time() + 3600, "/", '.capitalcityshoes.com');
                }

        }
```

Capitalcityshoes - DMS Users Update Store ID

This page will host the script to update DMS users' views so that they receive the email with their voucher when activated. Currently, this script is located in public_html/DMSUserStoreIDUpdate.php and has a cron job set for once every hour.

The Code

```
<?php
// This script will be set up with a cron job to ensure that, when DMS
users are created
// on the backend, their store_id associated is updated for them to
receive the voucher email.
include('var/variables/variables.php');

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

```
// Create the statement.
$sql = "Select * from customer_entity where website_id = 6 and email
like '%dms%' and store_id != 43";

// If there are records returned from the query, update them to the
proper store_id.
$resultGetRecords = $conn->query($sql);
if($resultGetRecords->num_rows > 0){
    $updateQuery = "update customer_entity set store_id = 43 where
website_id = 6 and email like '%dms%'";
    mysqli_query( $conn, $updateQuery);
}

?>
```

Capitalcityshoes - Import Users View & Credit Update

This script can be used to update the imported users for Capitalcityshoes. Once users have been imported, this script can update the clothing allowances for users based on email address, then update the views for the users (in this case Moore Bass) to be 79 and 66 ONLY; these views are b2b default store and b2b Moore Bass store view. These can be used separately for other instances as well. The final piece at the bottom that activates the users afterwards has been commented out since it doesn't fire off the email as anticipated.

EDIT: Findings discovered that Itoris tables were not auto-updating with incorrect customer_id, but were not updated at all. The default value for a user not found is 0, which makes the user have full views on every store. Updated and split code into two pieces to ensure that they are ran separately for best practices, and to ensure that the updates for the views are not impacted by the updates for the credits (i.e. an issue within the code for the credits won't hinder the process of the views).

The Code - Part 1 Credits

```
<?php
/**
 * Magento
 *
 * NOTICE OF LICENSE
 *
 * This source file is subject to the Academic Free License (AFL 3.0)
 * that is bundled with this package in the file LICENSE_AFL.txt.
 * It is also available through the world-wide-web at this URL:
 * http://opensource.org/licenses/afl-3.0.php
 * If you did not receive a copy of the license and are unable to
 * obtain it through the world-wide-web, please send an email
 * to license@magento.com so we can send you a copy immediately.
 *
 * DISCLAIMER
 *
 * Do not edit or add to this file if you wish to upgrade Magento to
newer
 * versions in the future. If you wish to customize Magento for your
 * needs please refer to http://www.magento.com for more information.
 *
 * @category    design
 * @package     base_default
```

```
 * @copyright   Copyright (c) 2006-2015 X.commerce, Inc. (http://www.
magento.com)
 * @license     http://opensource.org/licenses/afl-3.0.php  Academic
Free License (AFL 3.0)
 */
include('var/variables/variables.php');

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
$connectionCheck = "";
// Check connection
if ($conn->connect_error) {
    echo "Connection Failed";
    //die("Connection failed: " . $conn->connect_error);
}
    //echo "Connected successfully";
?>
<?php echo $this->getChildHtml('items_before'); ?>

<div id="checkout-review-table-wrapper">
 <form id="punchout-basket-form" action="<?php echo $_COOKIE
["callbackURL"];?>" method="post">
    <table class="data-table" id="checkout-review-table">
        <?php if ($this->helper('tax')->displayCartBothPrices()):
$colspan = $rowspan = 2; else: $colspan = $rowspan = 1; endif; ?>
        <col />
        <col width="1" />
        <col width="1" />
        <col width="1" />
        <?php if ($this->helper('tax')->displayCartBothPrices()): ?>
        <col width="1" />
        <col width="1" />
        <?php endif; ?>
        <thead>
            <tr>
                     <th rowspan="<?php echo $rowspan ?>"><?php echo
$this->__('Product Name') ?></th>
                <th colspan="<?php echo $colspan ?>" class="a-center"><?
php echo $this->__('Price') ?></th>
                <th rowspan="<?php echo $rowspan ?>" class="a-center"><?
php echo $this->__('Qty') ?></th>
                <th colspan="<?php echo $colspan ?>" class="a-center"><?
php echo $this->__('Subtotal') ?></th>
            </tr>
            <?php if ($this->helper('tax')->displayCartBothPrices()): ?>
                <tr>
                    <th class="a-right"><?php echo $this->helper('tax')-
>getIncExcTaxLabel(false) ?></th>
                    <th><?php echo $this->helper('tax')-
>getIncExcTaxLabel(true) ?></th>
```

```
                    <th class="a-right"><?php echo $this->helper('tax')-
>getIncExcTaxLabel(false) ?></th>
                    <th><?php echo $this->helper('tax')-
>getIncExcTaxLabel(true) ?></th>
                </tr>
            <?php endif; ?>
        </thead>
        <?php echo $this->getChildHtml('totals'); ?>
        <tbody>

    <?php //set counter variable // updated Description >> Get Name
        $counter = 1;
        $index = 0;
        $productNameArrayList = array();

        // Get count of all unique items in cart.
        foreach($this->getItems() as $_item){
            $index= $index +1;
        }
    ?>

        <?php

        $cartItems = Mage::getSingleton('checkout/session')->getQuote()-
>getAllItems();

        foreach($this->getItems() as $_item):
        {  $product = Mage::getModel('catalog/product')->load($_item-
>getProduct()->getId());
                $manu = $product->getAttributeText('manufacturer');
                $comcode = $product->getData('commodity_code');
                $itemid = $product->getData('item_no');
        $color = $product->getAttributeText('color');
        $prodtype =  $product->getTypeId();

        try{
            if($prodtype === "simple"){
                $simplename = $_item->getName();
            }
            else{
                $simplename = $_item->getOptionByCode('simple_product')-
>getProduct()->getName();
            }
        }catch(Exception $e){
            Mage::Log('Caught exception: ',  $e->getMessage(), "\n");
        }

        // Get the product options for the product.
        $optionsstring = "";
        foreach ($product->getOptions() as $_option) {
```

```
            $values = $_option->getValues();
            foreach ($values as $v) {
                $optionsstring .= $v->getTitle() . " ";
            }
        }

        if($optionsstring !== ""){
            $simplename .= " - OPTIONS - " . $optionsstring;
        }

        // Search database to get options not explicitly listed; get
quote id and sku first for each product.
        $session = Mage::getSingleton('checkout/session');
        $quote_id = $session->getQuoteId();
        $sku = $_item->getSku();

        $getItemIdQuery = "SELECT item_id FROM sales_flat_quote_item
where quote_id = $quote_id and sku = $sku limit 1";
        $resultGetItemId = $conn->query($getItemIdQuery);

        if($resultGetItemId->num_rows > 0){
            while($row = $resultGetItemId->fetch_assoc()){
                $productItemId = $row['item_id'];

                // Get the options for the products
                $getProductOptionsQuery = "SELECT
sales_flat_quote_item_option.code, sales_flat_quote_item_option.value
FROM sales_flat_quote_item_option where item_id = $productItemId
                    and code != 'info_buyRequest' and code !=
'attributes' and code != 'simple_product'";
                $resultGetItemOptions = $conn->query
($getProductOptionsQuery);

                if($resultGetItemOptions->num_rows > 0){
                    while($row2 = $resultGetItemOptions->fetch_assoc()){
                        $productCode = $row2['code'];
                        $productValue = $row2['value'];

                        if($productCode === "option_10067"){
                            if($productValue == 22579){
                                $simplename .= " BADGE: Gold Badge";
                            }
                            else if($productValue == 22578){
                                $simplename .= " BADGE: Silver Badge";
                            }
                        }
                        else if($productCode === "option_10068"){
                            // Replace the text to avoid redundancies
                            $simplename = str_replace("Silver Badge
Gold Badge", "", $simpleName);
```

```
                            $simplename .= " NAME STRIP: " .
$productValue;
                        }
                        //Mage::Log($countIndex . " " . $productCode);
                    }
                }
            }
        }

        // Gets the config product color - static only
        /* Updates from 7/29 - change to get simple product name
         * Updated on 7/23/2021 by Joshua Greenert
        */
    }

        ?>
       <?php echo $this->getItemHtml($_item) ?>

                <input type="hidden" name="items[<?php echo $counter?>].
itemCode" value="<?php echo $_item->getSku() ?>">
                <input type="hidden" name="items[<?php echo $counter?>].
itemQuantity" value="<?php echo $_item->getQty() ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
unitPrice" value="<?php echo $_item->getCalculationPrice() ?>"/>
                <input type="hidden" id="item_description<?php echo
$counter?>" name="items[<?php echo $counter?>].itemDescription" value="
<?php echo $simplename; ?> - <?php echo $product->getSku() ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
descriptionLanguage" value="en"/>
                <input type="hidden" name="items[<?php echo $counter?>].
currencyName" value="USD"/>
                <input type="hidden" name="items[<?php echo $counter?>].
unitOfMeasure" value="<?php // echo $_item->getUom() ?>EA"/>
                <input type="hidden" name="items[<?php echo $counter?>].
unspsc" value="<?php echo $comcode; ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
manufacturerName" value="<?php echo $manu; ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
manufacturerPartID" value="<?php echo $itemid; ?>"/>
                <input type="hidden" name="items[<?php echo $counter?>].
supplierBasketLineNumber" value="<?php echo $counter?>"/>

                <?php
        $counter++; //increment counter by 1 on every pass
        ?>

        <?php endforeach ?>


    </table>
```

```
    <input name="punchout" type="submit" title="Post Punchout" value="
Post Punchout" label="Post to Punchout" onclick="review.save();" id="
punchout">
        </form>
</div>
<?php echo $this->getChildHtml('items_after'); ?>
<script type="text/javascript">
//<![CDATA[
    decorateTable('checkout-review-table');
    truncateOptions();
//]]>
</script>
<div id="checkout-review-submit">
    <?php echo $this->getChildHtml('agreements') ?>
    <div class="buttons-set" id="review-buttons-container">
        <p class="f-left"><?php echo $this->__('Need to make a
change?') ?> <a href="<?php echo $this->getUrl('checkout/cart') ?>"><?
php echo $this->__('Edit Your Cart') ?></a></p>
        <?php echo $this->getChildHtml('button') ?>
        <span class="please-wait" id="review-please-wait" style="
display:none;">
            <img src="<?php echo $this->getSkinUrl('images/opc-ajax-
loader.gif') ?>" alt="<?php echo Mage::helper('core')->quoteEscape
($this->__('Submitting order information...')) ?>" title="<?php echo
Mage::helper('core')->quoteEscape($this->__('Submitting order
information...')) ?>" class="v-middle" /> <?php echo $this->__
('Submitting order information...') ?>
        </span>
    </div>
    <script type="text/javascript">
    //<![CDATA[
        review = new Review('<?php echo $this->getUrl('checkout/onepage
/saveOrder', array('form_key' => Mage::getSingleton('core/session')-
>getFormKey())) ?>', '<?php echo $this->getUrl('checkout/onepage
/success') ?>', $('checkout-agreements'));
    //]]>
    </script>
</div>
```

Capitalcityshoes - Insert Inventory Missing Rows

This script when executed performs a query to capture all products entered into the table that are not present in the table that shows inventory configurations; typically this happens as a result of an import of products not completing the table insert statements. Afterwards, this record is inserted into the table with the product that is missing the configuration for each and every product.

The Code

```
<?php
/*
 * Created by Joshua Greenert on 2/8/2021
```

```
 *
 * This script will create a database connection and insert missing
rows based on
 * the specified range provided.  This range will create an array of
product ids
 * that will be looped through to update the same information for each
product to
 * ensure that the imported products, that didn't carry over any table
data for inventory,
 * have a record and display properly on the site.
 *
 * Tables: cataloginventory_stock_item, catalog_product_entity
 */
include('var/variables/variables.php');

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Set the query and use it to navigate and insert records for each
product not listed.
$sqlQuery = "SELECT entity_id FROM catalog_product_entity where
entity_id NOT IN (Select product_id from cataloginventory_stock_item)";

$resultGetRecords = $conn->query($sqlQuery);
if($resultGetRecords->num_rows > 0){

    // Loop through each item and insert it into the secondary table.
    while($row = $resultGetRecords->fetch_assoc()){

        $entity = $row['entity_id'];

        $insertQuery = "INSERT into cataloginventory_stock_item
(product_id, stock_id, qty, min_qty, use_config_min_qty,
is_qty_decimal, backorders, use_config_backorders, min_sale_qty,
use_config_min_sale_qty, max_sale_qty,
            use_config_max_sale_qty, is_in_stock, low_stock_date,
notify_stock_qty, use_config_notify_stock_qty, manage_stock,
use_config_manage_stock, stock_status_changed_auto,
use_config_qty_increments, qty_increments,
            use_config_enable_qty_inc, enable_qty_increments,
is_decimal_divided) values ($entity, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1,
0, NULL, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0)";
        mysqli_query( $conn, $insertQuery);
    }
}

echo "Completed";
```

```
?>
```

Capitalcityshoes - Send Activation Email Automation

This page will host the script for sending emails to users upon account activation. Basically, it works by using the amasty_audit_log table to see if a user has performed an activation for a user. It then takes the element id (entity_id) to search for the user that was updated to see if they need to have an email sent out based on a 5 minute interval (when the cron is ran). The itoris table is reviewed afterwards to see what store should be aligned with the user for the proper email template from core_email_template.

IMPORTANT: Future stores, when added, should be placed in the switch statement to align the user with the appropriate email template. This process still requires the user to be sent a separate email for their password since the password isn't able to be visible in the code.

The Code

```
<?php
/*
 * Created by Joshua Greenert on 3/15/2021
 *
 * This script will ensure that activation emails send out when the user in the magento database
 * is activated.  It begins by checking the record changes to the amasty_audit_log table for a time
 * that is within 5 minutes (cron will run every 5) and gets every row where the category is equal to
 * "dep2820/customer" and grabs the element_id (prim key for user) field.  Customer_entity_text table
 * is used to check if the entity_id (element_id)value is set to 2.  IF yes, Element_id key is used to
 * get the email field from customer_entity, get the store ids from itoris table, look through the ids
 * to determine which email to send (keys matched to email template).
 *
 * Use template id to get template_text, template_subject, set to email object with mage mail and send to user.
 *
 * Tables: amasty_audit_log, customer_entity, customer_entity_text, itoris_storelogincontrol_customer, core_email_template
 */

include('var/variables/variables.php');
require_once('app/Mage.php');
Mage::app();

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Set up the mail function to call later in the script.
```

```
function sendMailAction($template_id, $template_text,
$template_subject, $email, $customer_name, $customer_id)
{
    //$customer = Mage::getModel('customer/customer')->load
($customer_id);
    //$custpassword = $customer.password;
    /* load template by id
    $email_template  = Mage::getModel('core/email_template')-
>loadDefault($template_id);

    $email_template_variables = array('customer_name' =>
$customer_name, 'password' =>$customer.password);
    $sender_email = 'orders@capitalcityshoes.com';
    $sender_name =  'The Shoe Box Sandbox';
    $email_template->setSenderName($sender_name);
    $email_template->setSenderEmail($sender_email);
    $email_template->send($email,
$customer_name,$email_template_variables);
    */

    // Parse the template passed through to remove the pieces not
needed and correct values for variables.
    $html = str_replace('{{inlinecss file="email-inline.css"}}', '',
$template_text);
    $html = str_replace('{{template config_path="design/email
/header"}}', '', $html);
    $html = str_replace('{{depend store_phone}}', '', $html);
    $html = str_replace('{{/depend}}', '', $html);
    $html = str_replace("{{var customer.custom('vouchercode')}", '',
$html);

    // Update the data points to ensure accurate information provided
for the user.
    $html = str_replace('{{template config_path="design/email
/footer"}}', 'Thank you, The Shoe Box!', $html);
    $html = str_replace('{{var store_email}}', 'orders@capitalcityshoes.
com', $html);
    $html = str_replace('{{var store_phone}}', '(850) 877-1784', $html);
    $html = str_replace('{{var store.getFrontendName()}}', 'The Shoe
Box', $html);
    $html = str_replace('{{htmlescape var=$customer.password}}',
"<b>Your password will be sent in a separate email</b>", $html);
    $html = str_replace('{{var customer.email}}', $email, $html);

    // Parse the subject to ensure that the user name is added.
    $subject = str_replace( "{{var customer.name}}", $customer_name,
$template_subject);

    $mail = Mage::getModel('core/email');
    $mail->setToName($customer_name);
```

```
    $mail->setToEmail($email);
    $mail->setBody($html);
    $mail->setSubject($subject);
    $mail->setFromEmail('orders@capitalcityshoes.com');
    $mail->setFromName('Capitalcityshoes');
    $mail->setType('html');
    $mail->setBodyHTML($html);
    try {
        $mail->send();
        Mage::getSingleton('core/session')->addSuccess('Your request
has been sent');
    }
    catch (Exception $e) {
        Mage::getSingleton('core/session')->addError('Unable to send.');
    }
}

// Create an array to hold values to reduce duplicates
$array = array();
$sentAlreadyArray = array();

// Get the records that show within 5 minutes from the current time.
$getRecentQuery = "SELECT distinct(element_id), date_time FROM
amasty_audit_log where SUBTIME(date_time,'04:00:00') > (NOW() -
INTERVAL 5 MINUTE) && category = 'dep2820/customer' order by date_time
desc";
$resultGetRecent = $conn->query($getRecentQuery);

if($resultGetRecent->num_rows > 0){
    while($row = $resultGetRecent->fetch_assoc()){
        $customer_id = '';
        $customer_name = '';

        // Correct the time to ensure no one is added outside of the 5
minute piece.
        $time = $row['date_time'];

        $datetime1 = new DateTime(date('Y-m-d H:i:s'));
        $datetime2 = new DateTime($time);
        $interval = $datetime1->diff($datetime2);
        $seconds = (intval($interval->format('%i')) * 60) + intval
($interval->format('%s'));

        if($seconds < 300){
            if(in_array($row['element_id'], $array)){
                continue;
            }
            else{
                array_push($array, $row['element_id']);
                $customer_id = $row['element_id'];
```

```
            }
        }
        else{
            exit("No users meet the desired timeline.");
        }

        // Set the customer name from the appropriate table data.
        $getCustomerName = "Select value from customer_entity_varchar
where entity_id = $customer_id && attribute_id = 5 or entity_id =
$customer_id && attribute_id = 7";
        $resultGetCustomerName = $conn->query($getCustomerName);

        if($resultGetCustomerName->num_rows > 0){
            while($row6 = $resultGetCustomerName->fetch_assoc()){
                $customer_name .= $row6['value'] . " ";
            }
        }

        // Trim the customer name for the additional space added on the
last name.
        $customer_name = substr($customer_name, 0, strlen
($customer_name) - 1);

        // Ensure that the records meet the activation requirements
with check or end
        $getActivatedConfirmationQuery = "Select * from
customer_entity_text where entity_id = $customer_id && value = 2 &&
attribute_id = 171";
        $resultGetActivated = $conn->query
($getActivatedConfirmationQuery);

        if($resultGetActivated->num_rows > 0){
            while($row2 = $resultGetActivated->fetch_assoc()){

                // Get the email associated with the entity_id
                $getEmail = "Select email from customer_entity where
entity_id = $customer_id";
                $resultGetEmail = $conn->query($getEmail);

                if($resultGetEmail->num_rows > 0){
                    while($row3 = $resultGetEmail->fetch_assoc()){
                        $email = $row3['email'];

                        // Get the stores associated with the user.
                        $getStores = "Select store_id from
itoris_storelogincontrol_customer where customer_id =$customer_id &&
store_id != 66 && store_id != 75 limit 1";
                        $resultGetStores = $conn->query($getStores);

                        if($resultGetStores->num_rows > 0){
```

```
                              while($row4 = $resultGetStores-
>fetch_assoc()){
                                  $store_id = $row4['store_id'];

                                  // Set the template id for the email.
                                  $template_id = 0;

                                  switch($store_id){
                                      case 79:
                                          $template_id = 26;
                                          break;
                                      case 50:
                                      case 54:
                                      case 61:
                                          $template_id = 20;
                                          break;
                                      case 27:
                                      case 43:
                                          $template_id = 15;
                                          break;
                                      case 40:
                                      case 45:
                                      case 46:
                                      case 48:
                                      case 49:
                                      case 50:
                                      case 53:
                                      case 56:
                                      case 58:
                                      case 60:
                                      case 62:
                                      case 64:
                                      case 66:
                                      case 71:
                                      case 72:
                                      case 74:
                                      case 76:
                                      case 77:
                                          $template_id = 19;
                                          break;
                                      case 5:
                                      case 6:
                                      case 7:
                                      case 8:
                                      case 9:
                                      case 10:
                                      case 11:
                                      case 12:
                                      case 13:
                                      case 14:
```

```
                                case 15:
                                case 16:
                                case 17:
                                case 18:
                                case 19:
                                case 20:
                                case 22:
                                case 23:
                                case 24:
                                case 25:
                                case 26:
                                case 27:
                                case 28:
                                case 29:
                                case 30:
                                case 31:
                                case 32:
                                case 33:
                                case 34:
                                case 35:
                                case 36:
                                case 38:
                                case 47:
                                case 51:
                                case 52:
                                case 55:
                                case 57:
                                case 59:
                                case 63:
                                case 65:
                                case 68:
                                case 73:
                                    $template_id = 13;
                                    break;
                                default:
                                    $template_id = 21;
                                    break;
                            }

                            // Get the email data for the mail
action to send to the user.
                            $getEmailTemplate = "Select
template_text, template_subject from core_email_template where
template_id = $template_id";
                            $resultGetTemplate = $conn->query
($getEmailTemplate);


                            if($resultGetTemplate->num_rows > 0){
                                while($row5 = $resultGetTemplate-
>fetch_assoc()){
```

```
                                        $template_text =  $row5
['template_text'];
                                        $template_subject = $row5
['template_subject'];

                                        // Check if the user has
already received an email, then don't send them another.
                                        if(in_array($sentAlreadyArray,
$email)){
                                            break;
                                        }

                                        sendMailAction($template_id,
$template_text, $template_subject, $email, $customer_name,
$customer_id);
                                        array_push($sentAlreadyArray,
$email);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        else {
            break;
        }
    }
}

 ?>
```

CBDRX4U - Create Folders and .htaccess Files With List

This script was written to produce 695 separate folders and .htaccess files that would have required to be built manually otherwise. The main details from this for reference are file operations, file renaming, and htaccess file setup through string usage.

Code for Mac

```
<?php
    /*
     * Created by Joshua Greenert 10/22/2020
     *
     * This script will use PHP to create folders and .htaccess files
for each
     * subdomain on the cbdrx4u site.
     */

    // Make two arrays.  One of the domain names, and one for the url
```

```
path.
    $names = array();
    $urls = array();
    $pagepaths = array();

    // Read the file contents.
    $namesfile = fopen("names.txt", "r") or die("Unable to open file!");
    $urlsfile = fopen("urls.txt", "r") or die("Unable to open file!");

    // Add the contents to the arrays.
    while(!feof($namesfile)) {
        array_push($names, fgets($namesfile));
        array_push($urls, fgets($urlsfile));
    }

    fclose($urlsfile);
    fclose($namesfile);

    // Do all operations for every folder.
    for($i = 0; $i < count($names); $i++){

        // Create pagepaths to add to future string.
        $pagepath = substr($urls[$i], strpos($urls[$i],"find-us/")+8,
strlen($urls[$i]) - 28);
        $names[$i] = str_replace("\n", "", $names[$i]);
        $names[$i] = str_replace("?", "", $names[$i]);
        $pagepath = str_replace("\n", "", $pagepath);

        mkdir($names[$i]);

        // Create the string to enter into the file.
        $htstring = "RewriteEngine on\n\n<IfModule mime_module>\n
AddHandler application/x-httpd-ea-php74 .php .php7 .phtml".
        "\n</IfModule>\n# php -- END cPanel-generated handler, do not
edit\nRewriteCond %{HTTP_HOST} ^". $names[$i] .
        "\\.cbdrx4u\\.com$ [OR]\nRewriteCond %{HTTP_HOST} ^www\\.".
$names[$i] . "\\.cbdrx4u\\.com$\n".
        "RewriteRule ^/?$ \"https\\:\\/\\/cbdrx4u\\.com\\/find\\-us\\/".
strval($pagepath)."\" [R=301,L]\n\n" .
        "RewriteCond %{HTTP_HOST} ^". $names[$i] . "\\.cbdrx4u\\.com$
[OR]\nRewriteCond %{HTTP_HOST} ^www\\.".
        $names[$i] ."\\.cbdrx4u\\.com$\nRewriteRule (.*)$ \"https\\:\\
/\\/cbdrx4u\\.com\\/$1\" [R=301,L]";

        // Create the file for the folder path and insert the string.
Then, rename it.
        file_put_contents(strval($names[$i]."/htaccess.txt"),
$htstring);
        rename($names[$i]."/htaccess.txt", $names[$i]."/.htaccess");
```

```
        // Set variables back to null.
        $pagepath = "";
        $htstring = "";
    }
?>
```

CBDRX4U - Update All HTML Files To Replace TM with R

This is the script used to replace all '™' with '®' on the CBDRX4u site. To remain safe, the .htaccess file within the given directory and any other file not listed (such as .., ., etc.) were removed from files to update.

```
<?php
/*
 * Created by Joshua Greenert 10/24/2020
 *
 * This script will fix all html files for AMP to ensure that they
correct the
 * TM to the R symbol.
 *
 */

// Set the array and add all of the files to it.
$files = array_diff(scandir('/home/cbdrxforu/public_html/amp/'), array
('..', '.'));

// Use two separate for loops to remove the files that don't belong.  I
don't understand why
// they have to be separate to accurately work but they do.
for($i = 0; $i < count($files); $i++){
    $files[$i] = strval($files[$i]);
    if($files[$i] == '.htaccess'){
        unset($files[$i]);
    }
}
for($i = 0; $i < count($files); $i++){
    $files[$i] = strval($files[$i]);
    if($files[$i] == ''){
        unset($files[$i]);
    }
}

// Correct the values to ensure that the references [0][1] etc. are
accurate.
$files = array_values($files);

// Add the constants to update.
$oldMessage = '™';
$deletedFormat = '®';
```

```
// Loop through the list and update all references that show the tm.
for($i = 0; $i < count($files); $i++){

    //read the entire string
    $str=file_get_contents(strval($files[$i]));

    //replace something in the file string - this is a VERY simple
example
    $str=str_replace($oldMessage, $deletedFormat,$str);

    //write the entire string
    file_put_contents(strval($files[$i]), $str);

}

print_r($files);

?>
```

CBDRX4U - Update Zoo Item Links In Database

This page hosts the script used to update the URLs for every zoo item that had a match from the new site http://getsunmed.com . While there were ones that did not have a 1 to 1 match, this covered the majority of the URLs needed to be updated for the client. Username, password, and database name were hidden for security purposes.

The Code

```
<?php

/* Created by Joshua Greenert on 10/5/2021

 This script will use a database connection to get the data from the
zoo_item table to capture the json object.
 This is used to test whether or not it will be feasible to make the
updates for all sites there for the ones that
 have the links available.
*/

$servername = "localhost";
$username = "";
$password1 = "";
$dbName = "";

// Create connection
$conn = new mysqli($servername, $username, $password1, $dbName);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
```

```
echo "Connected successfully";

// Set up the query to return the json object.
$sqlGetNewElement = "SELECT lname, newelement FROM
cbdrxfor_cbdrxfor_you2021.newurl_replace2";
$resultGetNewElement = $conn->query($sqlGetNewElement);

if($resultGetNewElement->num_rows > 0){
  while($row = $resultGetNewElement->fetch_assoc()){

    $index = 0;
    $name = $row['lname'];

    echo $name;

    $newElement = $row['newelement'];
    $newElement = addslashes($newElement);

    $sqlGetZooItem = "SELECT * FROM dr0vc_zoo_item where name = '$name'
limit 1";
    $resultGetZooItem = $conn->query($sqlGetZooItem);

    if($resultGetZooItem->num_rows > 0){
      while($row2 = $resultGetZooItem->fetch_assoc()){

        if($index === 0){
          $id = $row2['id'];

          echo $id;
          $sqlUpdateElement = "Update dr0vc_zoo_item set elements =
'$newElement' where id = $id";
          if($conn->query($sqlUpdateElement) === TRUE){
            echo "Record Updated";
          }
          else{
            echo "Error updating record: " . $conn->error;
          }
          $index++;
        }

      }
    }
  }
}
?>
```

Mywatersidesouth - Forgot Username

This post will include the script for setting up a forgot user function. This one was used specifically for Waterside, but the details are useful in future functions and processes.

The Code

```
<?php

/*
    This script asks the user to enter their email address.  If a valid
email address is
    found, the username is sent to their email with a link to login;
otherwise, the user
    is advised that the email is not found.
*/

/********************************** Header & Connection
***********************************/
// BEGIN PHP
include('header.php');

?>

<?php

/********************************* FUNCTIONS
********************************/
function sendEmail($email, $username){

    // Set the message.
    $emailmsg = setMessage($username);

    $headers = "From: " . "no-reply@waterside.com" . "\r\n";
    $headers .= "MIME-Version: 1.0\r\n";
    $headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";

    // Send the email to the user.
    mail($email,"Password Reset Requested",$emailmsg, $headers);
}
// Set the message for the email; returns the message and requires the
pin.
function setMessage($username){
    $link = "https://watertest.dang-designs.com";

    $emailmsg = "<html><body>" .
        "<h3>Your Water Username</h3>" .
        "$username". "<br>" .
        "Please click the link below to reset your password.<br>" .
        "<a href=".$link.">Sign In</a>" .
        "<br>Email us at <a href='watertest@support.
com'>watertest@support.com</a> if you have any questions." .
        "</body></html>";

    return $emailmsg;
```

```
}
function hideLoginForm(){
    ?>
    <style>
        #desktop-nav {
            display: none;
        }

    </style>
    <?php
}
function emailNotFound(){
    ?>
    <style>
    .alert {
      padding: 20px;
      background-color: red;
      color: white;
      opacity: 1;
      transition: opacity 0.6s;
      margin-bottom: 15px;
    }
    .closebtn {
      margin-left: 15px;
      color: white;
      font-weight: bold;
      float: right;
      font-size: 22px;
      line-height: 20px;
      cursor: pointer;
      transition: 0.3s;
    }
    .closebtn:hover {
      color: black;
    }
    </style>
    <div class="alert success">
      <span class="closebtn">&times;</span>
      <strong>Error!</strong> Email not found.
    </div>
    <script>
        var close = document.getElementsByClassName("closebtn");
        close[0].onclick = function(){
            var div = this.parentElement;
            div.style.opacity = "0";
            setTimeout(function(){ div.style.display = "none"; }, 600);
        }
    </script>
    <?php
}
```

```
function emailFound(){
    ?>
    <style>
    .alert {
      padding: 20px;
      background-color: #4CAF50;
      color: white;
      opacity: 1;
      transition: opacity 0.6s;
      margin-bottom: 15px;
    }
    .closebtn {
      margin-left: 15px;
      color: white;
      font-weight: bold;
      float: right;
      font-size: 22px;
      line-height: 20px;
      cursor: pointer;
      transition: 0.3s;
    }
    .closebtn:hover {
      color: black;
    }
    </style>
    <div class="alert success">
      <span class="closebtn">&times;</span>
      <strong>Success!</strong> Please check your email for the
username.
    </div>
    <script>
        var close = document.getElementsByClassName("closebtn");
        close[0].onclick = function(){
            var div = this.parentElement;
            div.style.opacity = "0";
            setTimeout(function(){ div.style.display = "none"; }, 600);
        }
    </script>
    <?php
}
/********************************** END FUNCTIONS
********************************/

if(empty($_POST)){
        hideLoginForm();
}
else{
    // Set the variable for the email.
    if(isset($_POST['email'])){
        $email = $_POST['email'];
```

```
        $sql="SELECT * from dashboard_users where email = '$email'";
            $result = $conn->query($sql);

        if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
                $user = $row['username'];
                sendEmail($email, $user);
                emailFound();
                hideLoginForm();
            }
        }
        else{
            emailNotFound();
            hideLoginForm();
        }
    }
}
?>


<style>
#username-form{
    background: #EDEDED;
    padding: 30px 30px 10px 30px;
    border-style: solid;
    border-color: lightgrey;
    width: 40%;
        margin: 170px auto;
        border-radius: 3px;
}
#username-form #homepage {
  background-color: #EDEDED;
  border: none;
  color: #0089C7;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  cursor: pointer;
  border-radius: 16px;
  width: 130px;
  border: none;
  width: 100%;
  font-weight: bold;
}
#username-form #homepage:hover {
  color: #002B61;
  outline: none;
}
#email-label{
```

```
    text-align: center;
    margin-bottom: 20px;
    color: #002B61;
}
#email-textbox{
    border-style: solid;
    border-color: #FDA54B;
    font-weight: bold;
        width: 100%;
        text-align: center;
        cursor: center;
        padding: 10px 10px 10px 10px;
}
#user-button{
  background-color: #0089C7;
  border: none;
  color: white;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  margin: 10px 2px;
  cursor: pointer;
  border-radius: 16px;
  width: 130px;
  border: none;
  line-height: 40px;
  width: 100%;
  padding-bottom: 5px;
  padding-top: 10px;
  font-weight: bold;
}
#user-button:hover{
  background-color: #002B61;
  color: white;
  border-color: #0089C7;
}
@media only screen and (max-width: 1000px) {
  #username-form{
    width: 70%;
  }
}
@media only screen and (max-width: 600px) {
  #username-form{
    width: 80%;
  }
}
</style>
<div class="holder">
    </div>
```

```
        </div>
            <form method="post" action="" id="username-form">
              <h2 id="email-label">Please enter your email</h2>
              <input type="email" name="email" id="email-textbox"
placeholder="Email Address" required>
              <input type="submit" name="submit" id="user-button"
value="Submit">
              <input class="homepage"  type="button" name="homepage"
id="homepage" onclick="goHome()" value="Return to Login">
            </form>
        <script>
            function goHome(){
                window.location.assign('index.php');
            }
        </script>
<br><br><br>


<?php include('footer.php'); ?>
```

Mywatersidesouth - Forgot Password

This page will store a copy of the code used for the forgot password process on the waterside dev site currently.

The Code

```
<?php
// BEGIN PHP
include('header.php');

/*********************************** FUNCTIONS
***********************************/
// Set the function that creates a random pin and starts the time clock
for entry.
function sendEmail($dbfound, $email, $userid, $pincode){

    // Set the hash variables.
    $newdbfound = password_hash($dbfound, PASSWORD_DEFAULT);

    // Set the message.
    $emailmsg = setMessage($newdbfound, $userid, $pincode);

    $headers = "From: " . "no-reply@waterside.com" . "\r\n";
    $headers .= "MIME-Version: 1.0\r\n";
    $headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";

    // Send the email to the user.
    mail($email,"Password Reset Requested",$emailmsg, $headers);
    hideLoginForm();
```

```
    emailFound();
}
// Set the message for the email; returns the message and requires the
pin.
function setMessage($newdbfound, $userid, $pincode){
    $link = "https://watertest.dang-designs.com/reset.php?&source="
.$newdbfound. "&pin=" .$pincode. "&id=".$userid;

    $emailmsg = "<html><body>" .
        "<h3>Forgot Password Link Requested</h3>" .
        "Please click the link below to reset your password.<br>" .
        "<a href=".$link.">Reset Password</a>" .
        "<br>Email us at <a href='watertest@support.
com'>watertest@support.com</a> if you have any questions." .
        "</body></html>";

    return $emailmsg;
}
function hideLoginForm(){
    ?>
    <style>
        #desktop-nav {
            display: none;
        }

    </style>
    <?php
}
function emailNotFound(){
    ?>
    <script>
        window.alert("Username Not Found");
    </script>
    <?php
}
function emailFound(){
    ?>
    <style>
    .alert {
      padding: 20px;
      background-color: #4CAF50;
      color: white;
      opacity: 1;
      transition: opacity 0.6s;
      margin-bottom: 15px;
    }
    .closebtn {
      margin-left: 15px;
      color: white;
      font-weight: bold;
```

```
      float: right;
      font-size: 22px;
      line-height: 20px;
      cursor: pointer;
      transition: 0.3s;
    }
    .closebtn:hover {
      color: black;
    }
    </style>
    <div class="alert success">
      <span class="closebtn">&times;</span>
      <strong>Success!</strong> Please check your email to complete the
password reset process.
    </div>
    <script>
        var close = document.getElementsByClassName("closebtn");
        close[0].onclick = function(){
            var div = this.parentElement;
            div.style.opacity = "0";
            setTimeout(function(){ div.style.display = "none"; }, 600);
        }
    </script>
    <?php
}
/*********************************** END FUNCTIONS
***********************************/

// Run the SQL statement to check for the email and report back to the
user.
if (!empty( $_POST ) ) {
    if (isset( $_POST['fpemail'] )) {

        // Getting submitted user data from database and set pin/pincode
        $email = $_POST['fpemail'];
        $pin = rand();
                $pincode = password_hash($pin, PASSWORD_DEFAULT);

        $sql="SELECT * from res_profile where email = '$email'";
            $result = $conn->query($sql);

        if ($result->num_rows > 0) {
            // output data of each row
            while($row = $result->fetch_assoc()) {
                    $email = $row["email"];
                    $userid = $row["id"];

                    if($email != null && $email != ""){

                        // Delete the pin if stored in the table and
```

```
generate the new one.
                        $deletepin = "delete from passwordresets where
userid = $userid and type = 'res'";
                        if ($conn->query($deletepin) === TRUE) {}
                        else {
                                    echo "Error line 2: " . $deletepin
. "<br>" . $conn->error;
                                    }

                        $addpin = "insert into passwordresets (userid,
pin, pincode, type) values ($userid, $pin, '$pincode', 'res')";
                        if ($conn->query($addpin) === TRUE) {

                                        // Set up the display for when
the user has successfully sent their email.
                                            $dbfound = "res_profile";
                        sendEmail($dbfound, $email, $userid, $pincode);
                        break;
                                    } else {
                                    echo "Error line 2: " . $addpin .
"<br>" . $conn->error;
                                    }
                    }
            }
        }
        else {
            // Set the query for the admin table and check for the
username again.
                $getuser = "select * from dashboard_users where email =
'$email'";
                $userresult = $conn->query($getuser);

                if($userresult->num_rows > 0){
                         while($userrow = $userresult->fetch_assoc()) {
                    $email = $userrow["email"];
                    $userid = $userrow["userid"];

                    if($email != null && $email != ""){

                        // Delete the pin if stored in the table and
generate the new one.
                            $deletepin = "delete from passwordresets
where userid = $userid and type = 'admin'";
                            if ($conn->query($deletepin) === TRUE) {
                                              //echo "New record
created successfully";
                                        } else {
                                        echo "Error line 2: " .
$deletepin . "<br>" . $conn->error;
                                        }
```

```
                            $addpin = "insert into passwordresets
(userid, pin, pincode, type) values ($userid, $pin, '$pincode',
'admin')";
                            if ($conn->query($addpin) === TRUE) {

                                                // Set up the display
for when the user has successfully sent their email.
                                                $dbfound =
"dashboard_users";
                            sendEmail($dbfound, $email, $userid,
$pincode);
                            break;
                                        } else {
                                        echo "Error line 2: " . $addpin
. "<br>" . $conn->error;
                                        }
                        }
                    }
                }
                else {

                                emailNotFound();
                                hideLoginForm();
                    echo "<script>window.location.assign('forgotpass.
php');</script>";
                }
        }
    }
}
else {

// END PHP
?>
<style>

/* Adding the page for forgotpass.php */
#desktop-nav {
    display: none;
}
.password_reset_form #reset_button {
  background-color: #0089C7;
  border: none;
  color: white;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  margin: 10px 2px;
  cursor: pointer;
```

```
  border-radius: 16px;
  width: 130px;
  border: none;
  line-height: 40px;
  width: 100%;
  padding-bottom: 5px;
  padding-top: 10px;
  font-weight: bold;
}
.password_reset_form #homepage {
  background-color: #EDEDED;
  border: none;
  color: #0089C7;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  cursor: pointer;
  border-radius: 16px;
  width: 130px;
  border: none;
  width: 100%;
  font-weight: bold;
}
.password_reset_form #reset_button:hover {
  background-color: #002B61;
  color: white;
  border-color: #0089C7;
}
.password_reset_form #homepage:hover {
  color: #002B61;
  outline: none;
}
#fpemail{
    border-style: solid;
    border-color: #FDA54B;
    font-weight: bold;
        width: 100%;
        text-align: center;
        cursor: center;
        padding: 10px 10px 10px 10px;
}
.password_reset_form{
    background: #EDEDED;
    padding: 30px 30px 10px 30px;
    border-style: solid;
    border-color: lightgrey;
    width: 40%;
        margin: 170px auto;
        border-radius: 3px;
}
```

```
.password_reset_form ::placeholder{
    text-align: center;
}
#email-label{
    text-align: center;
    margin-bottom: 20px;
    color: #002B61;
}
@media only screen and (max-width: 1000px) {
  .password_reset_form{
    width: 70%;
  }
}
@media only screen and (max-width: 600px) {
  .password_reset_form{
    width: 80%;
  }
}

</style>
</head>
    <div class="holder">
                        </div>
                            </div>
                    <form class="password_reset_form" action="" method="
post">
                        <h2 id="email-label">Please enter your email<
/h2>
                                        <div class="fpemail"><input
type="email" name="fpemail" id="fpemail" placeholder="Email Address"
required></div>
                                            <div><input class="
reset_button"  type="submit" name="reset" id="reset_button" value="
Reset Password">
                                            <input class="homepage"
type="button" name="homepage" id="homepage" onclick="goHome()" value="
Return to Login">
                            </form>
                    <script>
                        function goHome(){
                            window.location.assign('index.php');
                        }
                    </script>

<?php }
include('footer.php');
?>
```

This page will host the script used for reseting the password on the waterside dev site.

The Code

```
<?php
/* Created by Joshua Greenert on 9/22/2020
This file will provide the user with an interface to update their
password.  When
the user provides a password to both fields, the program will check
whether it meets
all password criteria, then updates the database with the information.
*/

include('header.php');

/********************************** FUNCTIONS
**********************************/

function compareDate($date){

    $moddate = date("Y-m-d H:i:s", $date);
    $today = date("Y-m-d H:i:s");
    if((strtotime($today)-(60*60*24)) < strtotime($moddate)){
        return false;
    }
    else{
        return true;
    }
}

function sendSuccess(){
    ?>
    <style>
    label, #resetform{
        display: none;
    }
    .alert {
      padding: 20px;
      background-color: #4CAF50;
      color: white;
      opacity: 1;
      transition: opacity 0.6s;
      margin-bottom: 15px;
    }
    .closebtn {
      margin-left: 15px;
      color: white;
      font-weight: bold;
      float: right;
      font-size: 22px;
```

```
      line-height: 20px;
      cursor: pointer;
      transition: 0.3s;
    }
    .closebtn:hover {
      color: black;
    }
    </style>
    <div class="alert success">
      <span class="closebtn">&times;</span>
      <strong>Password reset Successful!</strong> Return to the
homepage to login.
    </div>
    <input class="homepage"  type="button" name="homepage" id="
homepage" onclick="goHome()" value="Return to Login">
    <script>
        var close = document.getElementsByClassName("closebtn");
        close[0].onclick = function(){
            var div = this.parentElement;
            div.style.opacity = "0";
            setTimeout(function(){ div.style.display = "none"; }, 600);
        }
        function goHome(){
            window.location.assign('index.php');
        }
    </script>
    <?php
}
function sendError(){
    ?>
    <style>
    #desktop-nav {
        display: block;
    }
    label, #resetform{
        display: none;
    }
    .alert {
      padding: 20px;
      background-color: #0089C7;
      color: white;
      opacity: 1;
      transition: opacity 0.6s;
      margin-bottom: 15px;
      text-align: center;
    }
    .closebtn {
      margin-left: 15px;
      color: white;
      font-weight: bold;
```

```
      float: right;
      font-size: 22px;
      line-height: 20px;
      cursor: pointer;
      transition: 0.3s;
    }
    .closebtn:hover {
      color: black;
    }
    </style>
    <div class="alert error">
      <span class="closebtn">&times;</span>
      <strong>Error!</strong> The code is no longer valid.  You must
resubmit through the forgotten password link on the left.
    </div>
    <script>
        var close = document.getElementsByClassName("closebtn");
        close[0].onclick = function(){
            var div = this.parentElement;
            div.style.opacity = "0";
            setTimeout(function(){ div.style.display = "none"; }, 600);
        }
    </script>
    <?php
}

/********************************** END FUNCTIONS
***********************************/

// Get the GETS and continue, or show the user login form.
if(isset($_GET["source"]) && isset($_GET["pin"]) && isset($_GET["id"]))
{

    // Set the URL pieces to variables.
    $codesource = $_GET["source"];
    $pincode = $_GET["pin"];
    $userid = $_GET["id"];

    ?>
    <style>
        #desktop-nav {
            display: none;
        }
    </style>
      <br />
      <label>Password Requirements: 6 - 10 characters (numbers or
letters)</label>
      <form method="post" action="" name="update" id="resetform"
onsubmit="return checkPassword()">
          <input type="hidden" name="action" value="update" />
```

```
            <br /><br />
            <label><strong>Enter New Password:</strong></label><br />
            <input type="password" name="pass1" id="pass1" minlength="6"
maxlength="10" required />
            <br /><br />
            <label><strong>Re-Enter New Password:</strong></label><br />
            <input type="password" name="pass2" id="pass2" minlength="6"
maxlength="10" onchange="style.backgroundColor='white';" required/>
            <br /><br />
            <input type="submit" value="Reset Password" />
        </form>
        <script>
            function checkPassword(){
                var pass1 = document.getElementById("pass1").value;
                var pass2 = document.getElementById("pass2").value;

                if(pass1 != pass2){
                    alert('Passwords do not match!');
                    document.getElementById("pass2").focus();
                    document.getElementById("pass2").style.
backgroundColor="yellow";
                    return false;
                }
                else{
                    return true;
                }
            }
        </script>
    <?php

    // Check for the post from the user to check the results.
    if(empty($_POST)){

    }
    else{

        // Check the passwords that the user entered.
        $pass1 = $_POST['pass1'];
        $pass2 = $_POST['pass2'];

        // Hash the password field to be ready for input.
        $passhash = password_hash($pass1, PASSWORD_DEFAULT);

        // Set the table name by checking which table the source came
from.
        if(password_verify("res_profile", $codesource)){

            // Search the users for the one that matches the code.
            $sql = "SELECT pin, requestdate FROM `passwordresets` where
userid = " .$userid;
```

```
            $result = $conn->query($sql);

            if($result->num_rows > 0){
                while($row = $result->fetch_assoc()){
                    $pin = $row["pin"];
                    $date = $row["requestdate"];

                    if(password_verify($pin, $pincode) && compareDate
($date)){

                        $newsql = "UPDATE res_profile SET `res_profile`.
password = '" . $passhash . "' WHERE `res_profile`.id = '".$userid."'";
                        mysqli_query($conn, $newsql);
                        sendSuccess();
                        break;
                    }
                    else{
                        sendError();
                    }
                }
            }
        }
        else if(password_verify("dashboard_users", $codesource)){

            // Search the users for the one that matches the code.
            $sql = "SELECT pin, requestdate FROM `passwordresets` where
userid = " .$userid;
            $result = $conn->query($sql);

            if($result->num_rows > 0){
                while($row = $result->fetch_assoc()){
                    $pin = $row["pin"];
                    $date = $row["requestdate"];

                    if(password_verify($pin, $pincode) && compareDate
($date)){
                        $newsql = "UPDATE dashboard_users SET
`dashboard_users`.password = '" . $passhash . "' WHERE
`dashboard_users`.userid = '" .$userid. "'";
                        mysqli_query($conn, $newsql);
                        sendSuccess();
                        break;
                    }
                    else{
                        sendError();
                    }
                }
            }
        }
        else{
```

```
                sendError();
            }
        }
    }
    else{
        sendError();
    }
    ?>
```

SunFlora - Setting Up Cloud Storage Bucket From PHP Script

This page will discuss how to set up a bucket for cloud storage of files and, essentially, how to create the location where all back up files will be stored. These steps include: setting up the storage bucket, creating storage API credentials,

1. **Set Up the Storage Bucket > {**
    a. [Either type in "Storage" in the search bar, or select browser storage on the GCP menu]
    b. [Select Create Bucket]
    c. [Set the name for the bucket and choose multi-regional]
        i. *The choice of multi-regional allows the data to have redundancy options*
    d. [Set object-level and bucket-level permissions for the access control model]
        i. *This means that the bucket will be set as private or public based on the user's settings*
        ii. *The alternative is that all buckets are set with public access*
    e. [Press Create to create the instance of the bucket] **}**
2. **Create API Credentials for Storage Bucket > {**
    a. [Type "API" in the search bar and select "Credentials"]
    b. [Select "Create Credentials" and select "Service account key"]
    c. [Select "New service account"]
        i. [Set the name for the service account]
        ii. [Set the role as Storage Admin]
        iii. [Choose JSON file type for the bucket]
    d. [Press "Create" to create the service account key]
        i. *A file will be downloaded afterwards containing JSON credentials. }*
        ii. *This key should be kept secure and removed from visibility within code to ensure that others cannot use or access the key; it is a private key that must be used to access a secure connection with the Google Cloud Storage*
3. **Install Composer and Download Google Cloud Storage Library > {**
    a. [Composer download is available at https://getcomposer.org/download/]
        i. *Composer is the suggested dependency manager for PHP and supports the action of grabbing the library specific to Google Cloud Storage.*
        ii. *By default, the composer.phar file will be stored on the Desktop for Mac.*
    b. [Move the composer.phar file to where it can be accessed globally]
        i.
        ```
        sudo mv composer.phar /usr/local/bin/composer
        ```
        ii. *Root privileges are need to move this file*
    c. [Confirm that the composer is now accessible through terminal]
        i.
        ```
        composer --version
        ```
        ii. *If this returns anything but the composer version, try again.*
    d. [Download the Google Cloud Storage Library Only]
        i.
        ```
        composer require google/cloud-storage
        ```
            1. *This will download the Google Cloud storage library, create the composer.json file, and unpack the vendor /autoload.php file needed for secure connections to the Google Cloud Storage instance.* ***}***
4. **Make the code > {**
    a. [Include the autoloader and the imported Google Client Library at the top of the script file]

i.
```
require_once 'vendor/autoload.php';
```

ii.
```
use Google\Cloud\Storage\StorageClient;
```

b. *Since these next steps are based off of research and preference for the process, the details will be listed as a reference for when users attempt similar situations. At the end of the process, when everything is completed, the full script used will be displayed.*
c. [Check the OS type to determine the path variable to use]

i.
```
if(php_uname('s') == "Linux"){

}
else if(php_uname('s') == "Windows NT"){

}
```

1. *These items have been tested on each machine to ensure that they produce the result we see*

d. [The important two details for transferring files to the Google Cloud storage are listed below]

i.
```
// Set up the private key file content to send over.
$privateKeyFileContent = '{
    "type":
    "project_id":
    "private_key_id":
    "private_key":
    "client_email":
    "client_id":
    "auth_uri":
    "token_uri":
    "auth_provider_x509_cert_url":
    "client_x509_cert_url":
}';
```

ii.
```
// Set the function that will upload the file to google
Cloud Storage.
function upload_object($bucketName, $fileName, $source)
{

    $privateKeyFileContent = $GLOBALS
['privateKeyFileContent'];

    // Create the storage client and store the file in
the cloud.
    $storage = new StorageClient([
```

```
        'keyFile' => json_decode($privateKeyFileContent,
true)
    ]);

    $file = fopen($source, 'r');
    $bucket = $storage->bucket($bucketName);
    $object = $bucket->upload($file, [
        'name' => $fileName
    ]);
    // Print the success message to the user.
    printf('Uploaded %s to gs://%s/%s' . PHP_EOL, basename
($source), $bucketName, $fileName);

}
```

5. **You're done!**
   a. **Here is a copy of the code used to produce the final outcome.**
   b. *The JSON file information was removed from this instance to promote security*
   c.

```
<?php
/*
 * Created by Joshua Greenert 6/29/2020
 *
 * This script will create the baseline for the process of
storing the database file from Fishbowl
 * on the Google Cloud Storage Bucket.  Using the composer-
support library, the Google Storage Client
 * is able to open the connection and send the information to
the bucket instance.
 *
 */

require_once '../vendor/autoload.php';
use Google\Cloud\Storage\StorageClient;

// Set the parameters for Google cloud bucket name, filename,
and path to file.
$bucketName = "fishbowl-storage";
$filename = "";
$dataBackupSource = "";
$zipfilename = "";

// Obtain file through use of glob and current date with wild
cards.
$globfilename = glob("../../Sunflora_Inventory_" . date("Y-m-
d") . "-02-**.sql", GLOB_BRACE);

// Set the glob file to the file name.
foreach($globfilename as $file){
    $filename = $file;
```

```
}

// substring the contents to ensure that the file name is
correct.
$filename = substr($filename, 6, strlen($filename));
$zipfilename = $filename . ".zip";

// Set the function that will upload the file to google Cloud
Storage.
function upload_object($bucketName, $fileName, $source)
{

    $privateKeyFileContent = $GLOBALS['privateKeyFileContent'];

    // Create the storage client and store the file in the
cloud.
    $storage = new StorageClient([
        'keyFile' => json_decode($privateKeyFileContent, true)
    ]);

    $file = fopen($source, 'r');
    $bucket = $storage->bucket($bucketName);
    $object = $bucket->upload($file, [
        'name' => $fileName
    ]);
    // Print the success message to the user.
    printf('Uploaded %s to gs://%s/%s' . PHP_EOL, basename
($source), $bucketName, $fileName);

}

// Set up the private key file content to send over.
$privateKeyFileContent = '{
    "type": "service_account",
    "project_id": "dang-test-space",
    "private_key_id":
"7115a3242b4cf91606ea7a7fa2d3f80595bd0152",
    "private_key": "",
    "client_email": "fishbowl-storage@dang-test-space.iam.
gserviceaccount.com",
    "client_id": "115349375276367438503",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url": "https://www.googleapis.com
/oauth2/v1/certs",
    "client_x509_cert_url": "https://www.googleapis.com/robot
/v1/metadata/x509/fishbowl-storage%40dang-test-space.iam.
gserviceaccount.com"
}';
```

```
// Check the OS to determine whether Linux or Windows.
if(php_uname('s') == "Linux"){

    // Set the file path to grab back up and delete file
afterwards.
    $dataBackupSource = "/opt/Fishbowl/data/backups/" .
$filename;
    $deleteFilePathFile = "/opt/Fishbowl/data/backups/" .
$filename;
    $deleteFilePathZip = "/opt/Fishbowl/data/backups/" .
$zipfilename;

    // Create a zip file to place the file in before sending
over.
    $zip = new ZipArchive;
    if($zip->open($zipfilename, ZipArchive::CREATE)===TRUE){

       $zip->addFile($filename);
       $zip->close();
    }

    // Rename the file to change the directory location.
    rename($zipfilename, '/opt/Fishbowl/data/backups/' .
$zipfilename);

    // Use try/catch statement to catch issues with private
key.
    try{

        // run the function to place both backup files in the
Google Cloud Storage Bucket instance.
        upload_object($bucketName, $zipfilename,
$dataBackupSource);

        // Delete the old file stored in the path and the zip
file.
        if(!unlink($deleteFilePathFile)){
            print ("The file cannot be deleteted or isn't
present in this directory...");
        }
        else{
            print ("The file " . $filename . " has been
deleted.");
        }

        if(!unlink($deleteFilePathZip)){
            print ("The zip file cannot be deleteted or isn't
present in this directory...");
        }
        else{
```

```
            print ("The zip file " . $zipfilename . " has been
deleted.");
        }
    }
    catch (Exception $e){
        print "This process failed for the following reason: "
. $e;
        return false;
    }
}
else if(php_uname('s') == "Windows NT"){

    // Set the file path to grab back up.
    $dataBackupSource = "C:\\Program
Files\\Fishbowl\\data\\backups\\" . $filename;
    $deleteFilePathFile = "C:\\Program
Files\\Fishbowl\\data\\backups\\" . $filename;
    $deleteFilePathZip = "C:\\Program
Files\\Fishbowl\\data\\backups\\" . $zipfilename;

    // Create a zip file to place the file in before sending
over.
    $zip = new ZipArchive;
    if($zip->open($zipfilename, ZipArchive::CREATE)===TRUE){

       $zip->addFile($filename);
       $zip->close();
    }

    // Rename the file to change the directory location.
    rename($zipfilename, "C:\\Program
Files\\Fishbowl\\data\\backups\\" . $zipfilename);

    // Use try/catch statement to catch issues with private
key.
    try{

        // run the function to place both backup files in the
Google Cloud Storage Bucket instance.
        upload_object($bucketName, $zipfilename,
$dataBackupSource);

        // Delete the old file stored in the path.
        if(!unlink($deleteFilePathFile)){
            print ("The file cannot be deleteted or isn't
present in this directory...");
        }
        else{
            print ("The file " . $filename . " has been
deleted.");
```

```
        }

        if(!unlink($deleteFilePathZip)){
            print ("The zip file cannot be deleteted or isn't present in this directory...");
        }
        else{
            print ("The zip file " . $zipfilename . " has been deleted.");
        }
    }
    catch (Exception $e){
        print "This process failed for the following reason: " . $e;
        return false;
    }
}
else{
    print "The operating system has another name...";
}

?>
```

References
https://www.youtube.com/watch?v=pGSzMfKBV9Q
https://zatackcoder.com/upload-file-to-google-cloud-storage-using-php/
https://getcomposer.org/download/
https://cloud.google.com/storage/docs/reference/libraries#client-libraries-install-php

Operating System Specifics

This page will provide some OS specifics that will assist in managing the file directories and code needed to perform such steps. Unlike previous navigational guides, this documents serves as a reference page to provide information crucial for Google Cloud Storage file upload execution; in other words, the lines will be randomly listed and not step-by-step.

Windows Operating System	
'vendor/autoload.php' Path	'../vendor/autoload.php'
Certificate Issues are due to missing cert.pem file by curl and php.ini file requires path.	https://stackoverflow.com/questions/21114371/php-curl-error-code-60
Google cloud storage cURL error 60	https://stackoverflow.com/questions/35638497/curl-error-60-ssl-certificate-prblm-unable-to-get-local-issuer-certificate 1. open `vendor\guzzlehttp\guzzle\src\Handler\CurlFactory.php` 2. and change this

```
$conf[CURLOPT_SSL_VERIFYHOST] =
2;
$conf[CURLOPT_SSL_VERIFYPEER] =
true;
```

to this

```
$conf[CURLOPT_SSL_VERIFYHOST] =
0;
$conf[CURLOPT_SSL_VERIFYPEER] =
FALSE;
```

Linux Operating System	
‘vendor/autoload.php’ Path	‘../vendor/autoload.php’

Unlink not recognizing file path to delete file	File path must be specified by new source
	`$testSource = "/opt/Fishbowl/data/backups/" . $fileNameA;`
	Additionally, the path requires permissions for the user. Since there is only one user on the system, this command only provides permissions to the user.
	`chmod ugo+rwx /opt/Fishbowl/data/backups`

Document location to run	/home/support/Documents/FishbowlBackupGCP.php

SunFlora - 3PL API Import

This page hosts the code used to pull data from the 3PL API endpoint to capture the date, reference ID, and tracking number for orders sent there. As this data has not been checked yet, there may be more updates to come on its structure and setup. Regardless, it serves as documentation for future API connections.

The Code

```
<?php
/*
 * This script pulls data from the 3PL API to capture the process date,
reference ID,
 * and the tracking number for order sent within a specified date range.
 */
include('export3pl.php');
```

```
$curl2 = curl_init();

curl_setopt_array($curl2, array(
  CURLOPT_URL => 'https://secure-wms.com/orders?
pgsiz=100&pgnum=1&rql=readonly.processDate=gt=2021-06-11T17:16:00;
readonly.processDate=lt=2021-11-11T17:17:59&detail=All&itemdetail=All',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'GET',

  CURLOPT_HTTPHEADER => array(
   'Content-Type: application/json',
    'Accept: application/hal+json',
    'Accept-Language: en-US, en;q=0.8',
    "Authorization: Bearer $accesstoken"
  ),
));

// Response from curl_exec is json.  Decode it then request its data.
$response2 = curl_exec($curl2);
$jsondecodedresponse = json_decode($response2, true);

// Variables
$orderArray = array();
$skuTempArray = array();
$skuList = "";
$skuCountCorrection = 0;

$orderTotalCount = count($jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order']);

// For loop through the orders to get the contents for each.
for($i = 0; $i < $orderTotalCount; $i++){

    $orderId = $jsondecodedresponse['_embedded']['http://api.3plCentral.
com/rels/orders/order'][$i]['readOnly']['orderId'];
    $referenceNum = $jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order'][$i]['referenceNum'];
    $creationDate = $jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order'][$i]['readOnly']['creationDate'];
    $trackingNum = $jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order'][$i]['routingInfo']['trackingNumber'];

    // Get the skus involved with the order.
    $skuCount = count($jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order'][$i]['billing']['billingCharges'][0]
```

```
['details']);

    if($skuCount === 0){
        $skuCountCorrection = count($jsondecodedresponse['_embedded']
['http://api.3plCentral.com/rels/orders/order'][$i]['_embedded']
['http://api.3plCentral.com/rels/orders/item']);
    }

    // Add the skus to the list and account for the sku count
correction when needed.
    if($skuCount === 0){
        for($j = 0; $j < $skuCountCorrection; $j++){
            $skuTemp = $jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order'][$i]['_embedded']['http://api.
3plCentral.com/rels/orders/item'][$j]['itemIdentifier']['sku'];
            array_push($skuTempArray, $skuTemp);
            $skuTemp = "";

            if($j+1 == $skuCountCorrection){
                $skuList = implode(" , ",$skuTempArray);
            }
        }
    }
    else{
        for($j = 0; $j < $skuCount; $j++){
            $skuTemp = $jsondecodedresponse['_embedded']['http://api.
3plCentral.com/rels/orders/order'][$i]['billing']['billingCharges'][0]
['details'][$j]['sku'];

            // Skip elements that are empty.
            if($skuTemp === ""){}
            else{
                array_push($skuTempArray, $skuTemp);
            }

            $skuTemp = "";

            if($j+1 == $skuCount){
                $skuList = implode(" , ",$skuTempArray);
                $skuList = substr($skuList, 0, strlen($skuList));
            }
        }
    }

    // Set the items into the order array.
    array_push($orderArray, [$orderId,$creationDate, $referenceNum,
$trackingNum, $skuList ]);

    // Clear the temp array for the next iteration.
    $skuTempArray = array();
```

```
    $skuList = "";
}


for($i = 0; $i < count($orderArray); $i++){
    echo "Order ID: ".$orderArray[$i][0];
    ?> <br> <?php
    echo "Creation Date: ".$orderArray[$i][1];
    ?> <br> <?php
    echo  "Reference Number: ". $orderArray[$i][2];
    ?> <br> <?php
    echo  "Tracking Number: ". $orderArray[$i][3];
    ?> <br> <?php
    echo  "SKU: " .$orderArray[$i][4];
    ?> <br> <br><?php
}
```

JavaScript Projects

This page will host the collection of JavaScript pieces that have been worked on/with.

- Set Store Image Via Cookie (Delayed & NOT In Use)
- How To Create A Custom Input Field Through JavaScript

Set Store Image Via Cookie (Delayed & NOT In Use)

Saving this information to use in the future when applicable.

The Code

```
<script>
window.onload = function() {
    //SetStoreImage();
    HideCurrentStore()
};
function SetStoreImage(){
  // Get the cookie value and set it to the store name.
  <?php
  $cookieValue = $_COOKIE['store'];
  $imagePath = "";

  if($cookieValue == "sofg_fda"){
      $imagePath = "/media/images/websites/FL-DOA.png";
  }
  else if($cookieValue == "sofg_fdep"){
      $imagePath = "/media/images/websites/FL-FDEP.jpg";
  }
  else if($cookieValue == "sofg_fdfs"){
      $imagePath = "/media/images/websites/FL-DFS.png";
  }
  else if($cookieValue == "sofg_fsp"){
      $imagePath = "/media/images/websites/FL-DEP-PARK-SERVICE.png";
```

```
    }
    else if($cookieValue == "stateofflorida"){
        $imagePath = "/images/shoebox-logo-login.png";
    }
    ?>

    var cookieValue = "<?php echo $imagePath;?>";

    if(cookieValue == ""){}
    else{
        document.getElementById("StoreSelectionImage").src = cookieValue;
    }
}
</script>
<img src="" id="StoreSelectionImage" width="100"/>
```

How To Create A Custom Input Field Through JavaScript

While this piece was not able to be utilized for its purpose in being written, it may still have a purpose for future sites. Placing here to use for future reference.

The Code

```
var hiddenInput = document.createElement("input");
hiddenInput.type = "hidden";
hiddenInput.name = "items[<?php echo $counter?>].customization<?php
echo $counter?>";
hiddenInput.value = customizationData;

var hiddenElement = document.getElementsByTagName("tbody")[0];
hiddenElement.appendChild(hiddenInput);
```

WordPress Projects

This page will host all of the WordPress projects that utilize specific WordPress methods and architecture. This definition therefore segments the items placed in this umbrella to then include JavaScript, PHP, etc.

- 360twin - Universal Checkbox Confirmation

360twin - Universal Checkbox Confirmation

This page will host the process and data used to set up a confirmation window that appears when a user makes a change to the universal vehicle checkbox within products on 360twin. This checkbox is important since it determines whether or not the motorcycle part chosen is available for all bikes, or if it is specific to a host of vehicles.

There are two methods that are covered here. The first one being the main one to use instead of the secondary one since only one file is updated, and all data works congruently with it. The update is made to the admin.js unders wp-content/themes/Mobimax/js/admin.js. A quick way to find this function is to search the name of the function “toggleOffCar”

```
if (checked) {

      if (confirm("Are you sure you want to make this change?")) {
```

```
            $('body').append('<div class="uni-loading">Working...</div>');

            $('#product-vehiclechecklist input[type="checkbox"]').each
(function(){
                $(this).prop("checked", true);
            });

            setTimeout(function(){
                $('.uni-loading').remove();
            },3000);
        }
        else{
            // revert the option back to rememberOption state.
            document.getElementById
('enovathemes_addons_universal_product').checked = false;
        }


    } else {

        if (confirm("Are you sure you want to make this change?")) {
            $('#product-vehiclechecklist input[type="checkbox"]').each
(function(){
                $(this).prop("checked", false);
            });
        }
        else{
            // revert the option back to rememberOption state.
            document.getElementById
('enovathemes_addons_universal_product').checked = true;
        }


    }
```

Creating A WordPress Function (NOT USED)

Step 1

In order to effect the functions of the wp-admin section of the website, the functions.php within the main theme folder (*should be child theme for 360 but instead was in Mobimax*) will need a function call to hook into the necessary set of WordPress scripts. Additionally, to avoid this script from being loaded on all admin pages, the $hook variable can be used to identify the specific page within admin pages that you are looking to have it work on (*in this case post.php*).

Important Details:

1. **wp_enqueue_script has two arguments in this example**
    a. universal-check which aligns to the name of the file intended to be ran on execution
    b. The path specified below works when the file is placed in the root folder of public_html; other paths can be explored for better practices moving forward.
2. **The action must be added directly after the function to add the function to the scripts that are set to run in the queue.**
    a. The name of the function (2nd argument) must match the name of the function.
    b. admin_enqueue_scripts is used for wp-admin-related function through the site.

Functions.php

```
/*                Personally included functions
        Created by Joshua Greenert on 9/8/2021

        This script looks for wp-admin section where edit.php is being
used.  Then calls the script saved on the root
        of public_html where the name of the file is listed as
universal-check.js.
    */
    function universal_part_check_enqueue($hook) {
        // Only add to the edit.php admin page.
        if ('post.php' !== $hook) {
            return;
        }
        wp_enqueue_script('universal-check', '/universal-check.js');
    }

    add_action('admin_enqueue_scripts', 'universal_part_check_enqueue');
```

Step 2

The file listed above was required to be ran with other scripts; however, due to the speed of JavaScript, the function needed to wait until the page loaded fully to recognize the ids. Once the page loaded, we could then grab the checkbox value and store it in memory to check it against a future change. Thus, we needed to add an eventListener to check for the change if it happened, and call that script to prompt the user for confirmation before proceeding with their selection.

```
// Created by Joshua Greenert on 9/8/2021
// This script will check if the universal checkbox has been updated
(selected or deselected)
// and prompt the user to ensure they want to make the change they will
be making.

var rememberOption;
var rememberCheck;

window.addEventListener("load", function() {

    // Get value from product (on or off) and store it
    var rememberOption = document.getElementById
("enovathemes_addons_universal_product").value;
    var rememberCheck = document.getElementById
("enovathemes_addons_universal_product").checked;

    // Set an on change function to catch the difference for when the
user makes the selection.
    document.getElementById("enovathemes_addons_universal_product").
addEventListener("change", checkUniversalChange);

});
```

```
function checkUniversalChange() {

    var universalValue = document.getElementById
("enovathemes_addons_universal_product").value;

    if(universalValue !== rememberOption){
        var message = "Are you sure you want to make this change?";

        if (confirm(message)) {
            // Let the user continue with their action.
        } else {
            // revert the option back to rememberOption state.
            if(rememberCheck === "false"){
                document.getElementById
('enovathemes_addons_universal_product').checked = false;
            }
            else{
                document.getElementById
('enovathemes_addons_universal_product').checked = true;
            }
        }
    }
}
```

Fishbowl

This page will host children pages that address issues encountered, and solutions to resolve them, with Fishbowl specifically.

- Fishbowl SQL Errors & Solutions
- How To Restore A Database Backup On Fishbowl Server

Fishbowl SQL Errors & Solutions

This page will host errors previously encountered with Fishbowl tasks and how to resolve them. Now that we host the database, we can make the changes directly without having to require assistance from a support team externally.

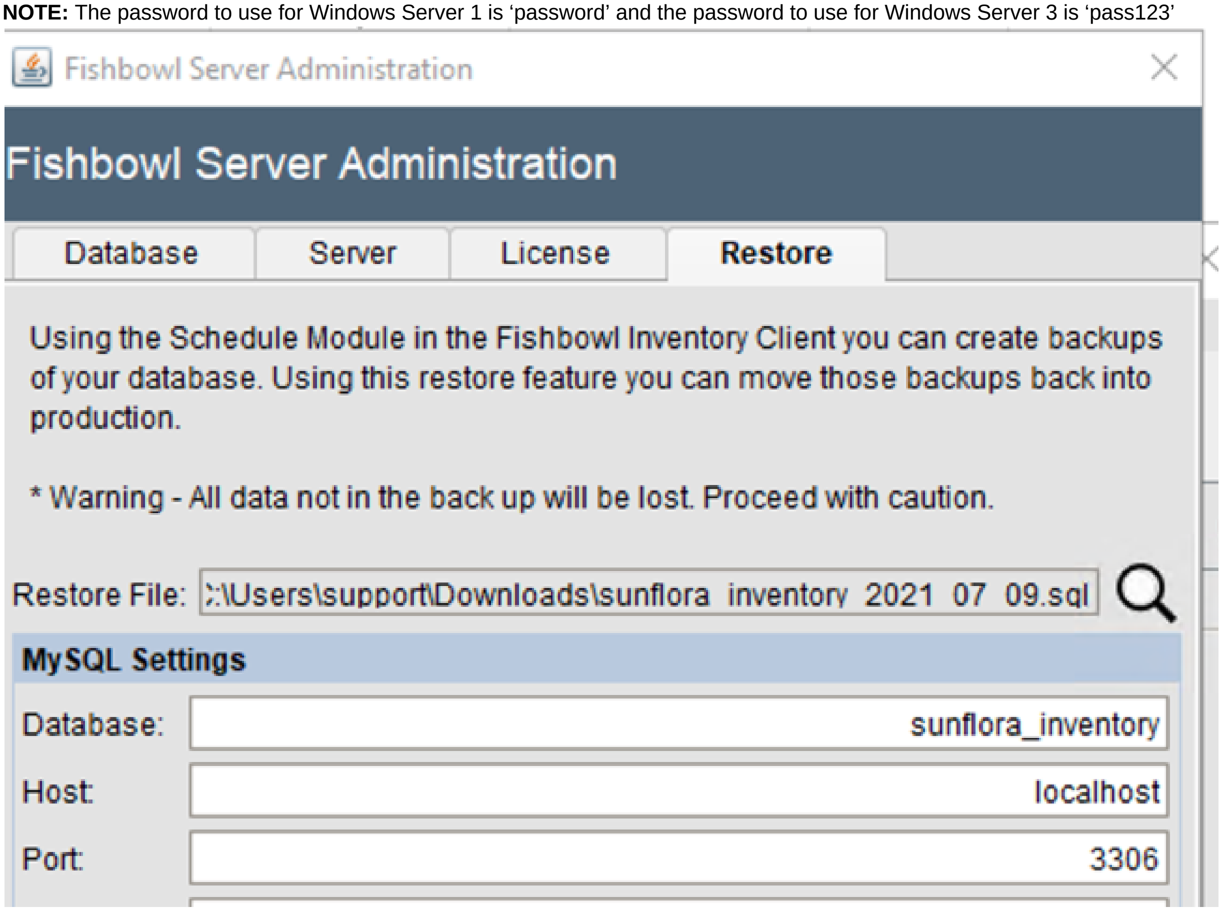
Errors and Solutions		
Error	**Solution**	**Notes**
`Failed to complete task. Request {method=POST, url=https://ssapi.shipstation.com/orders/createorders, tags={}} Response {protocol=http/1.1, code=400, message=Bad`	`select sunflora_inventory.so.num, sunflora_inventory.shipitem.*, sunflora_inventory.product.num from sunflora_inventory.shipitem join sunflora_inventory.so on`	**This note indicates that the shipitem table in the database has a value for the qtyshipped field that is less than 1. All values in this column should be 1 or greater, so decimals need to be updated to allow the shipstation task to run successfully.** **NOTE:** The same error can occur if the item is greater than a whole number, but still a decimal. For instance, 1.33 will throw and error for a 6 pack but was updated to 1 to correct.

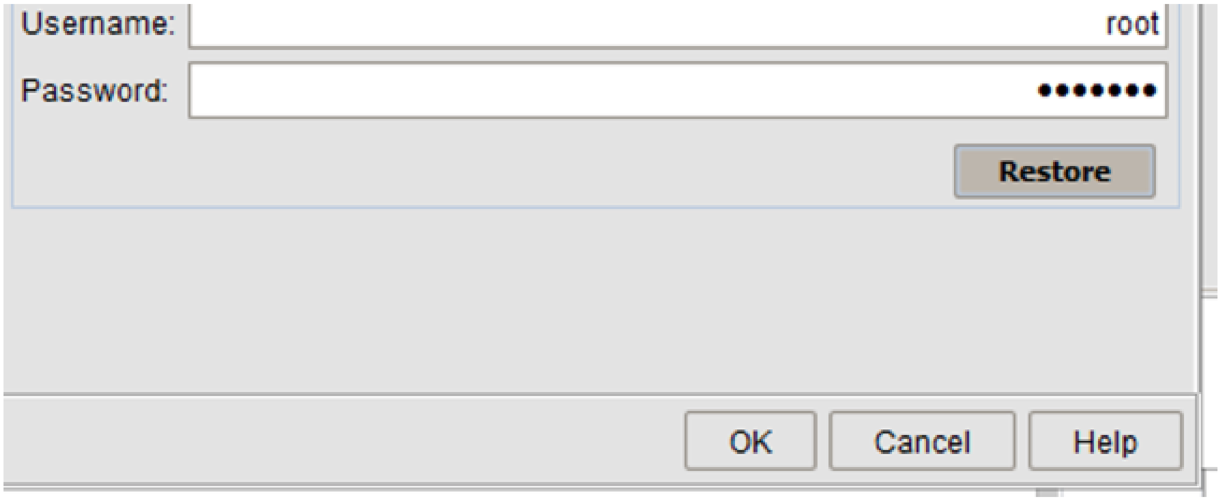
`Request, url=https://ssapi.shipstation.com/orders/createorders} {"Message":"The request is invalid.","ModelState":{"apiOrders[0].items[1].quantity":["Could not convert string to integer: 0.67. Path '[0].items[1].quantity', line 1, position 898."]}}`	`sunflora_inventory.so.id = sunflora_inventory.shipitem.orderid join sunflora_inventory.product on sunflora_inventory.shipitem.itemid = sunflora_inventory.product.id where ceil(qtyshipped) != qtyshipped`	
Unable to update database. Database version is newer than the one it is trying to update to.	Go to the newly recovered database and navigate to the databaseversion table. Inside this table, delete all references that aren't the version of the database it is actually at (at the time of this writing we are using 2019.11). `delete from sunflora_inventory.databaseversion where id != 1;`	**When recovering a database to a another windows server, records are stored in databaseversion table about what other version are available for the system to have. However, as an end result, the database attempts to use the other versions within the table to boot the server. Then the server refuses to boot up since the older version is not referenced as the only result in the table. Remove the other entries to allow the server to run as normal.**
Tried to decrement qtyCommitted by: 6 but qtyCommitted is: 4 on Tag num 45429 Line Number: 61	Update the qty for the Tag and shipitem tables to reflect the appropriate quantities. Cross reference this data using the tag, uom, and shipitem tables to ensure that the values are accurate, and updated if not.	**This issue was found on the shipstation import task as a failure. The reasoning as to why it occurred is still unclear. You must be cautious when making this update because if it isn't done correctly, when the task is ran again, a generic message will appear that states "data from the database does not match with what is stored". Be certain about the changes being made since the tables reference one another to gather data.**
ERROR 1273 (HY000) at line 6822: Unknown collation: 'utf8mb4_0900_ai_ci'	The Collate method used in the SQL file doesn't match with the version that it's being updated to. Open the SQL file in notepad/text editor and change `utf8mb4_0900_ai_ci`	**This issue occurs on an attempt to recover a database in a newer version of fishbowl from a file that comes from an older version of fishbowl. Speculation suggests that the likely result comes from the way the older version collated the data compared to how it does with its most recent version.**

	to `utf8mb4_general_ci`	

How To Restore A Database Backup On Fishbowl Server

This page will host the steps required to restore a database backup on the Fishbowl server. While the steps are pretty straight forward, always take caution when updating an database for live systems.

1. **Obtain a backup from either the virtual machine Windows Server 1 or from the Google Cloud Storage Bucket**
    a. **OPTION 1:** Google Cloud Storage Bucket
        i. https://console.cloud.google.com/storage/browser?orgonly=true&project=dang-test-space&supportedpurview=project&prefix=
        ii. Move to the last page where the date of the backup is the most recent.
    b. **OPTION 2:** From within Windows Server 1, navigate to Documents/Batch Files and run the executable GoogleCloudBucketTransfer.exe (need to check name)
2. **Download the backup to Google Drive to transport across the network**
3. **Enter the virtual machine instance where the server is hosted**
    a. RDP and Chrome Remote Desktop both are active for this selection
4. **Download the backup from Google Drive**
5. **Stop the Fishbowl server**
    a. The server must be fully stopped (with a red circle) in order for the next steps to be available
6. **Go to Tools > Options**
7. **Navigate to the tab called 'Restore'**
8. **Use the search icon to find the backup previously downloaded**
9. **Ensure that the MySQL Settings are similar to the following:**
    a. **NOTE:** The database name must match the database currently held within the system to refrain from creating additional database copies in the MySQL workbench
    b. **NOTE:** The password to use for Windows Server 1 is 'password' and the password to use for Windows Server 3 is 'pass123'
    c. 



10. **Press ‘Restore’**
11. **When the restoration process has completed, press ‘OK’ on the dialog box and press ‘OK’ again on the Fishbowl Server Administration window**
    a. If you close the window, instead of pressing okay, the connection may not be set to the proper database.
    b. If the recovery process is on a development server, make sure to disable all active tasks in the Fishbowl client
12. **You’re Done!**

How To Block Malicious Bots From Servers

This page will outline the steps required to block malicious bots from servers that have been reported. Additional bots that get reported can be added to this list and included in the Apache path using the same navigational guide.

1. **Log into the WHM server using the root account**
2. **In the search bar in WHM on the top-left, type “tools”**
3. **Select ModSecurity Tools**
4. **On the right-hand side, select “Rules List”**
5. **Select “Add Rule”**
6. **Copy this code into the Rule Text input box**

a.
```
SecRule REQUEST_HEADERS:User-Agent "@rx ^(?:MJ12bot|AhrefsBot|SemrushBo|Baiduspider|YandexBot|SeznamBot|DotBot|MauiBot)$" "msg:'Spiderbot blocked',phase:1,log,id:777777,t:none,block,status:403"
```

   i. The bots are blocked through the use of Regex and are blocked selectively by this piece

1.
```
?:MJ12bot|AhrefsBot|SemrushBo|Baiduspider|YandexBot|SeznamBot|DotBot|MauiBot
```

   ii. The id for the new Apache rule must be unique (i.e. id: 777777)
   iii. The status of bots that attempt to reach the server is set to forbidden (i.e. status:403)

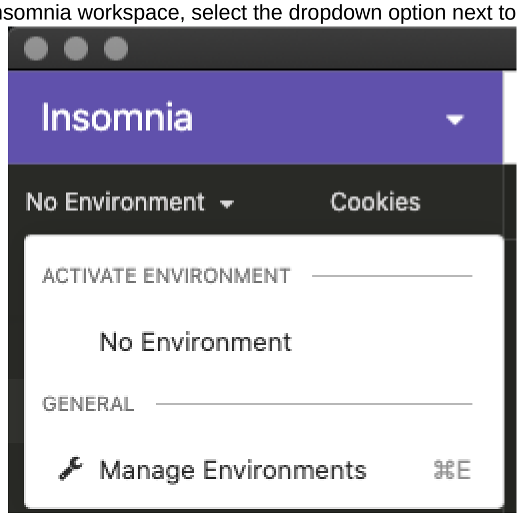
7. **Save the rule and redeploy Apache**
8. **Confirm that the new additions were successful**
    a. Navigate to the WHM terminal
    b. Enter the following curl command

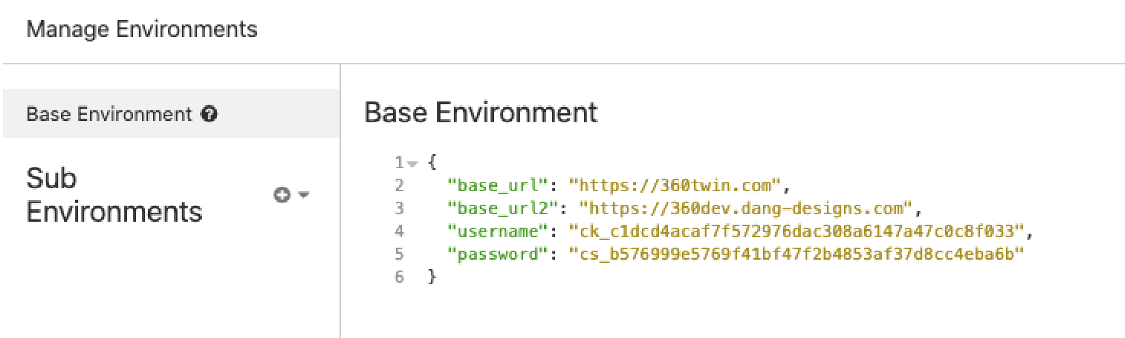
i.
```
curl http://localhost -H 'User-Agent: MauiBot' -I
```

1. This command should return a similar prompt to this

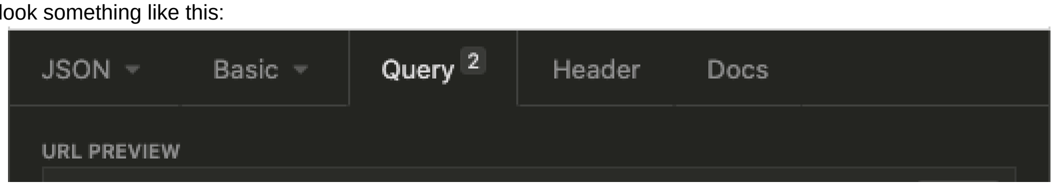
a. ```
HTTP/1.1 403 Forbidden
Date: Tue, 14 Dec 2021 15:50:54 GMT
Server: Apache
X-XSS-Protection: 1; mode=block
Referrer-Policy: no-referrer-when-downgrade
X-Permitted-Cross-Domain-Policies: none
Upgrade: h2,h2c
Connection: Upgrade
Accept-Ranges: bytes
Cache-Control: no-cache, no-store, must-
revalidate
Pragma: no-cache
Expires: 0
Content-Type: text/html
```

2. All of the bad bots can be checked in a similar fashion, but this one was just for demonstration

9. **You're Done!**

## How To Connect A Fishbowl Client Remotely - DANG designs Users

**This page will provide guidance on how to connect a remote Fishbowl client to the Windows1 and Windows2 servers hosted on Google Cloud Platform.**

1. **Download Fishbowl Client on machine -**
   a. **Download the 2019.11 - https://www.fishbowlinventory.com/support/updates/download**

      Fishbowl 2019 (Former Version - 2019.11)
      - Windows Version (64 bit)
      - Windows Version (32 bit) (**Server not supported**)
      - Mac Version (Limited Support, Must have 10.7.5 or newer)
      - Linux Version (Limited Support, Tested on Ubuntu)

   b. For Mac, the client will need to be added in the Root account -
      i. https://support.apple.com/en-us/HT204012
      ii. Download using an admin user, then start the installation
         1. you will get a warning that the app is not from a from
2. **Run the Installer**
   a. Company Information
      i. Select the
         1. Country - US
         2. Business Type (doesn't effect the Client Only option)- Manufacturer, Distributor, Retail
         3. Industry - Cannabis
   b. **Client Only:** This option will install the Client version of Fishbowl
      i. Do not install the "Server with Client"
3. **Before connecting to the instance, make sure that the fishbowl has three additional jar files in the plugins folder**
   a. These files are stored under DANG-work > CBDStore > Fishbowl > JAR Files
      i. Advanced-shipstation
      ii. Fishbowl-go
      iii. Magento-cart
4. **Open the Fishbowl client and set the connection criteria**
   a. Username and password are the same as the live credentials
      i. **Username:** dangdesigns
      ii. **Password:**
   b. If you are using the load balance to allow the health of the machine determine the machine used, then use the load balancer. Otherwise, use the static Windows IP Addresses below:
      i. **Load Balancer:** 34.74.50.49 - Most users
      ii. **Windows Server 1:** 35.231.247.69 - admin only
      iii. **Windows Server 2:** 35.243.225.145 - sandbox
   c. Ensure that the Server Port field is set for 28192
5. **Press 'Login' to connect to the instance**
   a. Should any issues arise, and you are using a new machine that does not have the IP address white listed, either reach out to joshg@dang-designs.com or manually update the Fishbowl Egress and Fishbowl Ingress firewall rules in Google Cloud Platform's Cloud Web Server project.
6. **You're Done!**

## How To Connect To API End Point Using Insomnia
```

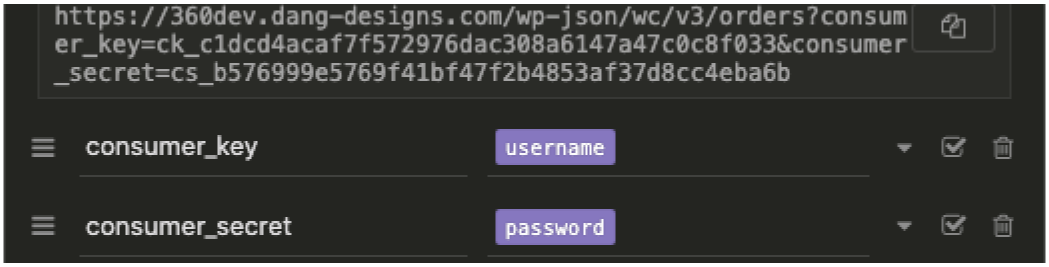
This page covers how to connect to a WordPress API endpoint. In this particular instance, I will be covering how to connect to the Woocommerce API endpoint through the use of the developer application called Insomnia. While this shouldn't be a necessarily difficult process, there are some caveats to keep in mind when things aren't working as intended. Some steps defined as we go along are optional and simply save time in the long run; as such, they will be noted with a status like so: OPTIONAL

1. **Install Insomnia Core**
2. **On WordPress, or the application/CMS required, navigate to the settings and create an API key**
    a. For Woocommerce, this is can be found by going to the Woocommerce plugin > Advanced > REST API (blue button) selection
    b. This should provide a consumer_key and consumer_secret
        i. After the page has been left for Woocommerce, this information will no longer be able to be obtained.
        ii. *This is fairly common for most API/Oauth keys obtained*
    c. Record the consumer key and secret in a place where they can be used later.
3. **Create a GET request with an appropriate name for the endpoint data request**
4. **Set the environment variables for the base URL to access data through and the key and secret** OPTIONAL
    a. In the Insomnia workspace, select the dropdown option next to the active environment; then, select manage environments
        i.


    b. The system uses a sub environment that JSON data can be added within. Enter the fields that you intend to use constantly as JSON pieces.
        i.


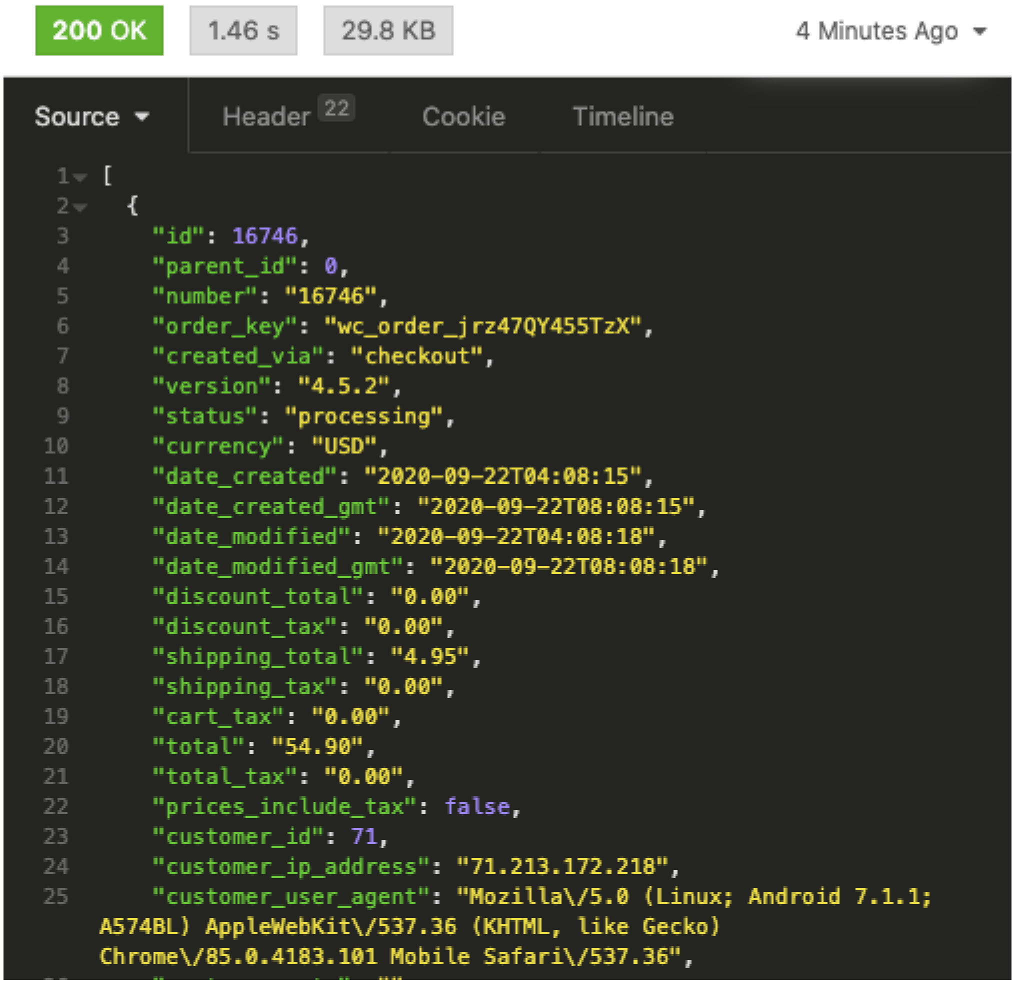
    c. When uses these in the data fields, use curly braces to register them as a variable.
        i. EXAMPLE: {{ base_url}} would call the variable above as "https://360twin.com"
    d. Press the 'Done' button on the bottom right of the screen when completed.
5. **Enter the URL into the GET field for the website to get the API endpoint from and add the API path to get data from the website /application**
    a. Should look something like this:
        i.


        ii. *In this instance, the environment variable is used which is why base_url2 is listed rather than "https://360dev.dang-designs.com"*
6. **Ignore the Body, Basic, Header, and Docs tabs listed under the GET field; we only need to add the consumer key and secret in the query fields**
    a. In the 'New name' fields, enter the consumer key and secret names. Enter the values directly next to them.
    b. Should look something like this:
        i.




    ii. Once again, keep in mind that the purple letter is the result of using environment variables.

7. **Press the 'Send' button and ensure that you are getting the output you desire.**
    a. For instance, for mine, I got this:
    b.



        i. *It's worth noting that all of the data I received from this request was not posted for security reasons as it provides primarily customer data.*

    c. If the output was not what you desired, check the path for the URL and see what details are specified in the application's/plugin's documentation.

8. **You're Done!**

References	
Source Link	**Comments**
https://www.youtube.com/watch?v=lPFOEufD6mQ	YouTube video where user explains how to use an alternative API with Insomnia to send email requests.
https://github.com/woocommerce/woocommerce/wiki/Getting-started-with-the-REST-API	GitHub Woocommerce troubleshooting guide to understand where issues are. (This piece helped me solve it for the first time)
https://sgwebpartners.com/how-to-use-woocommerce-api/	Simple walkthrough on getting Woocommerce API key and secret.
https://robotninja.com/blog/fix-common-woocommerce-rest-api-issues/	Document that discusses issues with API usage for Woocommerce; the correction to the 360 twin site was found here.

How To Correct Anacron Job Email Issues AND Other Linux Issues

This page covers the issues experienced from linux virtual machines and includes the email messages (snippets) that indicate an issue or item that needs to be addressed. While not all problems will have a direct solution, most do and will be documented here until solved; after which, this document will be updated to reflect the appropriate action to take to properly address the issue/item.

Anacron Job Messages		
Message	**Meaning**	**Solution**

`/etc/cron.daily /logrotate:`
 	
`Redirecting to /bin/systemctl restart imunify360-pam.service`
	Linux machines operate on a system of repositories that allow the system to check what files/folders to use. These structures update constantly and require the repos to be update to function accurately. So far as it has been seen, there are no egregious issues when not making these updates, but the emails will take place until they do.
	
	NOTE: The information preceding the etc/cron.daily/logrotate can always be different. The main detail that alerts the user to this issue is the word “redirecting”.
		Go to the virtual machines terminal and perform both of the following commands separately:
		
		`yum clean metadata`
		
		^ This command cleans the stale references and points the system in the direction of the proper locations
		
		`yum clean all`
		
		^ This one covers the files/folders remaining.

`Constant subroutine unbound_version redefined at /usr/local/cpanel/bin/autossl_check line 4294967295.`	This one does not impact the autoSSL functionality but does produce an email fairly consistently to remind the user about the issue.	No solution found at this time. The CPanel team have acknowledged the issue and plan to address it with one of the next releases.
Nobody:Nobody as file owner	The root of a magento system is supposed to create folders and assign a user to be the owner. For unknown reasons, capitalcityshoes doesn’t follow this criteria when setting new templates within media /customoptions/etc. This command on the right sets the user as the owner of the file directory to resolve the issue.	`sudo chown -R shopccshoes:shopccshoes [name_of_directory]`
Need to remove files/directorys (non-empty)	This is a command to remove full files and directories without the hassle of verifying each and every action.	`rm -rf dirname`
Need more information about PID	This is a command to see more information about the current running	`ps -Flww -p [PID]`

	process to find out where it is coming from along with a breakdown about specific pieces within it.	

How To Correct Web.dev Listed Issues

This page will show users how to correct various issues from each category from the web.dev site (provided information from lighthouse). There are four categories that will be broken down with solutions for each issue expected. While some solutions are all-encompassing, there are some issues to look out for which will also be addressed either in the 'Errors' section for the indicator or in the document referenced for the process.

Performance			
Metric	**Verbiage**	**Solution**	**Errors**
Opportunities	**Efficiently encode images**		
Metrics	**Speed Index**	Implement a CDN to ensure that the site is pulling from local servers. This one differs between sites but the general concepts deal with configuration. As this one is as expansive as it is, I'll link to another page to go through improvement options. Setting Up A CDN For Joomla Site	If the theme for the Joomla site is anything besides YOOtheme, ensure that the compression element within the template is not using further compression and is set to either 'minify' or 'none'

Accessibility			
Metric	**Verbiage**	**Solution**	**Errors**
Navigation	**Heading elements are not in a sequentially-descending order**	For Joomla, if the item is within a module, the title is added as the **H3/H2**, etc. object. One possible solution is to remove the heading from the module and include a new list item for the title element.	
Names and labels	**Links do not have a discernible name**	This essentially means that the item needs to have a label or **alt** text added to account for people who use screen readers. For select items, such as to-top buttons, it is also optional to use a title element (title="To Top") which will resolve this for users as well. If neither of those resolve the error, use the one suggested on the right.	In Joomla, if the site uses Yootheme, there is a custom code section to enter this or something similar. *This references the UIkit script listed below `el.setAttribute('aria-label', 'Facebook Link');`

Best Practices			
Metric	**Verbiage**	**Solution**	**Errors**

Trust and Safety	**Links to cross-origin destinations are unsafe**	Add `rel="noopener"` or `rel="noreferrer"` to any external links to improve performance and prevent security vulnerabilities.
		
		Joomla: Log into admin, go to extensions > templates, select default Yootheme template, customize, go to settings > custom code > add the code listed below
		
		`<script>`
		`UIkit.util.ready(function() {`
		`UIkit.util.$$('a').forEach(function (el) {`
		`  el.setAttribute('rel', 'noopener');`
			If Yootheme is not set up for the Joomla site, make sure to check all data by the database and local copy of the files to update the link (It may be hard-coded inside of the site). If it the link is hard-coded into a file or site, check the database files using the search or pull the files down to search using VS Code.

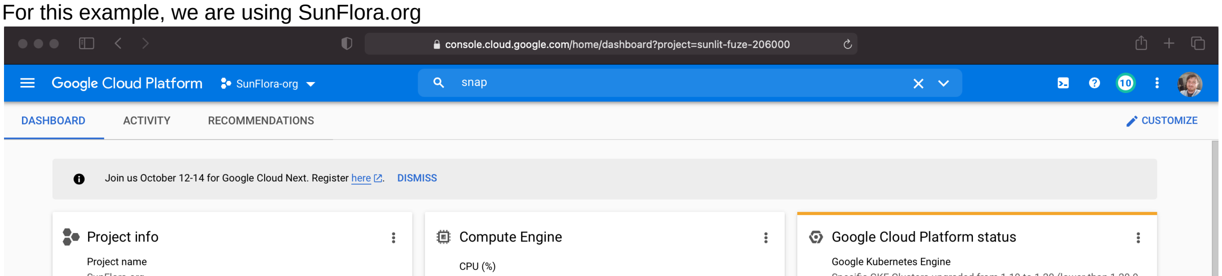
		}) }); </script>	
Trust and Safety	**Includes front-end JavaScript libraries with known security vulnerabilities**	Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers.	
User Experience	**Serves images with low resolution**		
General	**Browser errors were logged to the console**	Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. These items are specifically related to some portion of script or URL errors. Essentially, find the error and the fix should be easy to update.	
ARIA	**[role] values are not valid**	ARIA roles are defined to assist users of screen readers to better navigate web pages. Mainly, this element is included when typical fields supported by most HTML is not present. Anytime a role is found to be insufficient, it must be replace by a valid role to suit the purpose of the user's needs while associating with a screen reader's navigable library. Here is a link where these roles are defined: https://www.w3.org/WAI/PF/aria/roles#abstract_roles_header	

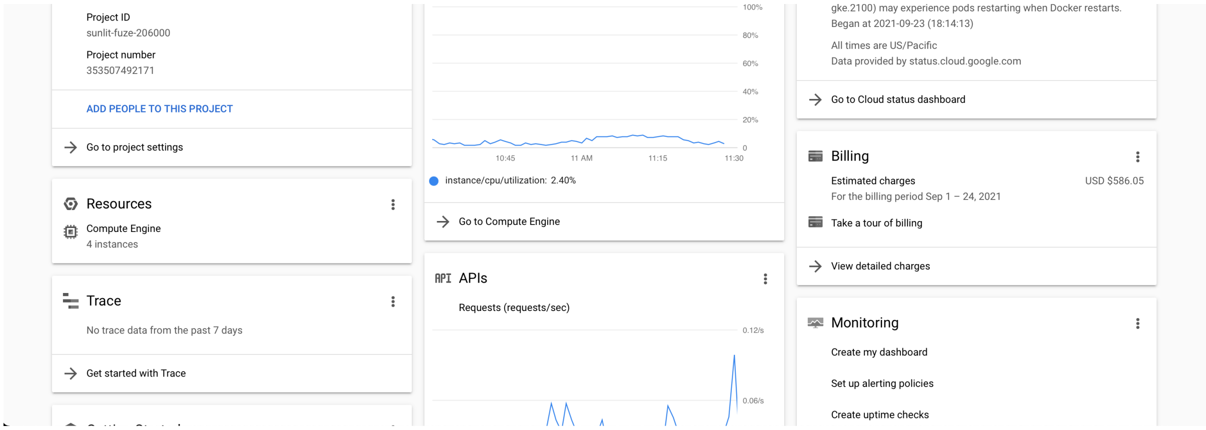
SEO			
Metric	**Verbiage**	**Solution**	**Errors**
Content Best Practices	**Document does not have a meta description**	**Joomla:** Log into admin, go to global configuration, update meta description **WordPress:** Log into admin WordPress site, make sure Yoast SEO plugin is downloaded, go to pages, go to home page, update SEO metadescription towards bottom of page	**404 Error on update from WordPress:** if you get this error, you will need to deactivate the plugin that is causing the issue before updating the Yoast information (could be WPBakery Builder or SiteOrigins).
Content Best Practices	**Links do not have descriptive text**		
Crawling and Indexing	**Links are not crawlable**		

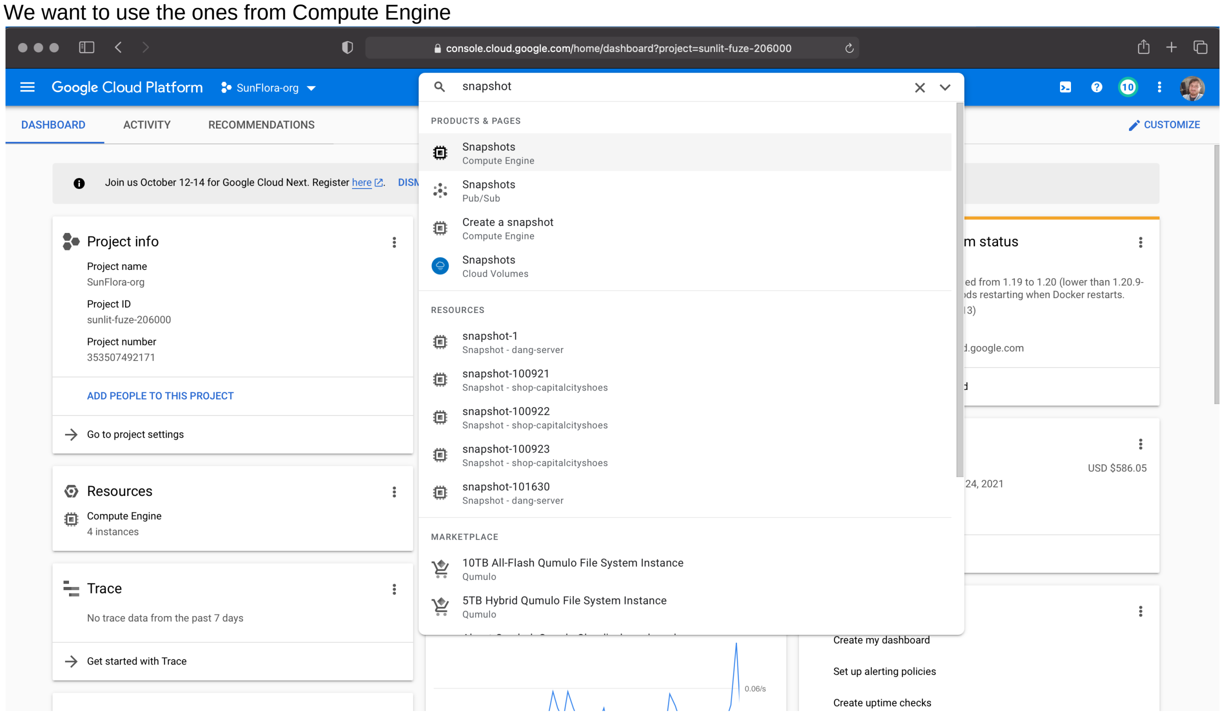
References	
Link	**Description**
https://yootheme.com/support/question/139234	Contains script for cross-origin update
https://stackoverflow.com/questions/32378881/adding-rel-external-to-every-link	Stackoverflow link where cross-origin script was initially found

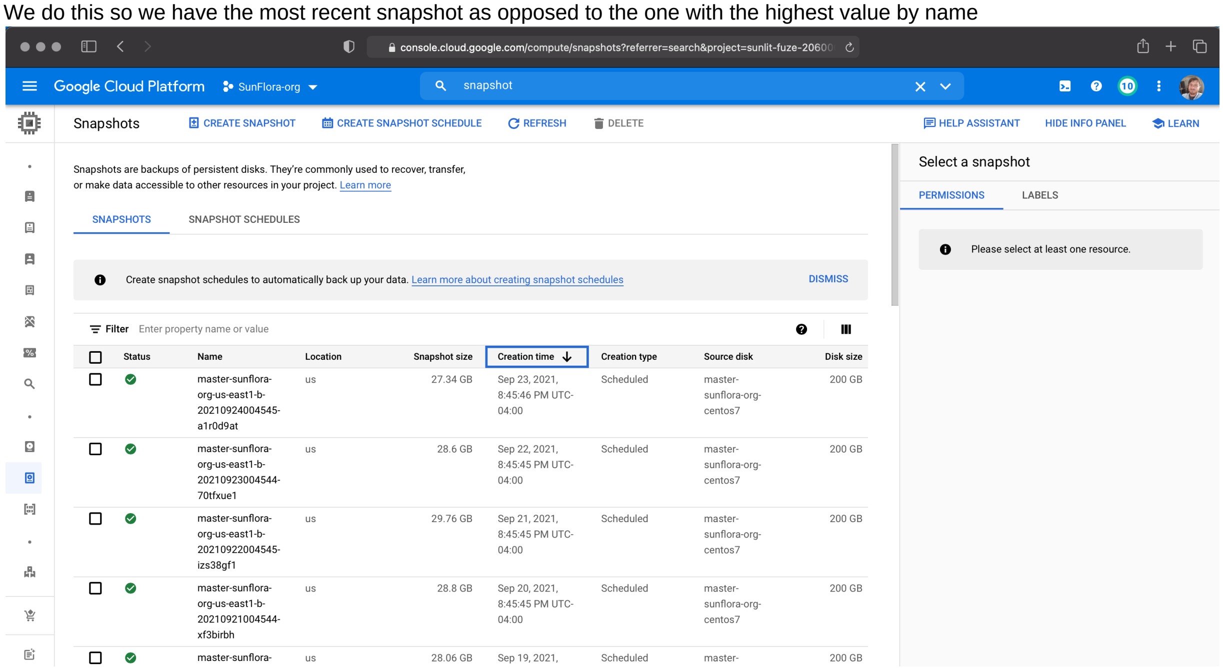
How To Create An Instance From A Snapshot (GCP)

This page will document the steps required to restore a backup from a snapshot on GCP. This example specifically will use Sunflora's CentOS VM. It's worth noting that the specs for the virtual machine are not imported over when you create the instance from the snapshot. Those details will be listed below as well for example used.

1. **Go to console.cloud.google.com**
2. **Sign into your Google Account (or the Google Account associated with the server needed)**
3. **Set the project to the one to get the daily-created snapshot from**
   a. For this example, we are using SunFlora.org
   b. 

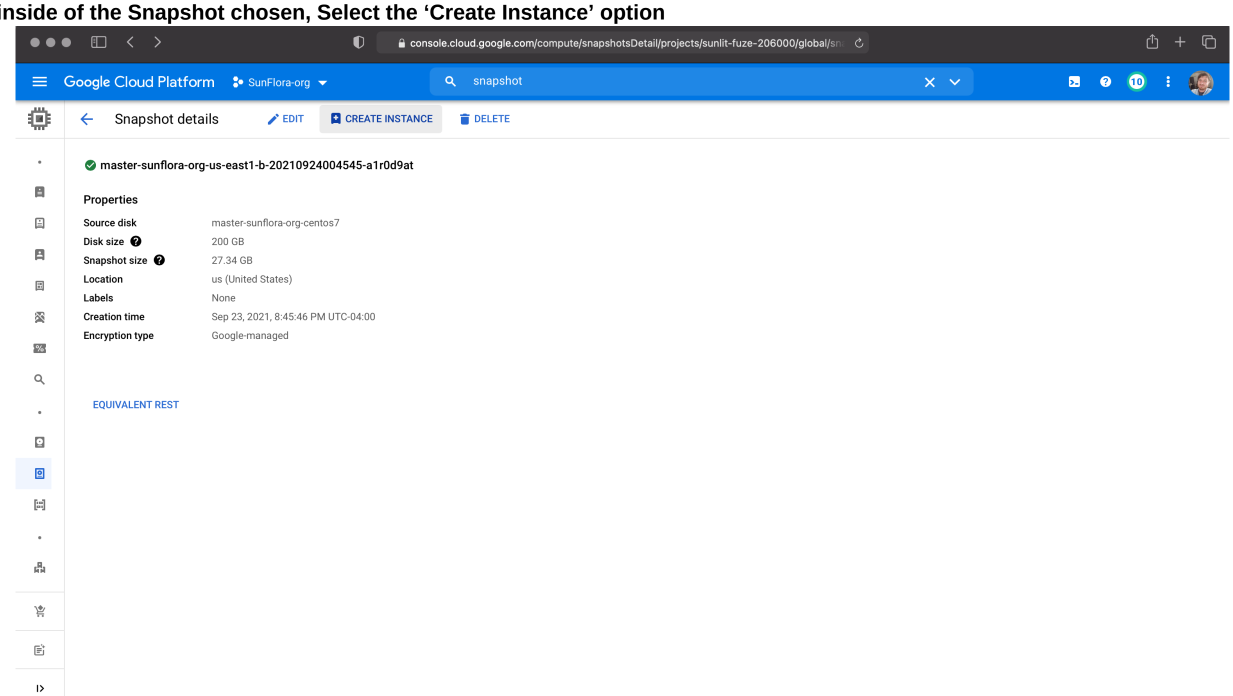


4. **In the search bar in the top-middle of the screen, type the word "snapshot"**
   a. We want to use the ones from Compute Engine
   b. 

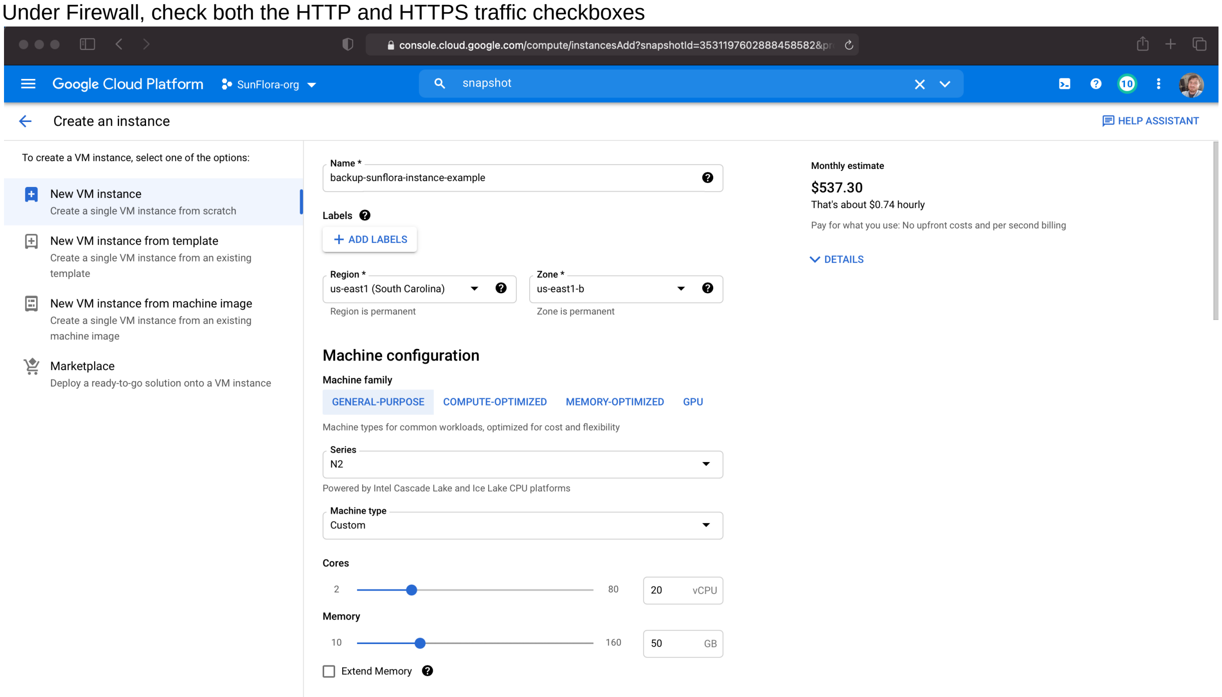
5. **On the Snapshots main page, sort the objects by creation time**
   a. We do this so we have the most recent snapshot as opposed to the one with the highest value by name
   b. 

   c. At the time of writing this document, the date was 9/24/2021, so the most recent should be the day before (9/23/2021).

6. **Click the link on the most recent snapshot by selecting the name**
   a. i.e. master-sunflora-org-us-east1-b-20210924004545-a10d9at in our screenshot above
7. **While inside of the Snapshot chosen, Select the 'Create Instance' option**
   a.



   b. This creates the instance with the data stored within the virtual machine; however, it doesn't capture the details about how the machine should be set up.
8. **Set the fields of the new instance to match those of the existing instance**
   a. Set the name of the instance
      i. Make sure to use dashes (-) instead of spaces or underscores
   b. Select the region as us-east1
   c. Set the zone for us-east-b
   d. Set the machine to be a N2 Custom
      i. Change the Cores option to 20 vCPU
      ii. Change the Memory to 50 GB
   e. Under Firewall, check both the HTTP and HTTPS traffic checkboxes
   f.



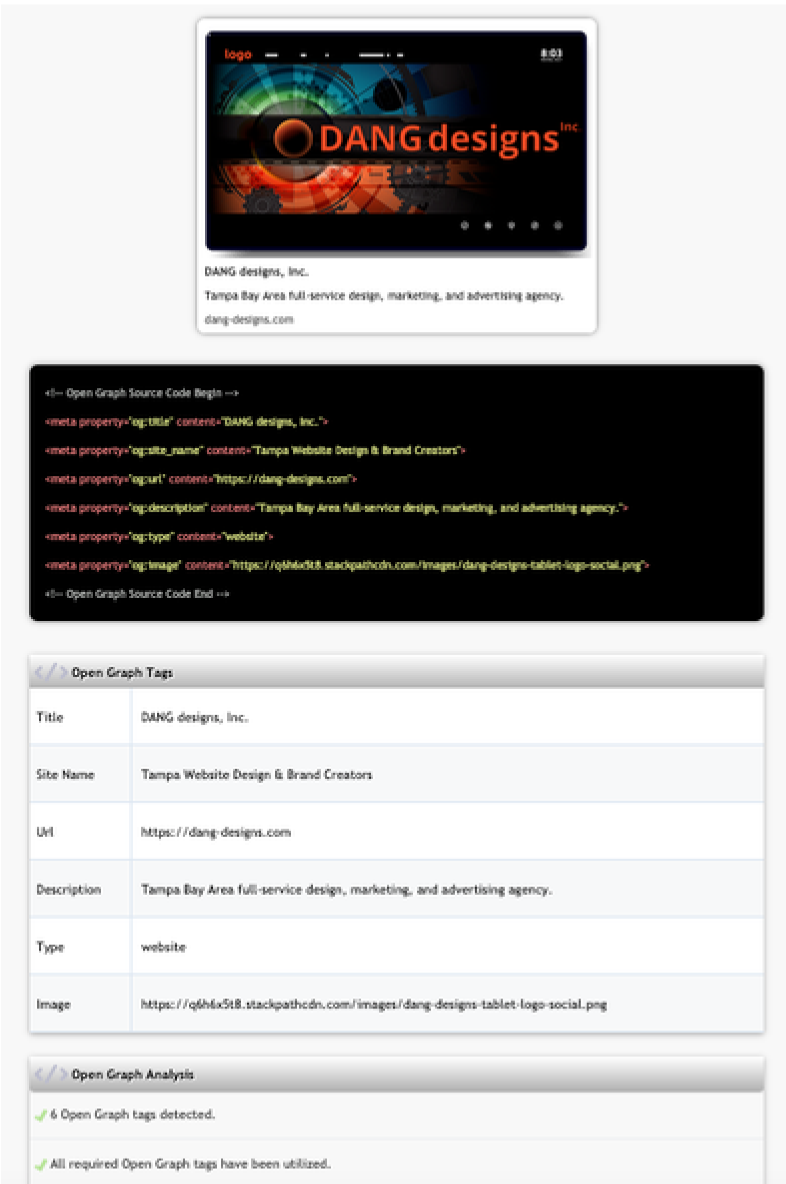
   g. Press Create
9. **You're Done!**

How to Create Social Media Open Graphs For Any Website Using Joomla.

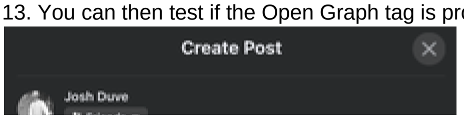
http://opengraphtools.com/

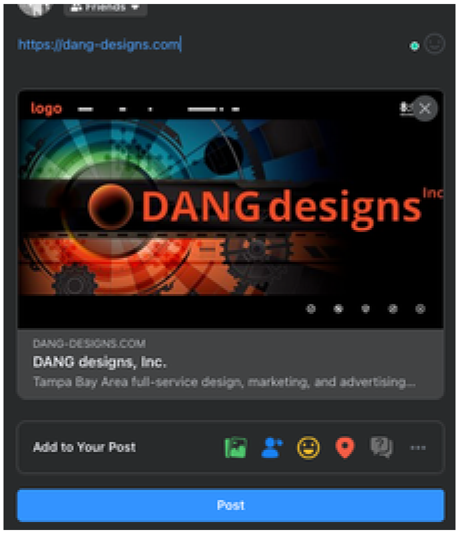
1. Using the above website tool, enter the website URL to view the current open graph tags.
2. If changes need to be made, proceed back to the tools home page and choose “Open Graph Generator”
3. Enter all of the info into the form on the left.
4. Click the “Copy Your Code” button on the right when ready.
5. Open the website backend and open the “Google Structured Data” Component.
6. Goto the Configuration > Advanced tab.
7. At the bottom find the “Custom Code” area.
8. Paste the code you just created into the Custom Code box.
9. Click Save.
10. Now go back to the opengraphtools.com site and run the query again for that website.
11. It should now be populated with the new info you added.
12. Make sure all 6 areas have proper info and get a green checkmark next to it ✅

See the image below for example:



13. You can then test if the Open Graph tag is properly working by posting a link to the website on Facebook.





Note: **It can take some time to populate on socials. To speed up the process, you can go to the "Facebook Debugger" website and scrape the URL, then see if it populates on Facebook. Remember to also clear the website cache on the backend.**

https://developers.facebook.com/tools/debug/

How To Perform Better Google Searches

While there are several examples for when this will be useful, the information below covers an example that is prominently noticed. Essentially, the practices are all-encompassing.

Example For When To Use This:

- When a store says someone got their personal number from Google… Use the following steps on how to find this number. We want to make sure that our system does not have the wrong number and wants to help get it off anywhere it could live. Ex. If the number is on five-stars you would contact Adam to remove the number.

Advanced Search For Phone Numbers:

- Using quotes will narrow searches and deliver better results.
- Search "(000) 000-0000"
- Search "000-000-0000"
- If you cannot find the number still try searching "000-0000"
  - This will pull more results

Advanced Google Searching		
Keyword/Function	**Example**	**Description**
NOT	social NOT security	Finds items that contain *social*, but not *security*.
OR	social OR security	Finds items that contain *social* or *security*.
Quotation marks	"social security"	Finds items that contain the exact phrase *social security*.
Parentheses	(social security)	Finds items that contain *social* and *security* in any order.

References
https://docs.microsoft.com/en-us/windows/desktop/lwef/-search-2x-wds-aqsreference
https://www.lifehack.org/articles/technology/20-tips-use-google-search-efficiently.html

How To Set Up Enhanced Ecommerce Conversion Tracking (GA + GTM ONLY)

This page will provide the steps used to obtain ecommerce data flow on Google Analytics using only the Google Tag Manager and the Google Analytics information. In the instance where this happens again and needs to be duplicated, these steps should outline the process rather swiftly for execution.

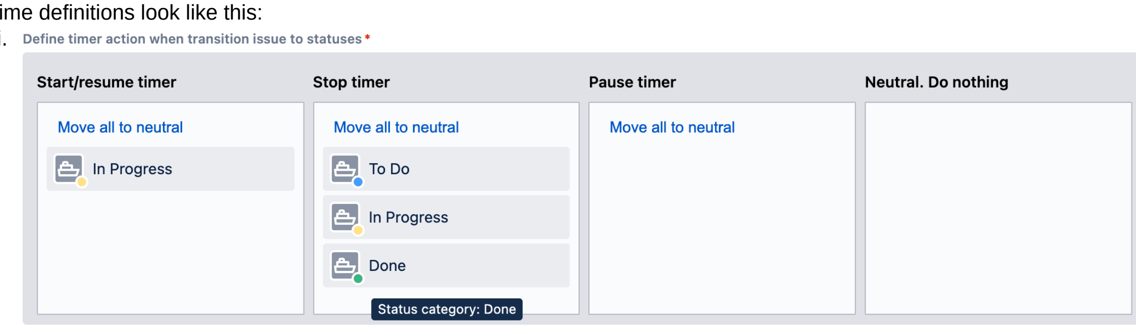
1. **Set up a Google Analytics variable that holds the value for your google analytics**
   a. On Sunflora GTM, this is labeled as Sunflora - Google Analytics
   b. Under Ecommerce (inside GA variable) enable enhanced ecommerce features and use the data layer

2. **Create a trigger that is set to page view with the URL that contains the success page (or order completed page)**
    a. On Sunflora GTM, this is labeled as *Sunflora - Purchase Trigger*
3. **Create a Tag with the Universal GA**
    a. On Sunflora GTM, this is labeled Sunflora - Purchase Conversion
    b. Use the Track Type of 'Transaction'
    c. Use the GA variable, enable overriding settings
    d. Assign the trigger Sunflora - Purchase Trigger
4. **Publish and wait to see the information on the following day.**
5. **You're Done!**

Link from Google that describes the process for set up:
https://support.google.com/tagmanager/answer/6107169

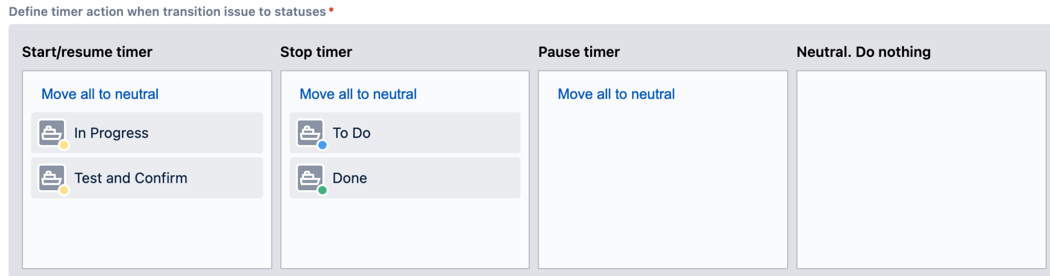
How To Set Up New Jira Projects

This page will discuss the use of what templates and what changes to make to ensure that time is tracked, fields are available, and data remains congruent from project to project.

1. **Navigate to Jira Cloud web app**
    a. The Desktop application doesn't support most of the configuration options that would need to be performed.
2. **Go to the Projects section**
3. **Create the new project**
    a. Select 'Create project'
    b. Set the project name
        i. Typically this is set as the name of the company we are working on or the name of the website
    c. Set the project template
        i. Typically we use the Kanban board template since it includes the boards for Todo, In Progress, and Done
    d. Press 'Create'
4. **Add a new column to the board for Test and confirm**
    a. Navigate on the left panel to 'Board'
    b. Press the Plus sign in the center of screen to add a column
    c. Name it 'Test and Confirm'
    d. Drag that column to be on the left of 'Done'
5. **Match the issue types to be similar to other projects**
    a. Use the left navigation panel to go into 'Project Settings'
    b. Navigate to 'Issue Types' on the navigation panel
    c. Add the fields to match all other projects
        i. Under description fields, add the Environement field to the 'Description fields'
            1. Do this by dragging the Environment field option on the right under the header 'Description fields' in the center
            2. Expand the Environment field to add in particular text to be generated on each task created
            3. Add the following within the environment field
                a. **File Path:**
                b. **Line of Changes:**
        ii. Under Context fields, add the Time Tracking field
        iii. Press 'Save Changes'
    d. Perform the steps listed in 4.C for Epics and Subtasks as well
6. **Correct the Automated Log Work settings for the project**
    a. Navigate back to the general project settings
    b. Select 'Apps'
    c. Select 'Automated Log Work'
        i. Automated Log Work is a Jira plugin/extension that was added to our Jira Software
            1. Documentation can be found here: https://marketplace.atlassian.com/apps/1211125/automated-log-work-for-jira?tab=support&hosting=cloud
    d. Ensure that Time Tracking Mode is set to Automatic
    e. Enable 'Show 'Start/Stop Timer' buttons in issue view'
    f. Start timer for user(s) > set to Assignee or Current User
    g. If the time definitions look like this:
        i. 

        ii. Set all items to neutral to ensure changes are recorded correctly. At the moment, these above settings mean that the timer won't start on it's own when the user places a task 'In Progress'
        iii. Drag 'In Progress' and 'Test and Confirm' to 'Start/resume Timer'
        iv. Drag 'To Do' and 'Done' to 'Stop Timer'

v. The end result should look something like this:

1. 

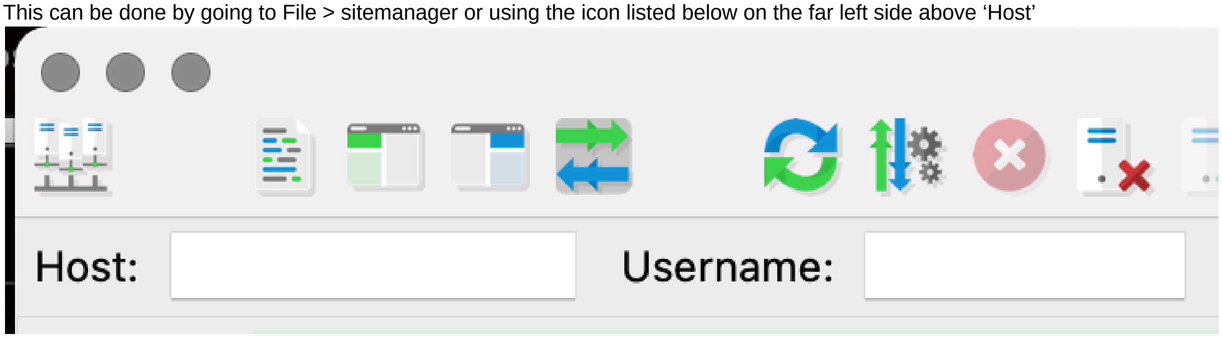
h. Ensure the Single timer options are checked for Singer Timer for user ONLY

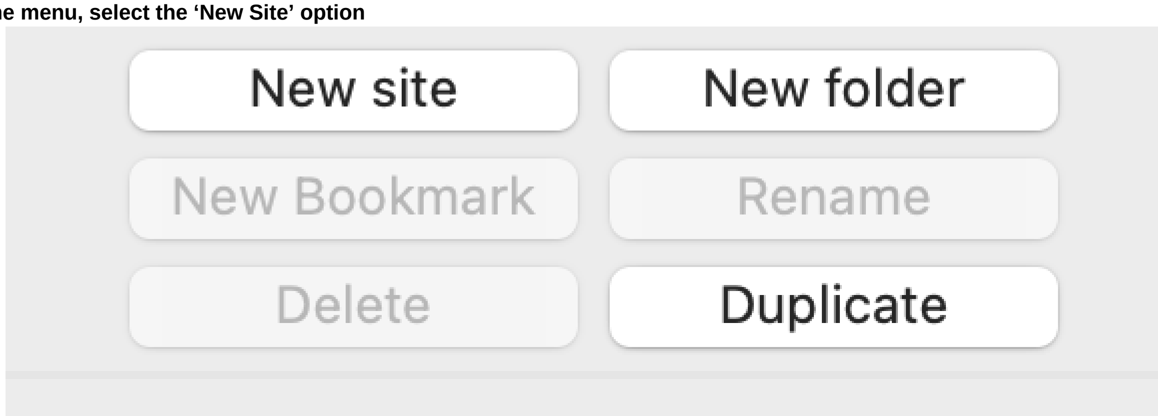
i. Press Save

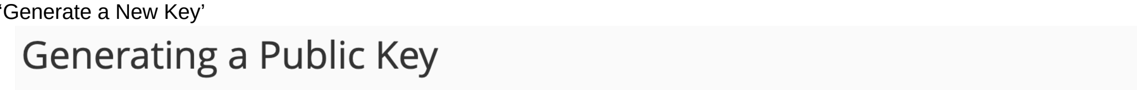
7. **Add the appropriate people to the project**
   a. Navigate back to project settings
   b. Select ‘Access’
   c. Select the big ‘Add people’ button on the top right of the page
   d. Add the appropriate people to the project as either members, viewers, or administrators
   e. Press Add for each one
8. **You’re Done!**

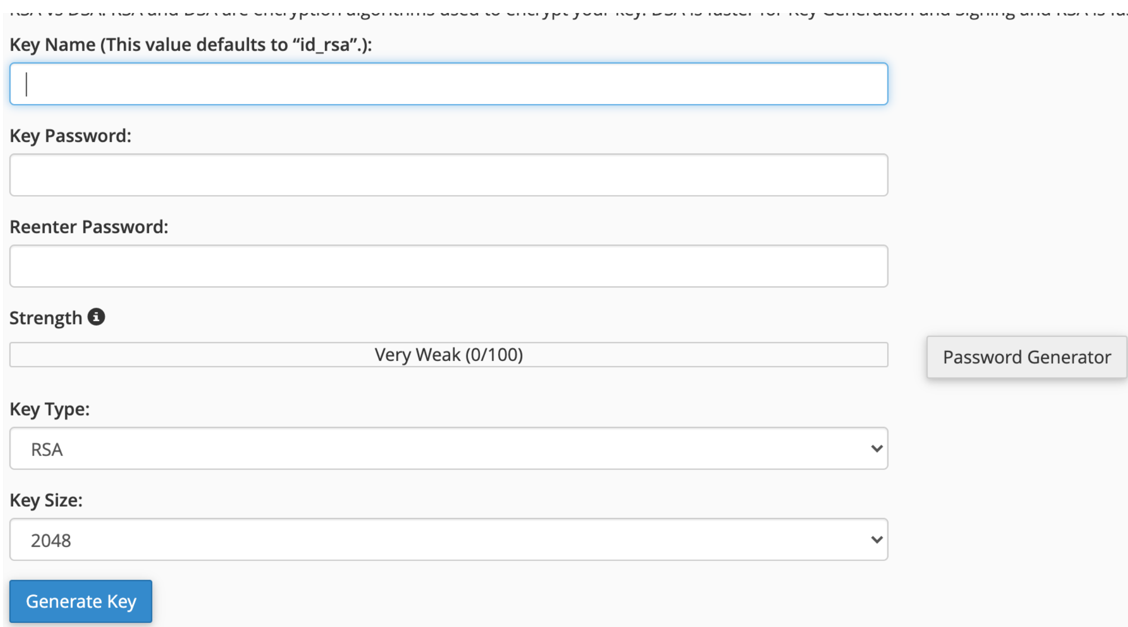
How To Set Up SSH Through FileZilla

This document will discuss and cover the steps required to establish a secure connection to server files through the use of FileZilla. In order for a user to set up FileZilla access properly, the user must have access, or another user that sets it up must have access, to the Cpanel account along with the details for the Cpanel account such as IP address, Cpanel username, Cpanel SSH access, and GitHub repo access for the respective domain.

1. **Ensure that FileZilla is installed**
   a. If FileZilla is not installed, use the link below
      i. https://filezilla-project.org/
2. **Open FileZilla and access the site manager**
   a. This can be done by going to File > sitemanager or using the icon listed below on the far left side above ‘Host’
   b. 

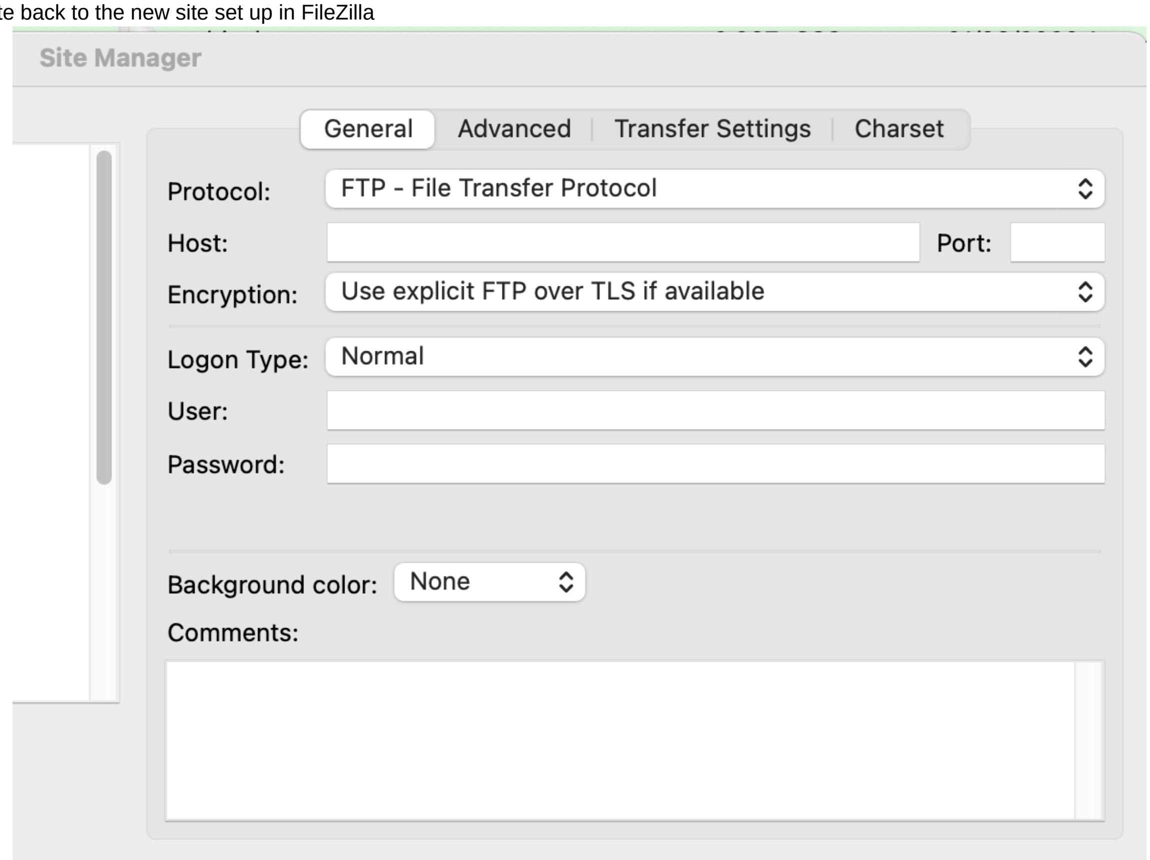
3. **From the menu, select the ‘New Site’ option**
   a. 

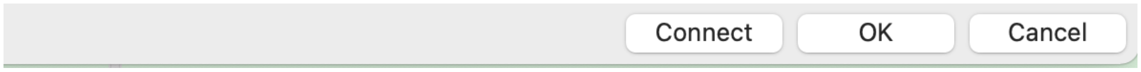
4. **Set the name as the [NAME_OF_SITE] - live or dev depending on the setup**
5. **Navigate over to the Cpanel used to set up SSH**
   a. In the search bar, type ‘SSH’
   b. Select SSH Access
   c. Select ‘Manage SSH Keys’
   d. Select ‘Generate a New Key’
      i. 



e. Set the key name as something unique that is distinguished enough to know who the owner is
f. Set the Key password fields or use the Password Generator to create the password for the SSH key
g. Press 'Generate Key'
h. Use the 'Go Back' option to return to the list of keys
i. Find the key, listed under Public Keys and select the 'Manage' option
j. Select the 'Authorize' option
k. Use the 'Go Back' option to return to the list of keys

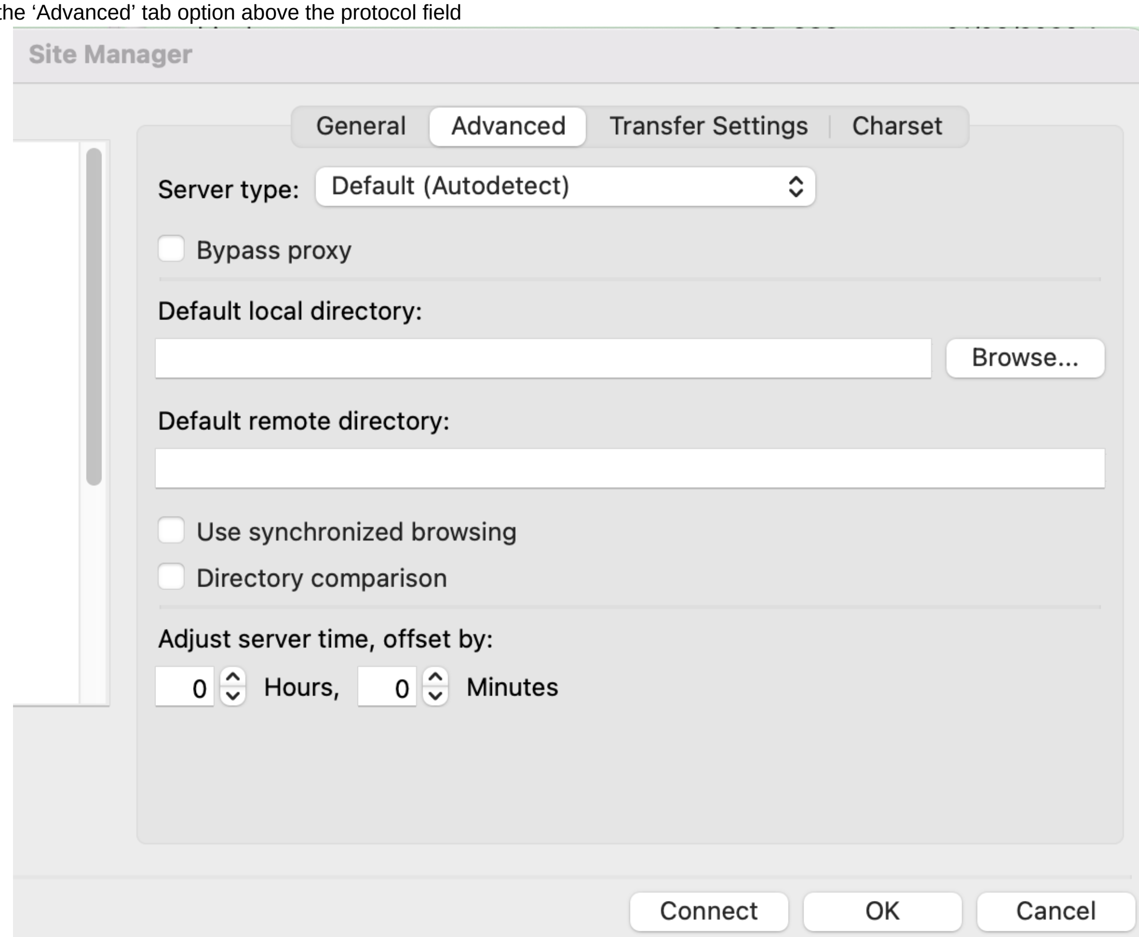
6. **Download the PPK file to use for FileZilla**
   a. Inside of the SSH Access screen, under Private Keys, select the 'View/Download' option
   b. Scroll to the bottom and enter the passphrase to unlock the key
      i. This converts the key to a PPK format to download
   c. Press 'Download Key'
   d. Save the file to secure location that you will remember
7. **Set up the site in FileZilla**
   a. Navigate back to the new site set up in FileZilla
      i.





b. Set the protocol to 'SFTP - SSH File Transfer Protocol'
c. Set the Host field to the shared IP address found inside of Cpanel under the Primary Domain field
d. Set the Logon type to Key File
e. Set the user as the Cpanel username
    i. This can be found under the home directory after the word 'home' on the Cpanel overview window
f. Use the browse option to set the PPK file saved earlier
g. OPTIONAL: set the background color to be aware if the files looked at are from live or dev
    i. Dev is set for green; changes can be made freely
    ii. Live is set for orange; changes must be made cautiously

8. **Set up Advanced options for file comparison and synchronization**
    a. Select the 'Advanced' tab option above the protocol field
        i.



    b. Set the Default local directory to the GitHub folder directory
    c. Set the Default Remote directory to the following
        i. /home/[CPANEL_USERNAME]/public_html/
    d. Enable the checkboxes for 'Use syncronized browsing' and 'Directory comparison'
    e. Press 'Connect'
    f. Enter the password to access the files and directories
9. **You're Done!**

Setting Up .gitignore File For Repository

This document will walk the user through steps of how to create a .gitignore file for a GitHub repository. To understand more about how .gitignore files work, please reference the link below:

How to Use .gitignore

1. **Create a text file called gitignore.txt (or gitignore.rtf, etc.) > {**
    a. For Mac, you can use textEditor for this process.
    b. Enter a note at the top line of the file with name of the website this file will be set for.
        i. Example: #360twin Ignore File
    c. Save this file within the root directory of the clone repository.
        i. Git > {Website Clone Repository Name} > [PUT FILE HERE] **}**
2. **Change the name of the file and remove the extension > {**

a. Through the terminal, navigate to the repository.
   i. Use ‘**cd**’ [DIRECTORY_NAME] to navigate
      1. Use ‘**ls**’ to show the directories contents (helpful while searching)
b. Once inside the directory, use the ‘**mv**’ command to update the file name.
   i. Example:

```
mv gitignore.rtf .gitignore
```

   ii. This process will convert the file and hide it following normal processes.
   iii. Ensure that you can see it in your folder by pressing [**CMD**] + [**SHFT**] + [**.**]
      1. For clarity sake, press the command, shift, and period simultaneously. **}**

3. **Optional: Update all files of this type to open up in preferred text editor > {**
   a. Open Finder to the directory of where the .gitignore file is.
      i. *Ensure that you can see it in your folder by pressing [CMD] + [SHFT] + [.]*
         1. For clarity sake, press the command, shift, and period simultaneously.
   b. Right-click on the .gitignore file and select ‘Get Info’ from the list.
      i. The focus on this item is not carried over naturally, so it may appear in the background behind other applications.
   c. Expand the section listed as ‘Open With’
   d. The default object will be listed; change this by clicking into it and updating it to the preferred application.
   e. Also Optional: Select ‘Change All’ to ensure that the specified application is used for all file types like that one. **}**
4. **Open the file up and add file names or directory names under your comment line**
5. **You’re done!**

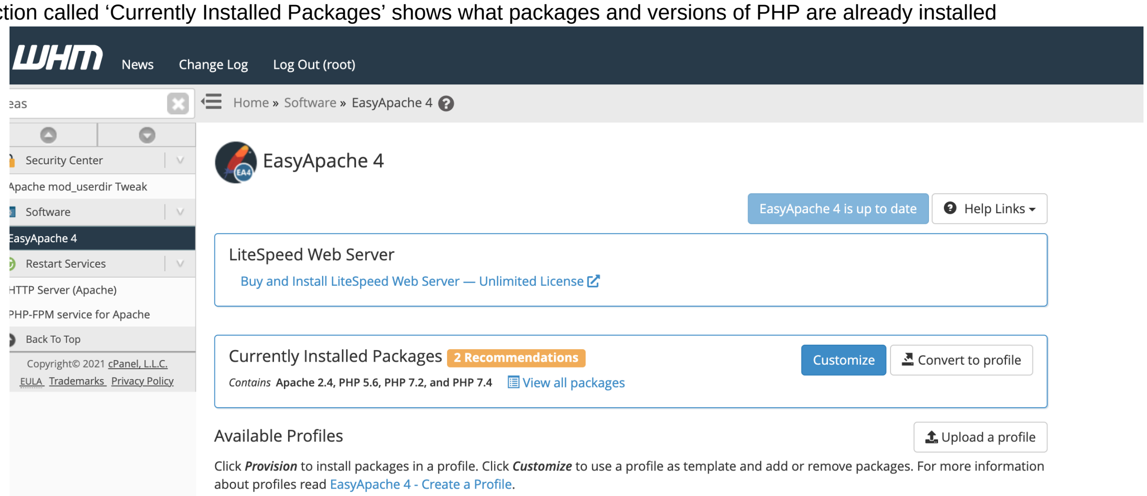
How To Set Up Magento 2 Site

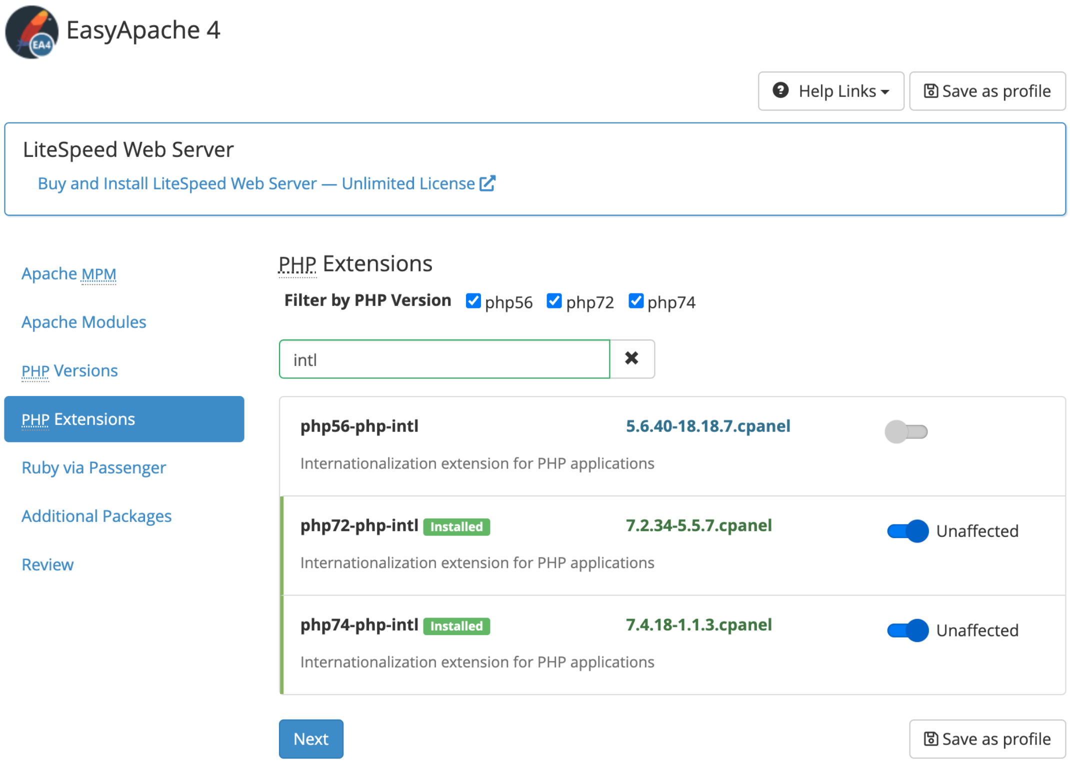
This page will serve as documentation of steps required to set up the Magento 2 site through the use of Composer. There are several steps that may be required depending on the setup and which steps have already been completed on the server.

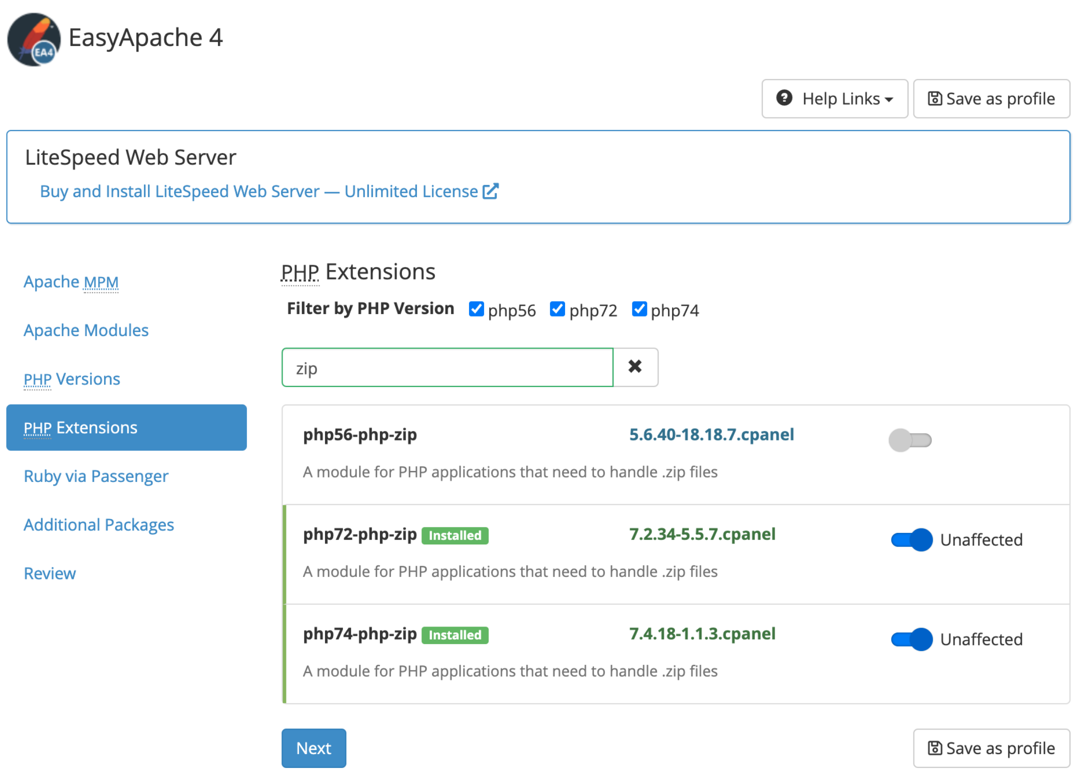
Requirements:

- PHP: 7.3 or higher
- Composer

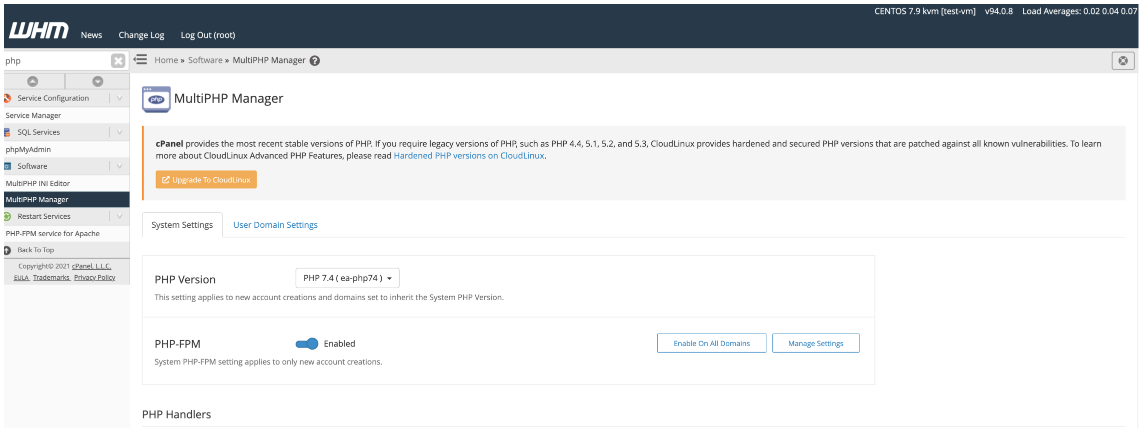
1. **Create the cpanel account for the new domain**
   a. From the ‘List Accounts’ page or the ‘Manage Your Accounts’ page, select ‘Create a New Account’
   b. Enter the domain name for the site
      i. For domains that using a subdomain, ensure to use the prefix.domain
   c. Enter a username and use the password generator to create a strong password
   d. Set the email to the group email so all notifications are received to the dev team
      i. dev@dang-designs.com
2. **Add the cpanel user to the wheel group to allow downloads for that account only**
   a. On the search bar on the left-hand side in WHM, type in ‘wheel’
   b. Under Security Center, select Manage Wheel Group Users
   c. There will be two boxes; one is for adding and the other is for removing. Select the user from the bottom box and press ‘Add to Group’
      i. This is done so that composer can be run from the user level instead of the root (which is consider unsafe)
3. **If the PHP version is not greater than 7.3, add it and provide the required dependencies**
   a. On the search bar on the left-hand side in WHM, type in ‘easy’
   b. Under Software, select EasyApache 4
   c. The section called ‘Currently Installed Packages’ shows what packages and versions of PHP are already installed
      i.



d. If a new version of PHP is required, select 'Customize' as shown in the picture above
e. Select PHP Versions from the menu on the left and select the version of PHP you would like to provision
f. With any new version of PHP, additional steps are needed to select the extension (or dependencies) need for composer
g. With PHP Extensions selected, in the search box type "intl" and select the toggle for your php version to include it in the packages to obtain
    i. 

h. With PHP Extensions selected, in the search box type "zip" and select the toggle for your php version to include it in the packages to obtain
    i. 

i. Select Review on the left-hand menu

j. Select Provision and the packages/versions of PHP will begin downloading to the system
k. The terminal may still reference the older package of PHP if it is not pointed at the new one. On the search bar on the left-hand side in WHM, type in ‘php’
l. Under Software, select MultiPHP Manager
m. Update the PHP Version to the new PHP version installed
    i. 

    ii. This sets the terminal to the proper php version, but does not affect the version used on your site. To update the version used on your site, go to User Domain Settings and update the individual site as needed to reflect the proper PHP version
n. Need to also install libsodium package for Magento 2 specifically; however, this package cannot be installed through easyApache. On the search bar for WHM, type ‘PECL’
    i. Use the terminal to get this first into the system
        1. Login to WHM and go to 'Terminal'.
        2. Type:

```
yum -y install libsodium libsodium-devel
```

    ii. Under Software, select Module Installers
    iii. Select PHP PECL
    iv. Perform a search for ‘Sodium’ and install the module
        1. This installation also places it in the PHP.ini file
            a. Make sure you have the right PHP version selected
o. Although it may not be needed, restart the Apache server to avoid any potential issues with the associated new packages and php Versions
    i. On the search bar on the left-hand side in WHM, type in ‘apache’
    ii. Under Restart Services, select HTTP Server (Apache)
    iii. Press “Yes” to restart the service
        1. Keep in mind that this restart is a restart of the service, and not a restart of the entire Google VM server.

4. **Increase the memory size for the PHP version to remove issues that may arise later**
    a. On the search bar on the left-hand side in WHM, type in ‘php’
    b. Under Software, select MultiPHP INI Editor
    c. Select your version of PHP
    d. Scroll down to memory_limit and set to 6400M
    e. Scroll down to the bottom and press Apply
5. **Install Elasticsearch and run the service**
    a. Use the steps here
        i. https://support.cpanel.net/hc/en-us/articles/360055446413-How-To-Install-Run-Elasticsearch
6. **Login as the cpanel user in the terminal and use composer to pull the magento depository**
    a. On the search bar on the left-hand side in WHM, type in ‘terminal’
    b. Under Server Configuration, select Terminal
        i. The terminal automatically jumps in as the root user.
    c. We need to go back a directory to then access the new account we set up. Enter the following command

```
i. cd ../
```

    d. Enter the home directory

    i. ```
       cd home
       ```

e. Using ‘ls’ you can see a list of all the directories/files available in this directory. While here, we need to set the new user as the active owner of their files to allow downloads, MKDIR permissions etc.

    i. ```
       chown m2devcapitalcity m2devcapitalcity
       ```

f. Now that the user has ownership over their account, we can enter the account and switch to the user for composer commands

    i. ```
       cd m2devcapitalcity
       su m2devcapitalcity
       ```

    ii. The user in the terminal should reflect this change of user where root@test-vm will be altered to m2devcapitalcity@test-vm (if your user is different, the username will be different for clarity’s sake)

g. Go to the public_html location for the user
    i. should be found in home/<USER>/public_html

h. Enter the following command

    i. ```
       composer create-project --repository-url=https://repo.
       magento.com/ magento/project-community-edition /home
       /m2devcapitalcity/public_html
       ```

        1. Make sure to edit <m2devcapitalcity> to be the name of your user

    ii. You will be prompted for a Magento Access Key
        1. Go to the link below to set one up (REQUIRES Account Setup)
        2. https://marketplace.magento.com/customer/accessKeys/
        3. Enter the public key as the username and the private key as the password

i. The system will download the contents of the repo to your web server in the public_html file

7. **Install the rest of the contents through the command line interface**
    a. Here is a sample of how this will need to be done

        i. ```
           php bin/magento setup:install --base-url=https://m2dev.
           capitalcityshoes.com/ \
                                   --base-url-secure=https://m2dev.
           capitalcityshoes.com/ \
                                   --admin-firstname=Josh \
                                   --admin-lastname=G \
                                   --admin-email=joshg@dang-designs.
           com \
                                   --admin-user=Joshg \
                                   --admin-password="vY-u6cHl4J?C" \
                                   --db-name="
           m2devcap_capitalcityshoes_2021" \
                                   --db-host="localhost" \
                                   --db-user="m2devcap_capital_user"
           \
                                   --currency=USD \
                                   --timezone=America/Chicago \
                                   --use-rewrites=1 \
                                   --db-password="vY-u6cHl4J?C"
           ```
           ```

ii. ^^ This is not the way listed in the actual documentation provided by Adobe and was found at the article below from Linuxize

b. If this worked successfully, a message like the one below should appear. If not, find out what other dependencies this behemoth needs and add it in

i.

```
Cache cleared successfully
[Progress: 1295 / 1298]
Disabling Maintenance Mode:
[Progress: 1296 / 1298]
Post installation file permissions check...
For security, remove write permissions from these directories: '/home/m2devcapitalcity/public_html/app/etc'
[Progress: 1297 / 1298]
Write installation date...
[Progress: 1298 / 1298]
[SUCCESS]: Magento installation complete.
[SUCCESS]: Magento Admin URI: /admin_10aq5z
Nothing to import.
[root@test-vm public_html]#
```

8. Create the Magento Cron tab

a. Navigate from public_html to bin

i.
```
cd bin
```

b. Install the cron tab

i.
```
php magento cron:install
```

c. Confirm working by testing it

i.
```
crontab -l
```

9. **Modify the htaccess file**

a. NOTE: The htaccess file to modify is found under public_html > pub

i. All items are passed through the pub folder by default for additional security

b. Inside of the htaccess file, remove the deny rules from the sections where access is denied to the cron.php and release notes.

c. For testing site:

i. Add this line to not allow any traffic that is not from our site

ii.
```
order deny,allow
deny from all
allow from 47.205.19.21
```

iii. May need to update the ip address based on current ip

d. This will allow the site to be pulled up

10. **If this error is found while attempting to login as the admin user**

a.
```
public_html/pub/static/adminhtml/Magento/backend/en_US
/requirejs-config.js" is not writable
```

b. Use the following command to update the access rights to the static folder

i.
```
sudo chmod -R 777 /home/m2devcapitalcity/public_html/pub
/static/
```

11. **Disable Two Factor Authentication**
    a. While this may need to be turned back on in the future, there is no way to set this up without having access to the information; therefore we need to disable to handle the backend first.
    b.
    ```
    bin/magento module:disable Magento_TwoFactorAuth
    bin/magento cache:flush
    ```
    c. From this process, the interceptor file may be removed. Add this back in with:
        i.
        ```
        php bin/magento setup:di:compile
        ```
12. **Set a cron to reindex every hour**
    a. Navigate to the Cpanel used for the site
    b. Find the cron jobs option and select it
    c. Under Common Settings > choose “Once per Hour”
    d. Under Command, place the following code:
        i.
        ```
        php bin/magento indexer:reindex
        ```
    e. Press ‘Add New Cron Job’
13. **Set up the catalog site**
    a. Create a new child theme for catalog
        i. Navigate to the vendor/Magento directory
        ii. Copy the new child theme (called ‘theme-frontend-luma-child’) to a new folder called ‘theme-frontend-luma-child-catalog’
        iii. Go into the new directory
        iv. Update composer.json
            1. Update the name to ‘magento/theme-frontend-luma-child-catalog’
        v. Update registration.php
            1. Update the second argument to ‘frontend/Magento/luma-child-catalog’
        vi. Update theme.xml
            1. Update the title to be ‘Magento Luma Child Catalog’
        vii. Navigate to public_html/app/design/frontend/Magento/
        viii. Create a new directory called ‘luma-child-catalog’
        ix. Copy the files over from luma-child to luma-child-catalog
        x. Enter the luma-child-catalog directory
        xi. Edit the composer.json, registration.php, and theme.xml files like aforementioned (must be identical)
        xii. Reindex the site to ensure the new theme is able to be selected
        xiii. Set the theme under Content > Configuration (on the backend of the site)
    b. Remove checkout as an option
        i. Navigate to the Stores > Configuration
        ii. Set Scope to Catalog website (not the store view)
        iii. Navigate on the configuration panel to Sales > Checkout
        iv. Change Enabled Onepage Checkout to No
        v. Press save
    c. Update the css to hide the fields required
        i. Navigate to the app/design/frontend/Magento/luma-child-catalog/web/css/local-m.css
        ii. Add the following changes to the file
            1.
            ```
            /* Updates to set up Catalog */
            div.price-box.price-final_price, .product-item-inner, .product-options-bottom, .product-social-links, #product-options-wrapper > div > div:nth-child(4) > label, #product-options-wrapper > div > div:nth-child(5) > label{
                display: none;
            }
            ```

```
#select_10013, #select_9799, #select_9913,
#select_10198, #options-10199-list, #select_9987,
#options-10019-list, #select_9897, #select_9968,
#select_9801, #select_9914, #options-10033-list{
    display: none;
}
div.header:nth-child(1) > ul:nth-child(1) > li:nth-
child(1), div.header:nth-child(1) > ul:nth-child(1)
> li:nth-child(2), div.header:nth-child(1) > ul:nth-
child(1) > li:nth-child(5){
    display: none;
}
.block.filter, .sidebar.sidebar-additional,
#search_mini_form > div.field.search > div > div.
nested > a, body > div.page-wrapper > footer > div >
ul > li:nth-child(3) > a {
    display: none;
}
body > div.page-wrapper > header > div.header.
content > div.minicart-wrapper > a{
    display: none;
}
#product-options-wrapper > div > div:nth-child(4) >
div, #product-options-wrapper > div > div:nth-child
(5) > div, #product-options-wrapper > div > div:nth-
child(4) > div > select{
    display: none;
}
#product-options-wrapper > div > div:nth-child(6) >
label, #product-options-wrapper > div > div:nth-child
(6) > div{
    display: none;
}
.copyright{
    text-align: center;
    background-color: #6e716e;
    color: #fff;
    padding: 10px;
}
.page-footer{
   background-color: #f4f4f4;
}
div.header:nth-child(1){
    color: white;
}
#maincontent > div.columns > div.column.main{
    width: 100%;
}
```

iii. Recompile the static files to ensure the changes are seen on the frontend

```
1. php bin/magento setup:di:compile
```

14. **You're Done!**

1. **LESSONS LEARNED**
   a. You may have to disable the elasticsearch module to run the installer; this is not covered in Adobe documentation.

```
i. php bin/magento module:disable {Magento_Elasticsearch,
   Magento_InventoryElasticsearch,Magento_Elasticsearch6,
   Magento_Elasticsearch7}
```

   b. The site not showing an external IP address is critical. The IP address must be set as the same as the one that the domain is set to (shared IP)
      i. This is done by removing the line from etc/domainips
      ii. afterwards, the IP can be used for the matching site
2. **You're Done!**

References	
Link	**Notes**
https://linuxize.com/post/linux-chown-command/	
https://devdocs.magento.com/guides/v2.4/install-gde/composer.html	
https://marketplace.magento.com/customer/accessKeys/	
https://linuxize.com/post/install-magento-2-on-centos-7/	
https://devdocs.magento.com/guides/v2.4/install-gde/system-requirements.html	Magento 2 requirements
https://support.cpanel.net/hc/en-us/articles/360055446413-How-To-Install-Run-Elasticsearch	Install Elasticsearch and run server

How To Migrate Data From Magento 1 To Magento 2

This page will store the steps used to migrate data from Magento 1 to Magento 2. These steps use composer, like the initial installation, and require certain ports to be opened. This document, like most others will be updated as the process is completed for conversion.

> Magento 1 extensions are not compatible with Magento 2

1. **Install Magento 2 Data Migration Tool via Composer**
   a. Using the WHM terminal, navigate to the public_html location of the new magento 2 site
   b. Find out the version used for the magento 2 site

```
i. php bin/magento --version
```

   c. Sign in as the user account for the site
      i. In the instance of mage 2 setup for capitalcityshoes, the account was m2devcapitalcity

```
1. su m2devcapitalcity
```

   d. Prep the composer configuration

i.
```
composer config repositories.magento composer https://repo.magento.com
```

ii.
```
composer require magento/data-migration-tool:<version>
```

1. Use the version found in step 1.b
2. If the version has -pl, remove that before submission

e. Log into the magento marketplace to retrieve your Access keys under the Products section
   i. This step may not be required if you used the keys to set up the files initially with the same access key

2. Set up stores and themes to match new installation
3. **Configure Magento 2 Data Migration Tool**
   a. The following location will contain the files used for the data migration tool

i.
```
/vendor/magento/data-migration-tool/etc/
```

1. There should be three options of folders: commerce to commerce, open source to commerce, and open source to open source

ii. For our purposes, we will use **opensource-to-opensource**

b. Create the config.xml file for the data migration tool
   i. rename **config.xml.dist** to **config.xml** within the directory of the older magento version you're running
      1. This mage version can be found in the bottom right corner of the admin panel OR in app/mage.php

2.
```
public static function getVersionInfo()
    {
        return array(
            'major'     => '1',
            'minor'     => '9',
            'revision'  => '3',
            'patch'     => '7',
            'stability' => '',
            'number'    => '',
        );
    }
```

3. In this case, the version is 1.9.3.7

ii. Update the configuration pieces with the config.xml
   1. should look like this

a.
```
<source>
        <database host="localhost" name="
magento1" user="root" password=""/>
    </source>
    <destination>
        <database host="localhost" name="
db_name" user="root" password=""/>
    </destination>
<options>
    <crypt_key>Magento1-Encrypted-Key<
```

```
/crypt_key>
</options>
```

b. to this

```
c. <source>
        <database host="localhost" name="
db_name_mage1" user="db_user_mage1" password="
db_password_mage1"/>
    </source>
    <destination>
        <database host="localhost" name="
db_name" user="db_user" password="db_password"/>
    </destination>
<options>
    <crypt_key><![CDATA[MAGE1_KEY]]></crypt_key>
</options>
```

d. May need to update localhost for live conversion
e. Crypt key is found in app/etc/local.xml for the magento 1 installation

2. Save the file

4. **Migrate Settings Using Magento 2 Data Migration Tool**
    a. Within the terminal, use the following command to transfer over the settings

```
i. php bin/magento migrate:settings --reset <path to your
config.xml> -a
```

1. path in this case would be similar to “vendor/magento/data-migration-tool/etc/opensource-to-opensource/1.9.3.7/config.xml”

ii. If error “No commands defined the in the migrate namespace” shows

```
1. rm -rf var/log/* generated/* var/cache/*
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento s:s:d -f
php bin/magento cache:flush
```

2. Reattempt the previous command in a.i

iii. After the script has been completed, there should be a message displayed “Migration Completed”

5. **Update references to correct domain names**
    a. Update store_website to replace sandbox with m2dev
    b. Update the google domain with references for the m2dev sites
6. **Migrate Data Using Magento 2 Data Migration Tool**
    a. Within the terminal, use the following command to transfer over data (products, categories, orders, customers)

```
i. php bin/magento migrate:data --reset <path to your config.
xml> -a
```

1. path in this case would be similar to “vendor/magento/data-migration-tool/etc/opensource-to-opensource/1.9.3.7/config.xml”

b. After the script completes, the site should be visible on the frontend and a message should be displayed that says “Migration Complete”

c. Debug.log in var/log directory may not allow access to the backend of the site as admin due to issues about not being able to write to the log

   i. Within the terminal, navigate to the debug.log file

   ii. Set the owner of the folder to m2devcapitalcity (or main user) instead of root

```
1. chown m2devcapitalcity:m2devcapitalcity debug.log
```

7. **Reindex the data and confirm all pieces were transferred over and working**

   a. Within the terminal, navigate to the root public_html folder

```
i. php bin/magento indexer:reindex
```

1. The indexer can also be ran for specific indexes separately to move through them faster as needed. To find the names associated to the fields to run the indexer for run:

```
2. php bin/magento indexer:info
```

3. This will display a list and the far left column contains the indexer to run

```
4. php bin/magento indexer:reindex [INDEX_NAME_HERE]
```

8. **Revert the file permissions for the static pub folder**

   a. Navigate through the file directory to pub/static

   b. Update the static folder to permissions 0775

9. **Add data to the .htaccess folder in pub for store redirects**

   a. In order for the stores to go to the appropriate location, the must also be routed there through the htaccess file like so

```
i. SetEnvIf Host catalog.m2dev.capitalcityshoes.com
MAGE_RUN_CODE=catalog
SetEnvIf Host catalog.m2dev.capitalcityshoes.com
MAGE_RUN_TYPE=website

SetEnvIf Host b2b.m2dev.capitalcityshoes.com
MAGE_RUN_CODE=b2b
SetEnvIf Host b2b.m2dev.capitalcityshoes.com
MAGE_RUN_TYPE=website

SetEnvIf Host ghs.m2dev.capitalcityshoes.com
MAGE_RUN_CODE=soga_ghs
SetEnvIf Host ghs.m2dev.capitalcityshoes.com
MAGE_RUN_TYPE=website

SetEnvIf Host stateoffloridagear.m2dev.capitalcityshoes.
com MAGE_RUN_CODE=stateofflorida
SetEnvIf Host stateoffloridagear.m2dev.capitalcityshoes.
com MAGE_RUN_TYPE=website
```

```
SetEnvIf Host flpv.m2dev.capitalcityshoes.com
MAGE_RUN_CODE=flpv
SetEnvIf Host flpv.m2dev.capitalcityshoes.com
MAGE_RUN_TYPE=website
```

ii. Without additional plugins, however, the stores will not require log in.
iii. Additionally, without the custom options being available, the products may not be seen either

b. Make sure to update the base URL and secure base URL for all of these sites; otherwise, the default will be the main store URL (m2dev.capitalcityshoes.com)
   i. Updated in Stores > Configuration > (Select Scope) > Web

10. **Correct Stores for indexing as needed**
    a. There was an issue where stores were orphaned and the data was transferred over from the orphaned stores. To correct this, we must update the database to update the stores associated group_ids to account for the difference.
       i. Sheet from sandbox.capitalcityshoes transfer over
          1. https://docs.google.com/spreadsheets/d/1JggQjgkyE_yT_wjfLkezDE5BxrUOZSYqkakPeSPiBO0/edit#gid=0
       ii. Take a copy of the list of stores and store groups and save them to a google sheet
       iii. Use a vlookup to find the stores that don't have a group
       iv. Remove or update those stores so they have an affiliated group
11. **Move the images over from the original site**
    a. All images stored in Media in Magento 1 need to be moved to Magento 2 in the pub/media folder
       i. Due to the mass size of Products, navigate to media/catalog/product and pull a zip of those files.
       ii. Move the customoptions and catalog/products separtately. Some files don't want to play well with others and the size of these files can be cumbersome
    b. Clear the cache and reindex the site
12. **Create a custom child theme**
    a. Navigate in the file manager to vendor/magento and find the bottom records of theme-frontend-luma and theme-frontend-blank
       i. These are the default themes
       ii. Create a folder called theme-frontend-luma-child
          1. copy the files from the luma theme here
    b. Create a new folder under the app/design/frontend/Magento folder
       i. Name the folder the same as the parent with "-child" attached to know where the data comes from
    c. Create a theme.xml file
       i. Add the following lines in (modify what doesn't match)

ii.
```
<!--
/**
 * Copyright © Magento, Inc. All rights reserved.
 * See COPYING.txt for license details.
 */
-->
<theme xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="urn:magento:
framework:Config/etc/theme.xsd">
    <title>Magento Luma Child</title>
    <parent>Magento/blank</parent>
</theme>
```

    d. Create a composer.json file
       i. Add the following lines in (modify what doesn't match)

ii.
```
{
    "name": "Magento/luma-child",
    "description": "N/A",
    "require": {
        "php": "~5.5.0|~5.6.0|~7.0.0",
        "magento/theme-frontend-luma-child": "100.0.*",
        "magento/framework": "100.0.*"
```

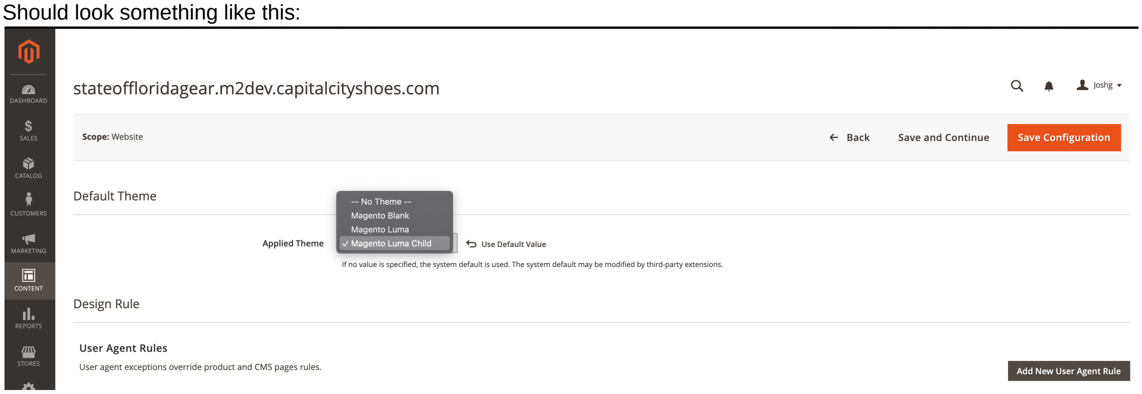
```
        },
        "type": "magento2-theme",
        "version": "100.0.1",
        "license": [
            "OSL-3.0",
            "AFL-3.0"
        ],
        "autoload": {
            "files": [
                "registration.php"
            ]
        }
    }
```

e. Create a registration.php file
    i. Add the following lines in (modify what doesn't match)
    ii.

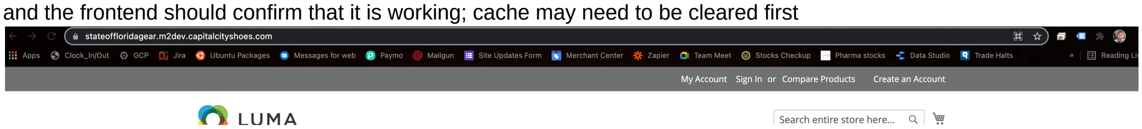
```
<?php
/**
 * Copyright © 2015 Magento. All rights reserved.
 * See COPYING.txt for license details.
 */
\Magento\Framework\Component\ComponentRegistrar::register(
    \Magento\Framework\Component\ComponentRegistrar::
THEME,
    'frontend/Magento/luma-child',
    __DIR__
);
```

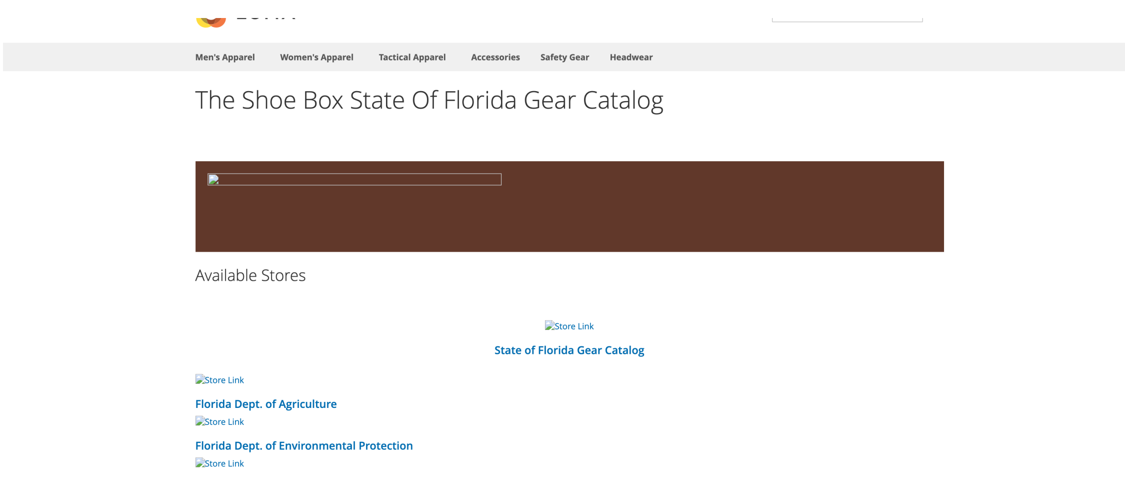
f. Create two directories called "etc" and "web"
    i. The etc is for a copy of the view.xml file from the theme folder created above
    ii. The web folder is for css and static change data
g. Navigate to the pub/static/frontend/Magento/ directory
h. Create a new folder called "luma-child" (matching what was already created previously)
i. Copy all the contents of the pub/static/frontend/Magento/luma/en_US directory to the new luma-child folder
j. Set the theme for the site under the admin panel (Content > Configuration)

to confirm that the new child theme is visible and working
    i. Should look something like this:
    ii.



    iii. and the frontend should confirm that it is working; cache may need to be cleared first
    iv.





v. The images missing in the photo are due to different issues not related to the theme working

k. Update the database theme table to use the default css files
   i. Got to the theme table in the database
   ii. Change the type of the child theme newly created from 1 to 0

13. **Move the store switcher to the header**
    a. Go to the main layout default xml
       i.
```
vendor/magento/module-search/view/frontend/layout/default.xml
```
    b. Add the line to move the element to the top of the page (after top.links) under the final reference block ending argument
       i.
```
<move element="store_switcher" destination="header.panel"
after="header.links" />
```
    c. By placing this item at the bottom, it ensures that the store switcher is added to the banner instead of the header and avoids dealing with css issues

14. **Update CSS**
    a. In the new child theme, updates to the CSS can be made in these two files (***NOT PERMANENT***)
       i. public_html/pub/static/frontend/Magento/luma-child/en_US/css/styles-l.css
       ii. public_html/pub/static/frontend/Magento/luma-child/en_US/css/styles-m.css
          1. This file is used primarily for mobile and desktop, while the other files is only for desktop; i.e. both are essential to retain a proper layout on the site.
    b. For permanent changes, update this file
       i. In app/design/frontend/Magento/[Child-theme-name]/
          1. Add the folders
             a. Magento_Theme/layout/
          2. Inside of the layout folder, add a file called `default_head_blocks.xml`
          3. Add this code to this file
             a.
```
<?xml version="1.0"?>
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../../../../vendor/magento/framework/Module/etc/module.xsd">
  <head>
    <css src="css/local-m.css" />
    <css src="css/local-l.css" media="screen and (min-width: 768px)"/>
  </head>
</page>
```

4. Create another file in the same directory called `default.xml`
    a. Add this code to the file

```
i. <?xml version="1.0"?>
<!--
/**
 * Copyright © Magento, Inc. All rights
reserved.
 * See COPYING.txt for license details.
 */
-->
<page xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:
noNamespaceSchemaLocation="urn:magento:
framework:View/Layout/etc
/page_configuration.xsd">
    <body>
        <referenceContainer name="header.
panel">
            <block class="
Magento\Framework\View\Element\Html\Links"
name="header.links">
                <arguments>
                    <argument name="
css_class" xsi:type="string">header links<
/argument>
                </arguments>
            </block>
        </referenceContainer>
        <referenceContainer name="footer">
            <block class="
Magento\Store\Block\Switcher" name="
store_switcher" as="store_switcher" after="
footer_links" template="Magento_Store::
switch/stores.phtml"/>
        </referenceContainer>
        <referenceBlock name="report.bugs"
remove="true"/>
        <move element="copyright"
destination="before.body.end"/>
    </body>
</page>
```

5. Inside of app/design/frontend/Magento/[Child-theme-name]/web/css/
    a. Add the following files listed from the above inclusion

```
i. local-m.css
```

ii. `local-l.css`

6. Recompile the static files to allow changes on the frontend

a. `php bin/magento setup:di:compile`

15. **Set up BrainTree Production Account for Credit Cards**
    a. Navigate to Stores > Configuration
    b. Navigate to Sales > Payment Methods
    c. Next to Braintree, press ‘Configure’
    d. There will be writing in blue text that says “click here to signup for a Braintree Account”, Press it
        i. Fill out the information and submit the application
        ii. Navigate to the home screen to obtain credentials (keep this screen active for the remainder of setup)
    e. Change the environment to production
    f. Payment action to Authorize
    g. Copy over the Merchant ID to the Merchant ID field
    h. Copy over the public key to the Public Key field
    i. Copy over the Private key to the Private Key field
    j. Press validate credentials
    k. Change “Enable Card Payments” to Yes
    l. Expand ‘Advanced Braintree Settings’
        i. Ensure that only accepted credit card types are selected
    m. Expand ‘Country Specific Settings’
        i. Select Specific Countries
        ii. Select United States from the list
    n. Add additional payment options as approved by Dan
16. **You’re Done!**

References	
Source	**Site**
Adobe Migration Tool Docs	https://devdocs.magento.com/guides/v2.3/migration/migration-tool-install.html
Adobe Migration Planning	https://devdocs.magento.com/guides/v2.3/migration/migration-plan.html
Alternative Migration Walkthrough	https://meetanshi.com/blog/migrate-from-magento-1-to-magento-2/
Custom Theme Creation	https://www.mageplaza.com/devdocs/how-to-create-magento-2-theme.html

How To Set Up Source Report for Data Studio (360twin)

This page will serve as a guide to walk a user through the process of getting the analytics report to show where a source came from. This report can be pulled from the Google Analytics information and narrowed to be submitted within Google Data Studio to ensure that the data looks accurately when displayed.

1. **Log into Google Analytics**
2. **Navigate to analytics account to use**
3. **On the side-bar navigation pane, select Behavior > Site Content > All Pages**
4. **In the secondary dimension, select Acquisition > Source**
5. **In the textbox for the advanced search field, enter “checkout”**
6. **Press ‘Enter’**
7. **To show all fields, select the ‘Show rows’ toggle and update it to 50 (*or the amount of records needed to display*).**
8. **Add a secondary dimension using Acquisition > source**
9. **Copy and paste the fields to a Google Sheet**
    a. Find the order acquisition page here:
        i. https://docs.google.com/spreadsheets/d/1JuoiovwhuQ8N4mx4rrzSXvVIqs9Rn6J3r2Nkbzwfgwl/edit#gid=980422346
10. **Set up the data to be ready for Analytics**
    a. OrderStage is used for new monthly data for orders
    b. sourceStage is used for new monthly data for sources from woocommerce
    c. orders is used to host all orders from 360twin

d. Sources is used to host the data that will be used for Data Studio

11. **You're Done!**

Removing Foreign Characters Within Website Text

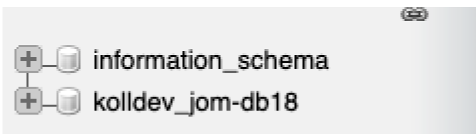
If you notice any strange characters within the text on the frontend of a website ie…
L-Sep with a square around it, It is caused by doing a copy/paste from the source directly to the content section of the Text Box within that article page you are working on.

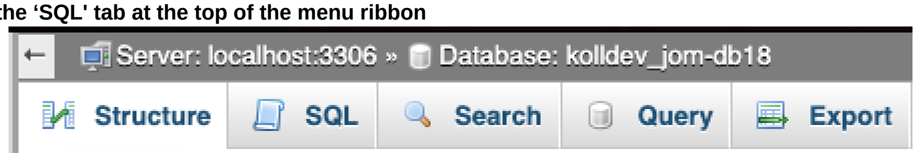
The recommended transfer method is to copy/paste to a text editor first to remove any formatting and then copy/paste to the Content Text box you are working on in the backend.

These types of foreign characters issues seem to only be visible in Windows 10 using a Chrome browser, when a site is being built in Joomla/Yootheme.

Injecting An Admin User In Joomla Via phpMyAdmin

This page will show how to add users through the C-Panel; more specifically, within the phpMyAdmin section.

1. **Log into the C-Panel for the site**
2. **Open the phpMyAdmin tool**
3. **Select the database by clicking on the name**
   a. *Make sure to select the site's database and* ***NOT*** *the general 'information_schema'*
   b. 

4. **Select the 'SQL' tab at the top of the menu ribbon**
   a. 

5. **Insert the code to add a user while making sure to update the values specified**
   a. ***PREFIX*** *is the piece amended to the beginning of each table (4 INSTANCES)*
   b. **NAME** is the name of yourself (i.e. Josh Greenert)
   c. **USERNAME** is the name that will be used for login purposes *(2 INSTANCES)*
   d. **PASSWORD** is the password for the login *(2 INSTANCES)*
   e.

```
INSERT INTO `PREFIX_users`
   (`name`, `username`, `password`, `params`, `registerDate`,
`lastvisitDate`, `lastResetTime`)
VALUES ('NAME', 'USERNAME','PASSWORD', '', NOW(), NOW(),
NOW());
INSERT INTO `PREFIX_user_usergroup_map` (`user_id`,`group_id`)
VALUES (LAST_INSERT_ID(),'8');
UPDATE `PREFIX_users` SET `password` = MD5('PASSWORD') WHERE
`PREFIX_users`.`username` = 'USERNAME';
```

6. **You're Done!**

Alternative Code Options

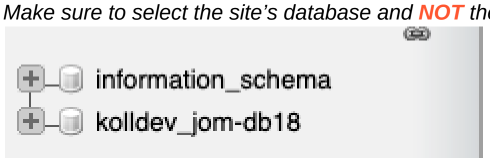
```
INSERT INTO `ufjb8_users`
   (`name`, `username`, `password`, `params`, `registerDate`,
`lastvisitDate`, `lastResetTime`)
VALUES ('dangteam', 'dangteam','dangteam', '', NOW(), NOW(), NOW());
INSERT INTO `ufjb8_user_usergroup_map` (`user_id`,`group_id`) VALUES
```

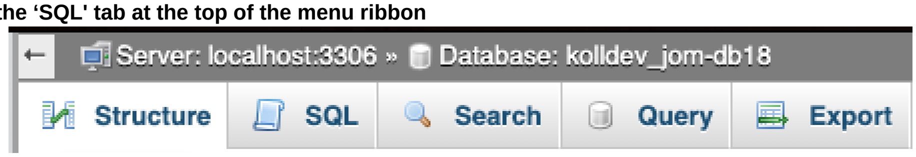
```
(LAST_INSERT_ID(),'8');
UPDATE `ufjb8_users` SET `password` = MD5('dangteam') WHERE
`ufjb8_users`.`username` = 'dangteam';
```

References
https://docs.joomla.org/How_do_you_recover_or_reset_your_admin_password%3F

Injecting An Admin User in WordPress Via phpMyAdmin

This page will show how to add users through the C-Panel; more specifically, within the phpMyAdmin section.

1. **Log into the C-Panel for the site**
2. **Open the phpMyAdmin tool**
3. **Select the database by clicking on the name**
    a. *Make sure to select the site's database and* ***NOT*** *the general 'information_schema'*
    b. 

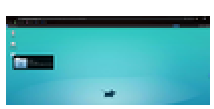
4. **Select the 'SQL' tab at the top of the menu ribbon**
    a. 

5. **Insert the code to add a user while making sure to update the values specified**
    a. Unlike the Joomla table updates, this one requires you to check the primary key to ensure that it is available to be set for the user being created.
        i. You can do this by performing a `select * from wp_users;`
    b. **DATABASE_NAME** = The name of your database *(3 instances)*
    c. **USERNAME** = The username used for logging in *(2 instances)*
    d. **PASSWORD** = The password used for logging in *(2 instances)*
    e. **NAME** = The name of the user for the account *(2 instances)*
    f. **EMAIL** = The email for the user's account
    g. **ID** = The primary ID that will be placed into the table (*3 instances*)
    h.

```
INSERT INTO `DATABASE_NAME`.`wp_users` (`ID`, `user_login`,
`user_pass`, `user_nicename`, `user_email`, `user_url`,
`user_registered`, `user_activation_key`, `user_status`,
`display_name`)
VALUES (ID, 'USERNAME', MD5('PASSWORD'), 'NAME', 'EMAIL', '',
NOW(), '', '0', 'NAME');
INSERT INTO `DATABASE_NAME`.`wp_usermeta` (`umeta_id`,
`user_id`, `meta_key`, `meta_value`) VALUES (NULL, ID,
'wp_capabilities', 'a:1:{s:13:"administrator";s:1:"1";}');
INSERT INTO `DATABASE_NAME`.`wp_usermeta` (`umeta_id`,
`user_id`, `meta_key`, `meta_value`) VALUES (NULL, ID,
'wp_user_level', '10');
UPDATE `wp_users` SET `user_pass` = MD5('PASSWORD') WHERE
`wp_users`.`user_login` = `USERNAME`;
```

6. **You're Done!**

Setting Up Ubuntu Server on Google Cloud VM



The easiest part of this process is setting up the VM instance within the Google cloud platform. However, choosing the wrong version of Linux can involve extra work and create difficulties throughout with missing libraries/dependencies/etc. Below I have listed the steps to get to a blank Ubuntu desktop running through a Chrome Remote Desktop connection. There are additional steps to take to customize after this process is completed, but those are preference-based and can be configured at the user's discretion.

1. **Select Create Instance** > {
    a. [choose a suitable name]
    b. [set the machine type to have the storage space/processor speed required]
    c. [Change the Boot disk option]
        i. [change operating system to Ubuntu]
        ii. [change version to Ubuntu 18.04 LTS]
            1. *This is the most recent stable release of for Ubuntu and contains the apt-get already configured*.
        iii. [set the size of the storage GB size needed]
    d. [leave all other values as default and select the create option at the bottom of the page.
        i. *This process may take about 2 - 5 minutes to finish up.*
    e. [Afterwards, under the connect column where VM instances are listed, on the right of SSH select the arrow and "Open in Browser Window"]
        i. *The connection will be made to the instance and the blank template of an Ubuntu machine will appear.}*
2. **Set Up The VM With a GUI and A Remote Connection** > {
    a. [Enter the following line into the shell]

```
sudo apt-get install ubuntu-desktop
```

        i. *This can take several minutes to process; be patient and wait until the process is fully complete*.
    b. [Use the documentation from Google to set up connection to Chrome Remote Desktop utility]
        i. https://cloud.google.com/solutions/chrome-desktop-remote-on-compute-engine
            1. *When installing an X Windows system desktop environment, choose Xfce; Cinnamon requires additional steps not included within the documentation and is regrettably a pain*
            2. *If you get an error when using the following command, ignore it. It won't hinder the process whatsoever*

```
sudo apt install --assume-yes task-xfce-desktop
```

            3. *Like above, if there are any error messages presented with the following command, ignore it*

```
sudo systemctl disable lightdm.service
```

            4. *Although the Chrome Browser is optional, ensure that this step is performed*
    c. [After the steps are completed within the Google documentation, sign into the support@dang-designs.com account and access the remote desktop panel.]
    d. [Connect to the new instance created and when prompted, use default configuration settings for Xfce.]}
3. **You're Done!**

Download MySQL Database & Workbench In Ubuntu

Installing the MySQL Database and Workbench require a few steps to ensure that all dependencies are added, user permissions are updated, and passwords are activated. While I intend to provide all steps possible, there may be missing parts (e.g. add user commands in SQL client, etc.). Additionally, while installing the mysql-server would generally be enough to have a working mysql database system, the workbench is needed for visual and plug-in support.

1. **Log into the root account Ubuntu > {**
    a. [If there is no password set up for the root account, use:]

```
sudo -i
```

b. [If there is a password set up for root account, use:] }

```
su
```

i. *Logging in as the root user ensures less issues later on with access and dependencies while also promoting good practices.}*

2. **Install MySQL Database and Workbench > {**

```
sudo apt-get install mysql-server
```

a. [If prompted on whether to install full file, press y (yes)]
    i. [Testing this process was complete as easy as typing ‘mysql’ into the terminal]
        1. *You’ll notice that the terminal jumps directly into a mysql script where commands can be entered to be processed by mysql*.
        2. IMPORTANT: *if you want to access the terminal as a general user, a password must be set and the user must us the following command:*

```
mysql -p
```

b. [Now that the server is installed, install the workbench]

```
sudo apt-get install mysql-workbench
```

i. *It’s important to be in root user for this process as the process will be denied otherwise*

c. [After this has installed, a new tab will be added to the Applications menu called ‘Development’]
    i. [Use this tab to open the workbench in the future, but for now leave it closed since there are still pieces to correct to connect to the database] }

3. **Add the General User to the MySQL Database > {**

a.
```
CREATE USER  ‘new_user’@'localhost' IDENTIFIED BY ‘PASSWORD’;
```

i. *The fields new_user and PASSWORD should be replaced with the user you wish to enter and the password for the user respectively.*
ii. *A password will be needed since this is how the database connection will be allowed. }*

4. **Grant User and Root Permissions for MySQL > {**

a.
```
GRANT ALL PRIVILEGES ON *.* TO 'user_name'@'localhost';
```

b.
```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost';
```

i. *By default, root user should have all permissions already but we must always be certain. Trust me, you don’t want to hop back and forth when errors occur.*

5. **Set the Root Password Through Native (deprecated) Process > {**

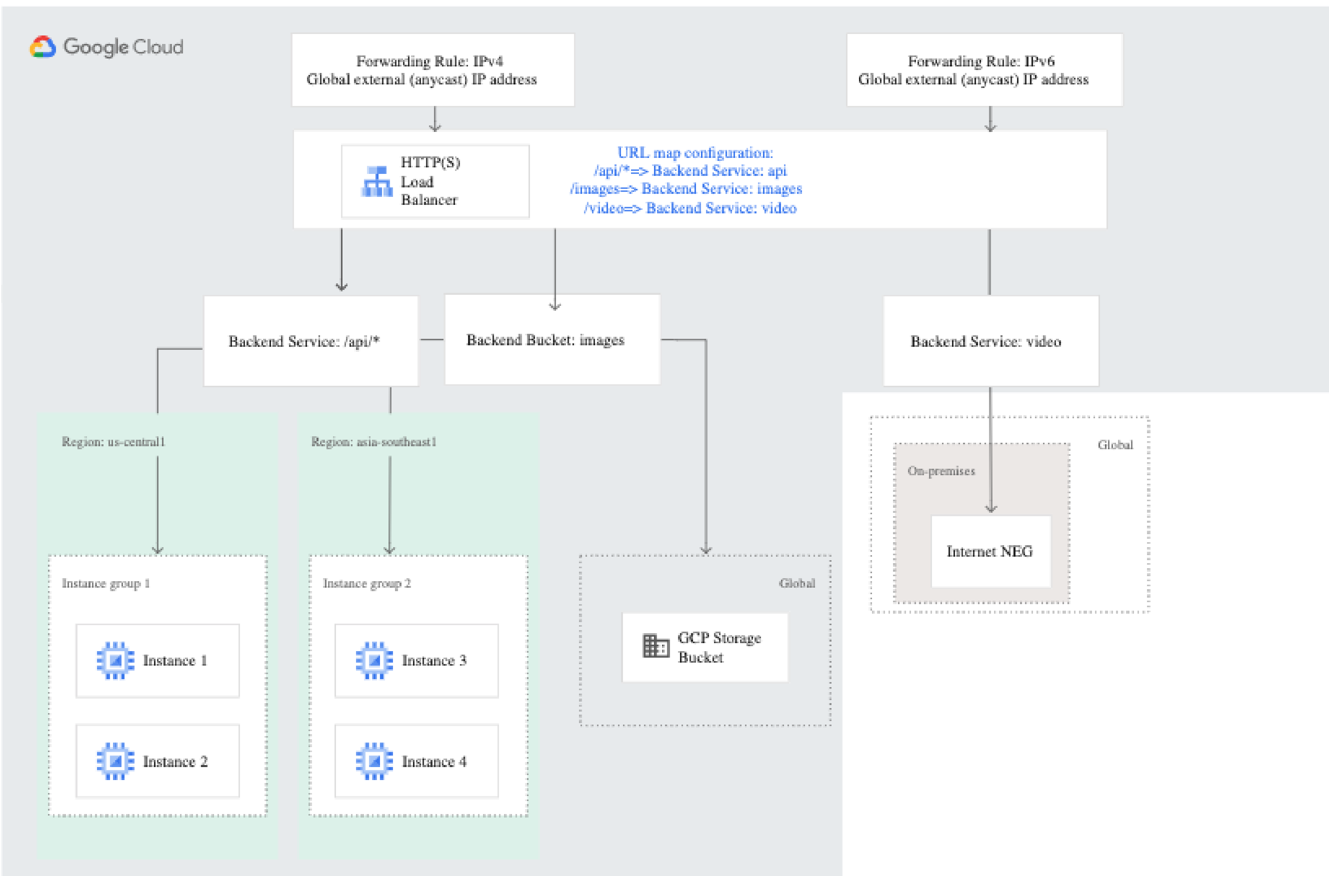
```
a. ALTER USER 'root'@'localhost' IDENTIFIED WITH
   mysql_native_password BY 'PASSWORD';
```

i. *Now when you open the MySQL Workbench, you will be prompted to enter a password before performing any actions. Once the password is entered, the Server starts up and allows connection.* }

6. **Connect to Database > {**
   a. [Open the Workbench Applications > Development > MySQL Workbench]
   b. [Enter the password when prompted for root]
   c. [Go to Database > Connect to Database]
   d. [Use default settings and click OK] }
7. **You're Connected and Done!**

Setting Up Load Balancing Between Instances

This document will detail the steps required to balance the network traffic for the Windows + Linux Google Cloud Instances. Regardless of whether the Windows instance remains, this document will provide information ensuring multiple instances can be established. Furthermore, the connection must establish that if one instance is taken down, the other instance handles the traffic in its absence.



Above is an image as a suggestion from Google for several different regions. While multi-region support isn't going to be performed in this process, the model above is similar to the process that will be used for the Windows and Linux instances.

Supported operating systems

Each supported operating system that is available on Compute Engine requires specific guest environment packages. Either Google or the owner of the operating system builds these packages.

The following outlines the creators and maintainers of the available guest environment packages:

- CONFIRMED Ubuntu 16.04 or later published to its Universe repository
- CONFIRMED Windows Server 2019

Network Load Balancing Characteristics

Network Load Balancing has the following characteristics:

- Network Load Balancing is a managed service.
- Network Load Balancing is implemented by using Andromeda virtual networking and Google Maglev.
- The network load balancers are not proxies.
- Responses from the backend VMs go directly to the clients, not back through the load balancer. The industry term for this is *direct server return*.
- The load balancer preserves the source IP addresses of packets.
- The destination IP address for packets is the regional external IP address associated with the load balancer's forwarding rule.

Network Load Balancing Set Up Caveats

Follow the steps outlined here to set up the load balancer successfully. There are a few details that can be ignored, and pieces that need to be updated for our specific usage. I'll detail those below in a set of bullet points that follow the document listed to better assist with future set ups, if needed.
https://cloud.google.com/load-balancing/docs/network/setting-up-network#configure_the_load_balancing_service

- gcloud tools are accessed through the shell directly and for our instances come pre-installed; however, they are not necessary for this process.
- Ignore the bash scripts and download information regarding apache.
- Make sure to choose the appropriate name for the firewall as it cannot be changed after set up.
- The IP addresses that should be allowed into the network listing for the firewall rules are the local IP address, and both VM external IP addresses
    - While the process defined states to list port 80, make sure to use port 28192 instead for Fishbowl servers to connect.
    - Future IP addresses of different locations can be added afterwards. Don't worry about keeping a machine too secure.
- The health check is a good feature to add but beware that if you change the port after the initial set up, both instances will always reflect as if their health check went poorly.
- Ignore the curl command listed at the bottom as this script, when entered is an infinite loop that returns nothing and performs no process in regards to what actions we need.

References
https://cloud.google.com/load-balancing/docs/network/setting-up-network
https://cloud.google.com/load-balancing/docs/https/setting-up-https
https://cloud.google.com/load-balancing/docs/https
https://cloud.google.com/load-balancing/docs/load-balancing-overview
https://turbofuture.com/computers/How-to-Setup-Remote-Desktop-Connection-Broker-for-Windows-2016
https://cloud.google.com/compute/docs/images/guest-environment
https://cloud.google.com/load-balancing/docs/network

Setting Up Log File Containing Every Update To GitHub

Each resource has its own respective information that will be required for the process to function properly. As such, each resource will have this document to serve as its parent. The below code snippet shows the available pieces that would be essentially to send information over to GitHub through PHP; however, the caveat to that is that the file sent over must be a manipulated string since the contents of the file don't actually send over.

Important Things to Note

- A SHA, or secure hash algorithm, must be present to send over to the GitHub repository. Therefore, in the code ($shafile) is an already named variable where $shafile = "sha.txt";
- $filename references a string of text that displays what the update is; thus, the string that's placed into this object must end with a file object extension (.rtf, .txt, etc.).
    - As the text files and SHA file will not be needed after completion of the code, these files can be deleted to not allow excess files to be present after script execution. Hence the unlink at the bottom of the PHP code.
- The authorization token is the PAT specified from the DangDesign GitHub account

Github Information

```
// Create the file name as the string, open it to create it, then close
it.
$filename = $shlstring;

$sqlfilename = fopen($filename, "w") or die("Unable to open txt file!");
fclose($sqlfilename);

// Put sha code into the file and close it.
$shatxtfile = fopen($shafile, "w") or die("Unable to open sha file!");
$sha = sha1_file($filename);
fwrite($shatxtfile, $sha);
fclose($shatxtfile);

// <https://github.com/joshdangdesign/updateCollection/tree
/simplehistorylogs>
// This array contains the sha, committer, and converts the new
filename to a base64 encoded object.
$data_git = array(
    'sha'=>file_get_contents($shafile),
    'message'=>'updated on ' . date('d-m-Y'),
    'content'=> base64_encode($filename),
    'committer'=> array(
        'name'=>'jgreenert+automation',
        'email' => 'jgreenert+automation@dang-designs.com')
);

// This portion converts the previous array to a json encoded object to
be able to
// send through GitHub API and the curl sends it over.
$data_string_git = json_encode($data_git);

$ch_git = curl_init();
curl_setopt($ch_git, CURLOPT_URL, '<https://api.github.com/repos
/dangdesigns/UpdateCollection/contents/360twin/simplehistorylogs/'> .
$year . "/" . $month . "/" . $day . "/" . $filename);
curl_setopt($ch_git, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch_git, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch_git, CURLOPT_POSTFIELDS, $data_string_git);
curl_setopt($ch_git, CURLOPT_HTTPHEADER, array(
    'User-Agent: USERNAME',
    'Authorization: token 828fd75c682d477f6a0c9fa1ac9e327f78995aec'
));

$result_git = curl_exec($ch_git);

// Close the URL contents.
curl_close($ch_git);

// Delete the file that was used.
unlink($filename);
```

```
// API only allows 60 records to be updated per hour. Moreover, the API
detects abuse if done too quickly.
sleep(5);
```

Getting WordPress (SimpleHistory) Logs

Through the Dev WP-ADMIN page changes throughout the site, made within the instance of direct contact, are stored by the SimpleHistory plugin. Scripts for each plugin directive are stored at:

```
WP File Manager > wp-content > plugins > simple-history > loggers
```

Each logger extends from the SimpleLogger class to gain access to the SimpleLogger class members while achieving the overall functionality of the plugin. From which, a slug object is called which appears to be unused on the surface. However, since slugs are used to return references, it's likely that the slug object is being called by the superclass to acknowledge which objects are being called and what update category is being used.

- Slug Definition - Basically, a slug is an id reference for the hyperlink information. Typical usage allows the slug to take the place of the post URL ending to provide clarity to the user.

Furthermore, each class appears to use a **getInfo** function that, through the use of an array, returns a collection of strings/information obtained by various method calls throughout the script created.

Redacted File Contents Push to GitHub

The logs from SimpleHistory are stored within the database in CPanel in the phpAdmin section. This means that a query can be made to grab the data from the table, placed into a string, and sent to a text file for simplicity.

After considerable time and effort, here is a script that compiles the logs into a legible format for the text file to be stored with.

```
<?php
/*
 * Created by Ron F on 7/8/2020
 * Modified by Josh Greenert on 7/8/2020
 *
 * This script will be run by a cron to search for records not
updated in the simple_history_git
 * table to place in a text file to send to Github.  Afterwards, to
limit space the file created
 * will be deleted afterwards.
 */
include('../wp-includes/var/variables.php');

// Create the text file and string object to be used to pass
information.
$shafile = "sha.txt";
$filename = "";

// Create date folder constructs.
$year = "";
$month = "";
$day = "";
$time = "";
```

```
// Open file for edits and place sha in sha.txt file.
$shlstring = "";

//Create the connection to get data from the tables.
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create flag to update when no file returns.
$flag = true;

// Create query that only pulls files if not found in the history
reference table.
$sql1 = "SELECT wp_simple_history.id, wp_simple_history.date,
wp_simple_history.message FROM wp_simple_history where id not in
(select historyid from wp_simple_history_git) order by
wp_simple_history.date asc";
$result1 = $conn->query($sql1);

while($row1 = $result1->fetch_assoc()) {
     $historyid = $row1["id"];
     $eventdate = $row1["date"];
     $message = $row1["message"];

     // Instantiating year, month, and date strings.
     $year = date('Y', strtotime($eventdate));
     $month = date('m', strtotime($eventdate));
     $day = date('d', strtotime($eventdate));
     $time = date('h:i:s', strtotime($eventdate));

     // Use regular expression to change spaces to underscores.
     $message = preg_replace('/ /', '_', $message);

     // Check the length of the message to strip ending message if
too long.
     strlen($message) > 200 ? $message = substr($message, 0, 200) :
null;

     // Inside each record pulled, set the items up to add to the
string.
     $shlstring = $time . "_" . $message . ".txt";


    // Add all records pulled into the simple_history_git table for
reference on next execution.
    $addrecord = "insert into wp_simple_history_git (historyid)
values ($historyid)";
    if ($conn->query($addrecord) === TRUE) {
    } else {
```

```
    //  echo "Error: " . $addrecord . "<br>" . $conn->error;
    }



    // Check if the filename is empty and return.
    if($shlstring == ""){
        $flag = false;
        break;
    }

    // Create the file name as the string, open it to create it, then
close it.
    $filename = $shlstring;

    $sqlfilename = fopen($filename, "w") or die("Unable to open txt
file!");
    fclose($sqlfilename);

    // Put sha code into the file and close it.
    $shatxtfile = fopen($shafile, "w") or die("Unable to open sha
file!");
    $sha = sha1_file($filename);
    fwrite($shatxtfile, $sha);
    fclose($shatxtfile);

    // https://github.com/joshdangdesign/updateCollection/tree
/simplehistorylogs
    // This array contains the sha, committer, and converts the new
filename to a base64 encoded object.
    $data_git = array(
        'sha'=>file_get_contents($shafile),
        'message'=>'updated on ' . date('d-m-Y'),
        'content'=> base64_encode($filename),
        'committer'=> array(
            'name'=>'jgreenert+automation',
            'email' => 'jgreenert+automation@dang-designs.com')
    );

    // This portion converts the previous array to a json encoded
object to be able to
    // send through GitHub API and the curl sends it over.
    $data_string_git = json_encode($data_git);

    $ch_git = curl_init();
    curl_setopt($ch_git, CURLOPT_URL, 'https://api.github.com/repos
/dangdesigns/UpdateCollection/contents/360twin/simplehistorylogs/' .
$year . "/" . $month . "/" . $day . "/" . $filename);
    curl_setopt($ch_git, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch_git, CURLOPT_CUSTOMREQUEST, "PUT");
```

```
    curl_setopt($ch_git, CURLOPT_POSTFIELDS, $data_string_git);
    curl_setopt($ch_git, CURLOPT_HTTPHEADER, array(
        'User-Agent: josh',
        'Authorization: token
828fd75c682d477f6a0c9fa1ac9e327f78995aec'
    ));

    $result_git = curl_exec($ch_git);

    // Close the URL contents.
    curl_close($ch_git);

    // Delete the file that was used.
    unlink($filename);

    // API only allows 60 records to be updated per hour. Moreover,
the API detects abuse if done too quickly.
    sleep(5);

}


?>
```

After realizing the file would only send over the name, and not the file contents, attempting to take new direction to upload to GitHub file name as the most recent change. Overall, the content should look similar to the script listed below.

- Cleanup was added to ensure the files don't stay within the folder and are removed after use. Each file, when not already listed in the history, should upload to Github.
- The filename will be empty if the query has no new data within it that meets the criteria.
- Still need to test but this should solve the problem.

```
<?php
/*
 * Created by Ron F on 7/8/2020
 * Modified by Josh Greenert on 7/8/2020
 *
 * This script will be run by a cron to search for records not updated
in the simple_history_git
 * table to place in a text file to send to Github.  Afterwards, to
limit space the file created
 * will be deleted afterwards.
 */
include('../wp-includes/var/variables.php');

// Create the text file and string object to be used to pass
information.
$shafile = "sha.txt";
$filename = "";
```

```
// Create date folder constructs.
$year = "";
$month = "";
$day = "";
$time = "";

// Open file for edits and place sha in sha.txt file.
$shlstring = "";

//Create the connection to get data from the tables.
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create flag to update when no file returns.
$flag = true;

// Create query that only pulls files if not found in the history
reference table.
$sql1 = "SELECT wp_simple_history.id, wp_simple_history.date,
wp_simple_history.message FROM wp_simple_history where id not in
(select historyid from wp_simple_history_git) order by
wp_simple_history.date asc";
$result1 = $conn->query($sql1);

while($row1 = $result1->fetch_assoc()) {
     $historyid = $row1["id"];
     $eventdate = $row1["date"];
     $message = $row1["message"];

     // Instantiating year, month, and date strings.
     $year = date('Y', strtotime($eventdate));
     $month = date('m', strtotime($eventdate));
     $day = date('d', strtotime($eventdate));
     $time = date('h:i:s', strtotime($eventdate));

     // Use regular expression to change spaces to underscores.
     $message = preg_replace('/ /', '_', $message);

     // Check the length of the message to strip ending message if too
long.
     strlen($message) > 200 ? $message = substr($message, 0, 200) :
null;

     // Inside each record pulled, set the items up to add to the
string.
     $shlstring = $time . "_" . $message . ".txt";

    // Check if the filename is empty and return.
```

```
    if($shlstring == ""){
        $flag = false;
        break;
    }

    // Create the file name as the string, open it to create it, then
close it.
    $filename = $shlstring;

    $sqlfilename = fopen($filename, "w") or die("Unable to open txt
file!");
    fclose($sqlfilename);

    // Put sha code into the file and close it.
    $shatxtfile = fopen($shafile, "w") or die("Unable to open sha
file!");
    $sha = sha1_file($filename);
    fwrite($shatxtfile, $sha);
    fclose($shatxtfile);

    // https://github.com/joshdangdesign/updateCollection/tree
/simplehistorylogs
    // This array contains the sha, committer, and converts the new
filename to a base64 encoded object.
    $data_git = array(
        'sha'=>file_get_contents($shafile),
        'message'=>'updated on ' . date('d-m-Y'),
        'content'=> base64_encode($filename),
        'committer'=> array(
            'name'=>'jgreenert+automation',
            'email' => 'jgreenert+automation@dang-designs.com')
    );

    // This portion converts the previous array to a json encoded
object to be able to
    // send through GitHub API and the curl sends it over.
    $data_string_git = json_encode($data_git);

    $ch_git = curl_init();
    curl_setopt($ch_git, CURLOPT_URL, 'https://api.github.com/repos
/dangdesigns/UpdateCollection/contents/360twin/live/simplehistorylogs/'
. $year . "/" . $month . "/" . $day . "/" . $filename);
    curl_setopt($ch_git, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch_git, CURLOPT_CUSTOMREQUEST, "PUT");
    curl_setopt($ch_git, CURLOPT_POSTFIELDS, $data_string_git);
    curl_setopt($ch_git, CURLOPT_HTTPHEADER, array(
        'User-Agent: jgreenert+automation',
        'Authorization: token 828fd75c682d477f6a0c9fa1ac9e327f78995aec'
    ));
```

```
    $result_git = curl_exec($ch_git);

    // Close the URL contents.
    curl_close($ch_git);

    // Delete the file that was used.
    unlink($filename);

    // Add all records pulled into the simple_history_git table for
reference on next execution.
    $addrecord = "insert into wp_simple_history_git (historyid) values
($historyid)";
    if ($conn->query($addrecord) === TRUE) {
    } else {
    //  echo "Error: " . $addrecord . "<br>" . $conn->error;
    }

    // API only allows 60 records to be updated per hour. Moreover, the
API detects abuse if done too quickly.
    sleep(5);
}
// closing connection
mysqli_close($conn);

?>
```

References
https://github.community/t/how-to-upload-a-file-using-github-api/14031

Getting Joomla Logs

Getting Magento Logs

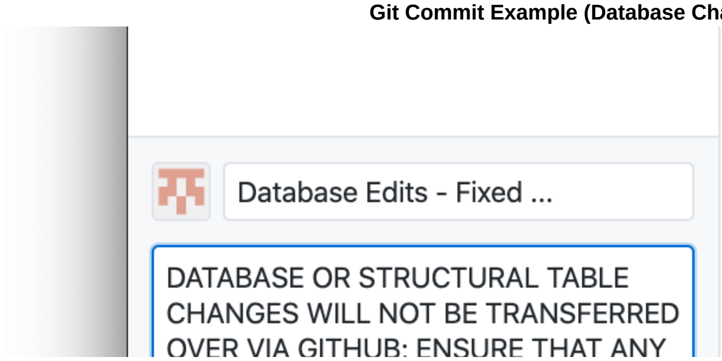
Getting Code Logs

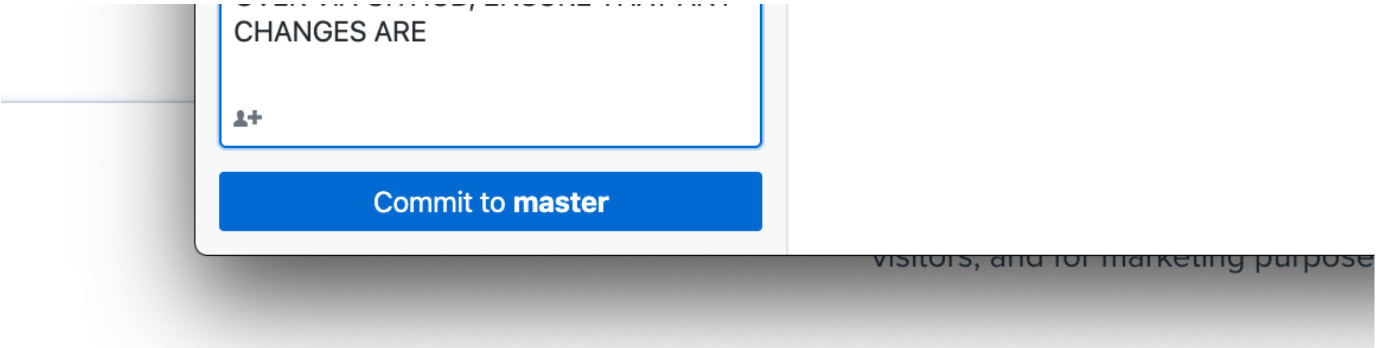
Setting Up Local MAMP Environment

This document serves as a guide to show the steps on how to set up a MAMP environment for Joomla or WordPress sites for the express purpose of getting branches pushed and pulled from/to GitHub.

**** ANY DATABASE OR STRUCTURAL TABLE CHANGES WILL NOT BE TRANSFERRED OVER VIA GITHUB; ENSURE THAT ANY CHANGES ARE COMMUNICATED PROPERLY AND UPDATED PRIOR TO COMMITTING TO THE MASTER BRANCH ****

Git Commit Example (Database Change Instance)





1. **Download MAMP/MAMP Pro > {**
    a. Visit this website to download the package from MAMP's website
        i. https://www.mamp.info/en/downloads/
    b. Follow the startup package installer directions **}**
2. **Download GitHub Repo from GitHub master branch > {**
    a. If the website has not been created yet, create the website up to where the database credentials are set up and the site is able to be accessed, then create a repository and ensure all files for the site are uploaded to the repository.
        i. *Then create a clone of the repository to your system.*
    b. **If the GitHub Repo has not been downloaded already, follow these steps.**
        i. Once a repo has been shared to you, you must clone the repository using the GitHub Desktop Application
        ii. Inside the GitHub application, on the top left side, to the right of Filter, select **add > Clone a Repository**
        iii. Copy and paste the URL from the shared repository into the URL option listed.
            1. *There are three options that will be offered (GitHub.com, GitHub Enterprise Server, and URL)*
            2. *Obviously, use the one that meets your needs the best if performing another type of cloning.*
        iv. Update the local path to the GitHub file destination
            1. *Most likely /Users/[YOU]/Git/[WEBSITE_NAME]*
    c. **If the GitHub Repo has already been download, follow these steps.**
        i. Ensure that the Current Repository listed is the one you plan to make changes to.
            1. *located on the top-left of the GitHub application*
        ii. Set the current branch to the master branch
            1. *This ensures that the files that will be received are the most accurate and updated.*
            2. *This is on the right of the current repository.*
        iii. To the left of the current branch, press the button stating either Fetch Origin or Pull Origin
            1. *This will update the files in your path to ensure they are all updated to what the master branch has.*
            2. *** If you have any updates within the files that will be in conflict with the update, they will be overridden by the master branch possibly *** **}**
3. **Download the database from the remote website's PHPMyAdmin panel (export) > {**
    a. Navigate to the C-panel for the website you will be making updates to and sign in.
    b. Select PHPMyAdmin
    c. Select the Export option at the top of the page
    d. Change the export method from 'Quick' to 'Custom'
    e. In the Output section, change compression to 'zipped'
    f. Press 'Go'
    g. The file should download all databases needed for setup; make sure to save them in a path you'll remember as we will need those soon. **}**
4. **Update MAMP ports to ensure matching for all website configuration files > {**
    a. Open the MAMP application
    b. With MAMP in focus, select the MAMP option (next to the Apple icon on the top left of the screen)
    c. Select Preferences
    d. Navigate to Ports
    e. Select '*Set Web & MySQL ports to 80 & 3306'*
    f. Press 'OK' and the servers will restart. **}**
5. **Set the Document Root folder for the web server > {**
    a. Within the MAMP Preferences, navigate to the *Web Server* tab
    b. To the right of Document Root: press '*Select*'
    c. Set the path to [USER] > Git > [WEBSITE_FOLDER]
    d. Press '*OK*' and the servers will restart once more, now pointing to right destination
        i. *This step will need to be performed anytime a local environment will need to be change for a different website.*
    e. Once the servers restart, MAMP should automatically open the MAMP startup webpage. **}**
6. **Import the database to the local system to ensure all data matches for local connection > {**
    a. Within the GitHub repository, pull up the '**wp-config.php**' (WordPress) file or the '**configuration.php**' (Joomla) file
        i. *The information within this file will be used as a template to match the local database name, user, and user password.*
    b. From the MAMP startup page, open the PHPMyAdmin page
        i. *This will be set for your local machine and will require updates*
        ii. *If you have closed the MAMP startup page, no worries. The page can be accessed here:*
            1. http://localhost/MAMP/phpmyadmin.php?lang=en
    c. On the top menu ribbon in PHPMyAdmin, select 'Import'

    d. Choose the zip file that was downloaded in step 3
    e. Press ‘Go’
        i. *This will update your databases to match what the remote site will have.* **}**
7. **Create the user account to access the database information > {**
    a. On the top menu ribbon, select ‘User accounts’
    b. If the user is not listed, select ‘Add user account’
        i. *If the user is already listed, check the information from the* ***wp-config.php*** *(WordPress) file or the ‘****configuration.php’*** *(Joomla) file to ensure that the data matches and move on to step 8.*
    c. Copy and paste the User name to the Username field from the config file.
    d. Copy and paste the password to the password field from the config file.
    e. For global privileges, place a check in the ‘Check All’ checkbox.
    f. Press ‘Go’ **}**
8. **Follow the next steps based on the website type > {**
    a. Setting Up WordPress Site
    b. Setting Up Joomla Site
    c. (Magento Site Info Coming Soon!)

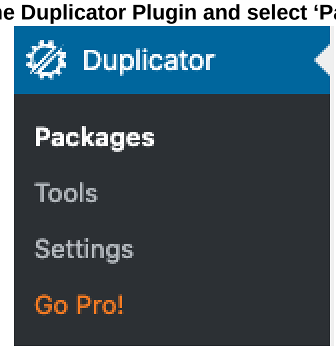
Setting Up WordPress Site

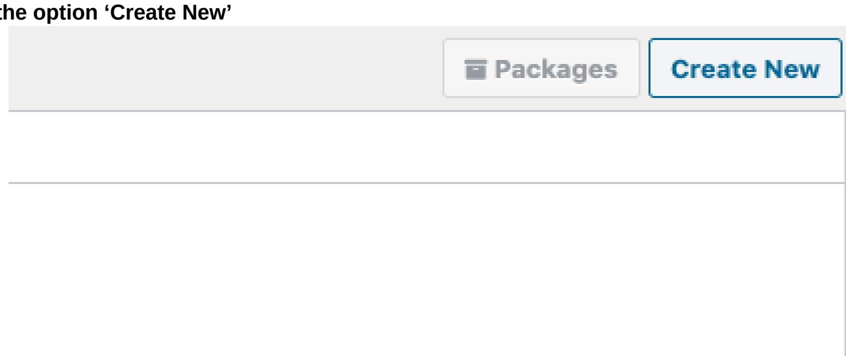
This page will provide directions on how to set up a local WordPress site from the remote site for testing and updates. Granted, this process deviates from the traditional GitHub process, it has proven to be 100% successful on each site tested. The process uses the Duplicator plugin created for WordPress (Not available for Joomla, unfortunately).

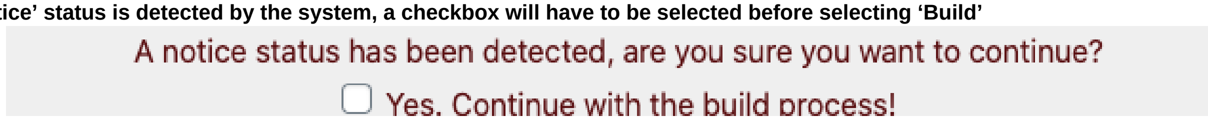
The following steps assume the following:

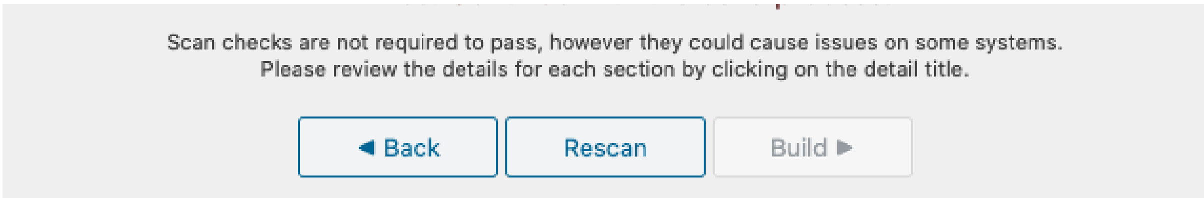
- *The local MAMP product is installed and ready for minor configuration.*
- *The database file has been imported into the phpmyadmin locally from the remote*
- *The user for the database was added to match that of the remote data.*

Final Steps

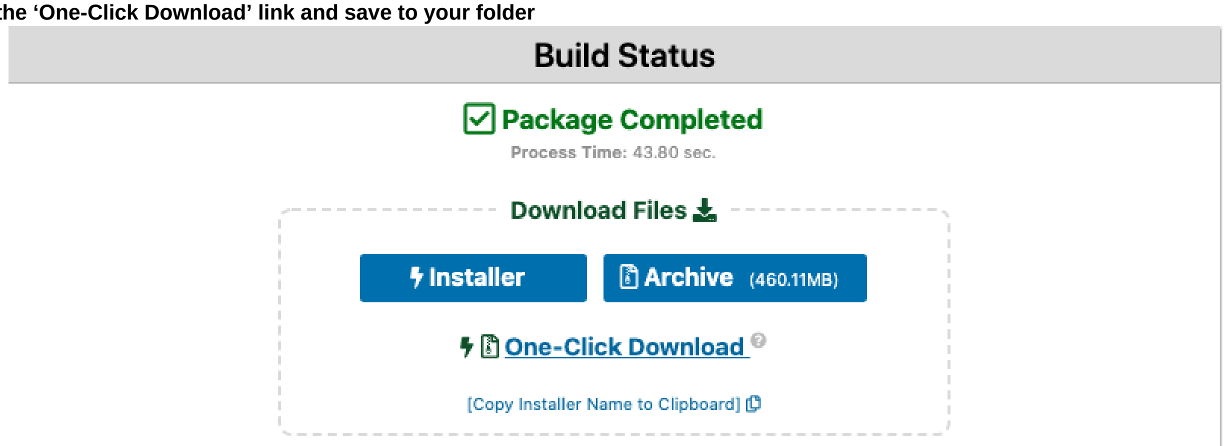
1. **Navigate to the WordPress site and log in**
2. **Find the Duplicator Plugin and select ‘Packages’**
    a. 

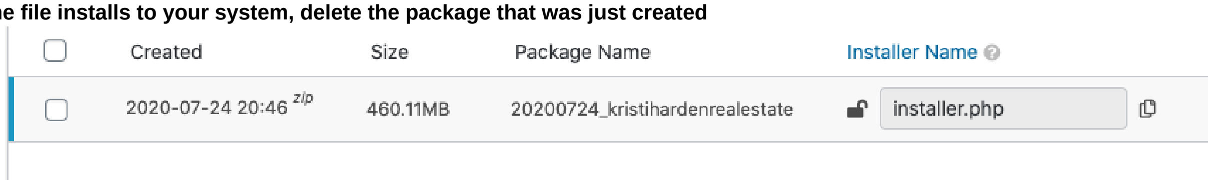
3. **Select the option ‘Create New’**
    a. 

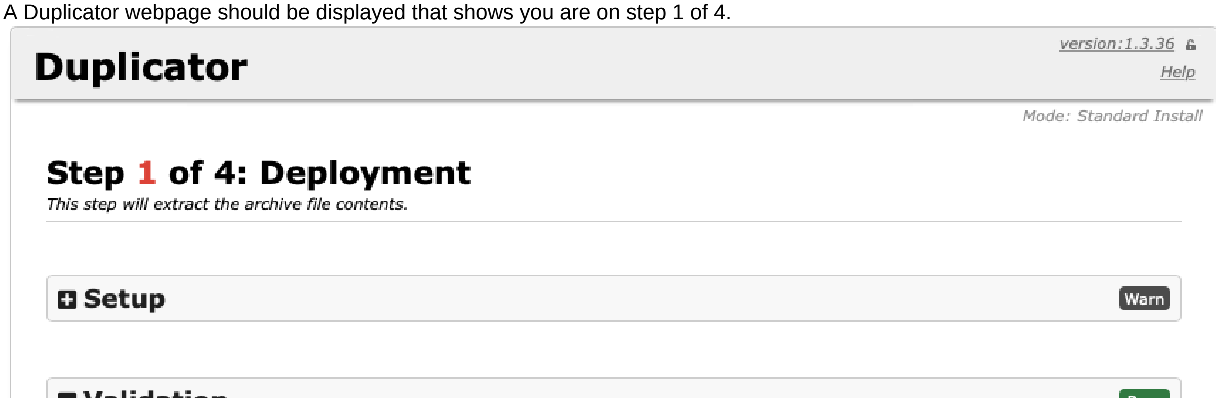
    b. *This will be located on the far right-hand side of the screen. All packages should be removed after installed to the local system as their respective weight and impact on the site has not been reviewed.*
4. **Leave everything default except for the ‘Host’ and ‘Host Port’ fields under the ‘Installer’ header**
    a. In the ‘Host’ field, type ‘localhost’
    b. In the ‘Host Port’ field, type ‘3306’
5. **Select ‘Next’ at the bottom right-hand side of the page**
    a. *A quick scan will complete of the site when this button is pressed.*
6. **If a ‘Notice’ status is detected by the system, a checkbox will have to be selected before selecting ‘Build’**
    a. 

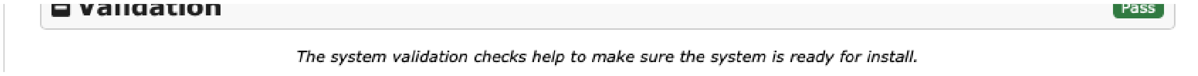


b. *The package will build its required content and create an installer.php file along with an archive file. Both of these files will be necessary!*

7. **Create a folder for these contents to be placed into (Not the GitHub)**
   a. *This information doesn't pull from GitHub and, therefore, doesn't need to be mixed with it and cause problems. A good suggestion would be to have a file called 'localtesting' inside of the GitHub folder location so that file changes can be easily passed over to the GitHub folder needed (still need to discuss best practice on this)*
8. **Select the 'One-Click Download' link and save to your folder**
   a. 

   b. *Note: This will ask you to select a folder twice*
   c. *Depending on the size of the site, the zip archive file could take a while. 360dev took 15 minutes, for example.*
9. **Once the file installs to your system, delete the package that was just created**
   a. 

   b. Select the Duplicator Packages again
   c. Select the checkbox on the far left of the package created.
   d. Change the dropdown that states 'Bulk Actions' to 'delete' and press 'Apply'
   e. Press 'Ok' when prompted
10. **Open the folder where the zip archive was stored and extract it by double-clicking on it**
11. **Delete the extra zip file and copy all the contents from the folder into the main subdirectory with your installer.php file**
12. **Set MAMP to point to the right location**
   a. Provide focus to the MAMP application by clicking on it
   b. In the top-left corner of the screen, click MAMP and select 'Preferences'
   c. Within the MAMP application, select 'Web Server'
   d. Set the document root location to the folder created in step 7
   e. Press 'Ok'
13. **Pull up your development browser and clear the cache**
   a. If your cache refers to the original placement, the localhost may pull up previously viewed information.
14. **In your development browser, enter the following URL:**
   a. http://localhost/installer.php
   b. A Duplicator webpage should be displayed that shows you are on step 1 of 4.
   c. 



15. **Select the checkbox for term & notices and select ‘Next’**
16. **Enter the fields requested for step 2 of 4:**
    a. host = 'localhost'
    b. database = [NAME_OF_YOUR_DATABASE]
    c. user = [NAME_OF_YOUR_DATABASE_USERNAME]
    d. password = [NAME_OF_YOUR_DATABASE_USER_PASSWORD]
        i. *You may need to pull up your wp-config.php file to ensure this information is entered correctly.*
17. **At the bottom of the page, select ‘Test Database’**
    a. This ensures that the information was entered correctly and the installer checks for any issues to report.
18. **Select ‘Next’ at the bottom of the page**
19. **An Install Confirmation input box will appear; press ‘OK’**
20. **Accept the defaults and select ‘Next’**
21. **Log into the WordPress Admin panel to complete the configuration**
    a. Select the ‘Admin Login’ option while keeping the checkbox that states it will auto delete installer files after login selected
    b. Use the WordPress credentials to authenticate and the localhost will connect and display that the installer dependencies have been removed
22. **You’re Done!**
    a. **NOTE**: For larger sites that require a bit more plugins, allow another 3 - 5 minutes for the site to fully have all of the content.
    b. **NOTE:** The 'WPBakery Page Builder' plugin may be disabled locally to ensure that everything is set up correctly. Make sure that this is activated before jumping into the site.

Setting Up Joomla Site

This page will provide directions on how to set up a local Joomla site for development purposes. Another step may be missing from the process to obtain the content that is not explicitly sent over. Still testing to ensure all steps are fully proven and documented.

WARNING **This process has been proven to have issues if files are stored on the Joomla site directly for content.**

1. For **Joomla**, the steps require a bit more detail.
    a. If any attempts to open the Joomla site have already been made before following these steps, make sure to clear you cache on your browser or you will receive the same results!
    b. Navigate to the directory where the files are stored and ensure that hidden files are visible.
        i. Press CTRL + SHFT + . (that’s a period)
        ii. If this doesn’t work, open the terminal and type the following command:

```
1. defaults write com.apple.finder AppleShowAllFiles -
   bool true
```

    c. Open the .htaccess document in your preferred code editor
        i. Comment out this line with a pound sign ('#')
            1. # AddHandler application/x-httpd-ea-php72 .php .php7 .phtml
                a. This piece tells the compiler to download the proper php library even if it is present.
            2. Save the file and close it.
    d. Open the file configuration.php
        i. Modify the public $force_ssl from this to that.

```
ii. public $force_ssl = '2';
```

```
iii. public $force_ssl = '0';
```

        iv. Update the file paths for the log_path and tmp_path
            1. Should look something like this:

2.
```
public $log_path = '/home/hardenoffroad/public_html/administrator/logs';
public $tmp_path = '/home/hardenoffroad/public_html/tmp';
```

3. Update these to:

4.
```
public $log_path = '/Users/JoshG/Git/harden-offroad/administrator/logs';
public $tmp_path = '/Users/JoshG/Git/harden-offroad/tmp';
```

    a. Without this update, you won't be able to log into the administrator section for the website.

5. Finally, comment out the live site path to ensure all updates and directions make local updates only.
    a. Line should look similar to this:

    b.
```
public $live_site = 'https://harden-offroad.com';
```

v. Save the file and close it. **}**

1. **Enter the localhost address in your URL and the webpage should display > {**
    a. http://localhost/ **}**
2. **You're Done!**

Error Table With Corrections			
Error Description	**Picture**	**Solution**	**Comments**

Website for local migration pulls up blank page with "**Error**" being displayed.		**This could be a few issues, so it's wise to rule out the main ones.**
		• **Either the database name, database user, or database password are incorrect**.
		  • Check these within the configuration file to ensure accuracy
		• **The database content is missing.**
		  • Open the file and ensure that the tables are present for the database
		• **A path in the .htaccess file needs to be commented out.**
		  • `php_value session.save_path "/var/cpanel/php/sessions/ea-php73"`
		  • This should appear twice in the .htaccess file (comment out with '#')
			Any other issues like this can be viewed easier by updating the configuration.php file field where `public $error_reporting = 'none';` by changing the value 'none' to 'maximum'. This will show you more information as to what the root error is.

Setting Up A Batch File To Transfer Backups And Delete Them

This page defines the process used to automate the storage of database backups from Fishbowl to the Google Cloud Storage Bucket. Essentially, a set of batch files are used to run a gsutil command which transfers the data over to the “fishbowl-storage” bucket, then the file that was transferred over is removed from the system. This effort is intended to promote security of data while ensuring costs remain low for storage on a virtual machine.

1. **Set up the Google Cloud Storage Bucket**
    a. Information for the process is found here
        i. SunFlora - Setting Up Cloud Storage Bucket From PHP Script
            1. Unfortunately, the size of the database has increased beyond the amount allowed for php to send the file over; moreover, the files are not able to be broken into pieces and appended to one another. Thus, a new solution is born.
2. **Ensure that the proper access is available for the user inside of the VM**
    a. If the bucket and the VM are on different projects, then use the following option instead in the command prompt:
        i. Ensure that Google Cloud SDK is installed
        ii.
        ```
        gcloud auth login
        ```
            1. This requires the user to login that has the appropriate permissions regardless of the project.
3. **Confirm that files can be passed to the Google Cloud Storage Bucket**
4. **Set up the batch files to utilize the gsutil command to pass the large files over**
    a. One batch file is needed to copy the file over, and another to delete the file off of the system that remains; this must be done in two batches since the execution of the gsutil command overrides the batch file and forces it to quit early.
        i. **Batch File (Transfers to Bucket)**

```
@ECHO OFF
:: Created by Joshua Greenert on 2/24/2021
:: This batch file will be used to run the
SendToGGCStorageBucket.php file.
TITLE Google Storage Backup Transfer
ECHO
==========================================================
====================
ECHO Starting batch file: Sending database backup to
Google Cloud Storage Bucket...
ECHO
==========================================================
====================
set filedate=_%date:~-4%_%date:~4,2%_%date:~7,2%
mysqldump --user root --password="password"
sunflora_inventory > "C:/Program Files/Fishbowl/data
/backups/sunflora_inventory%filedate%.sql"
gsutil cp "C:/Program Files/Fishbowl/data/backups/*.sql"
gs://fishbowl-storage
ECHO
==========================================================
====================
ECHO Batch process completed successfully!
ECHO
==========================================================
====================
```

        ii. **Batch File (Cleanup)**

```
@ECHO OFF
:: Created by Joshua Greenert on 2/24/2021
:: This batch file will be used to delete the file after
sent over to GC Bucket.
TITLE Database Backup Cleanup
ECHO
==========================================================
====================
ECHO Starting batch file: Removing extra database backups
leftover...
ECHO
==========================================================
====================
del "../../../../Program Files/Fishbowl/data/backups/*.
sql"
ECHO
==========================================================
====================
ECHO Batch process completed successfully!
ECHO
==========================================================
====================
```

5. **Set up the task scheduler to run the batch files**
    a. Navigate to the windows machine and search windows for 'Task Scheduler'
    b. Expand the task scheduler library and create a new folder for the batch files
        i. *Not necessary but good practice
        ii. Select the 'Task Scheduler Library' and right-click to select new Folder
    c. Right-click on the newly created folder and select 'Create Basic Task'
    d. Enter a name and description that outlines what the task will do
    e. Select Daily
    f. Set the time to run for early in the morning (3:00 AM)
    g. Set the action to 'Start a program'
    h. Select the batch file from the Browse file selector and press finish
6. **Confirm the task worked and executed as expected by following up**
7. **You're Done!**

References	
Link	**Description**
https://cloud.google.com/functions/docs/tutorials/storage	Google Storage using gsutils

Setting Up A Cron Job

This page details how to set up a cron job within the c-panel to ensure that a script file is ran consistently based on time-driven triggers.

1. **Log into the c-panel for the site > {**
    a. *Make sure that the site file manager already includes the php script file to execute.*
        i. You'll also need to know the path to the file as well. **}**
2. **Navigate to the Advanced section select the icon labeled 'Cron Jobs'**
3. **Update the common settings to the needed duration for the time-based trigger > {**
    a. ignore the cron email section as this will send an email every time that the cron job runs
        i. If this is a file that updates constantly, this will just be annoying
    b. The main options are available in a dropdown menu once you select '*Common Settings*'; so choose the one that best suits your need. **}**
4. **Enter the command to execute the script file > {**
    a. If you're using a PHP script then your command to enter will look similar to this:

i. `php /home/three60twin/public_html/simple-history-git/simplehistorypushtogit.php`

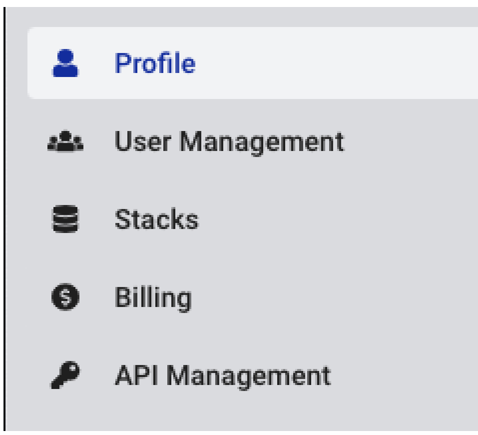
1. **Note:** the language used must be called before the file path. **}**

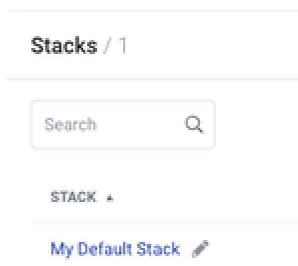
5. **Select 'Add New Cron Job'**
6. **You're Done!**

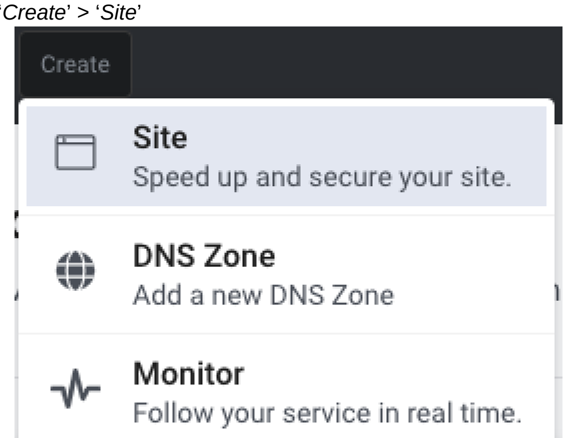
Setting Up A CDN For Joomla Site

This page will cover the essential steps to improve speed performance through the use of a CDN. There are certainly other details to consider to make these improvements holistically better, but those are being determined as we test and discover.

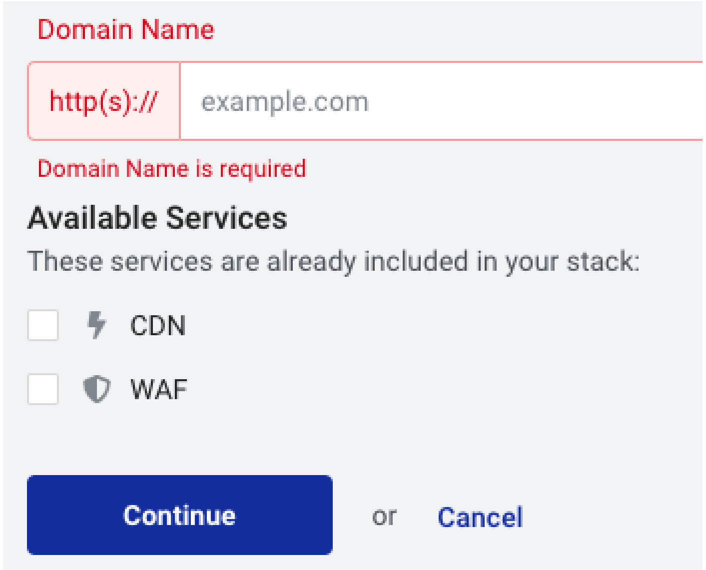
StackPath Credentials	
Username	Password
dev@dang-designs.com	9420LazyLane#CDN

1. **Add the website to StackPath > {**
   a. Log into StackPath
   b. Select '*Stacks*'
      i.


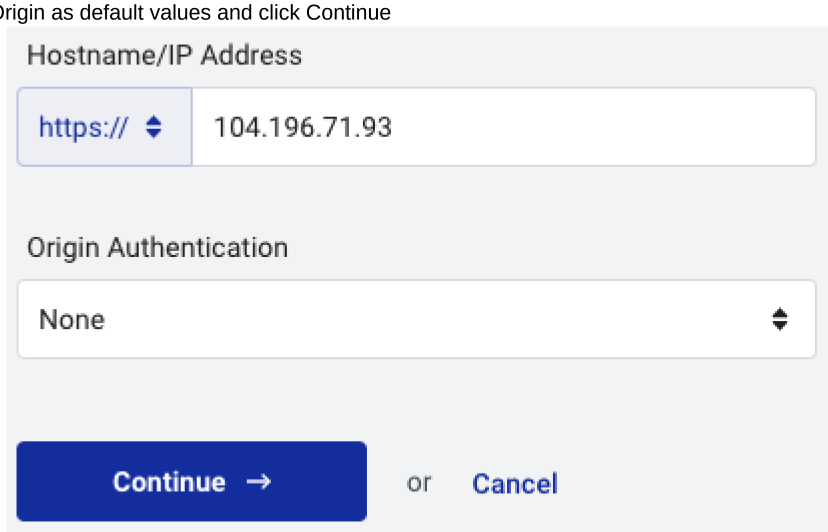
   c. Select '*My Default Stack*'
      i.


   d. Select '*Create*' > '*Site*'
      i.


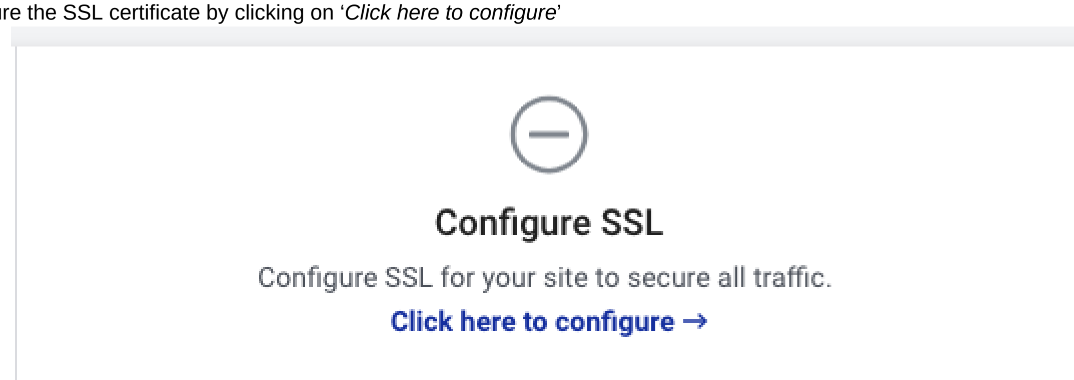
e. Enter the domain in the '*Domain Name*' field and select the checkboxes for **CDN** and **WAF**, then click 'Continue'

   i. 

f. Leave Origin as default values and click Continue

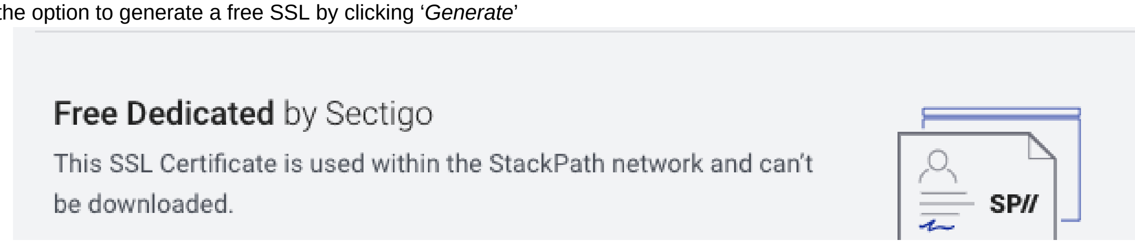
   i. 

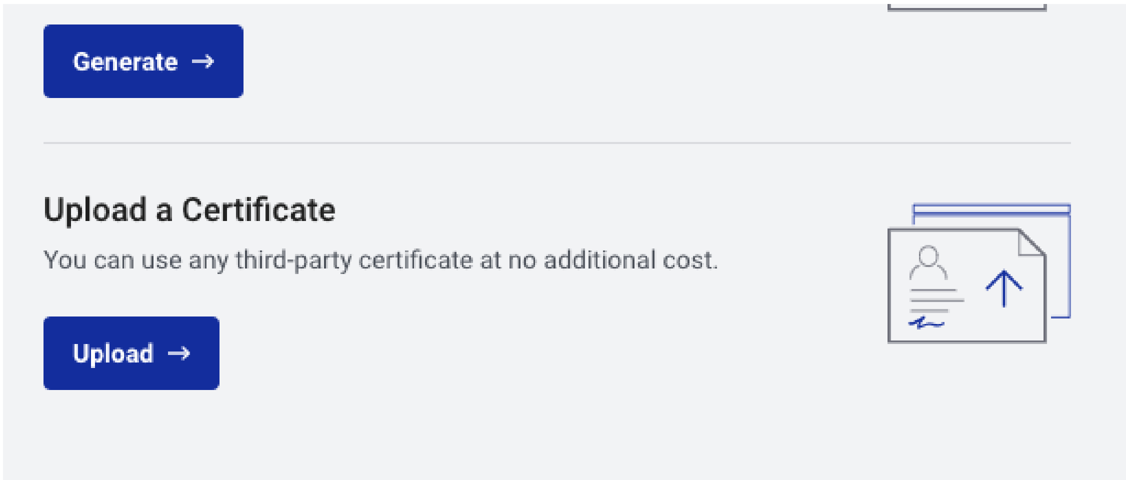
g. Configure the SSL certificate by clicking on '*Click here to configure*'

   i. 

h. Click 'Create Certificate'

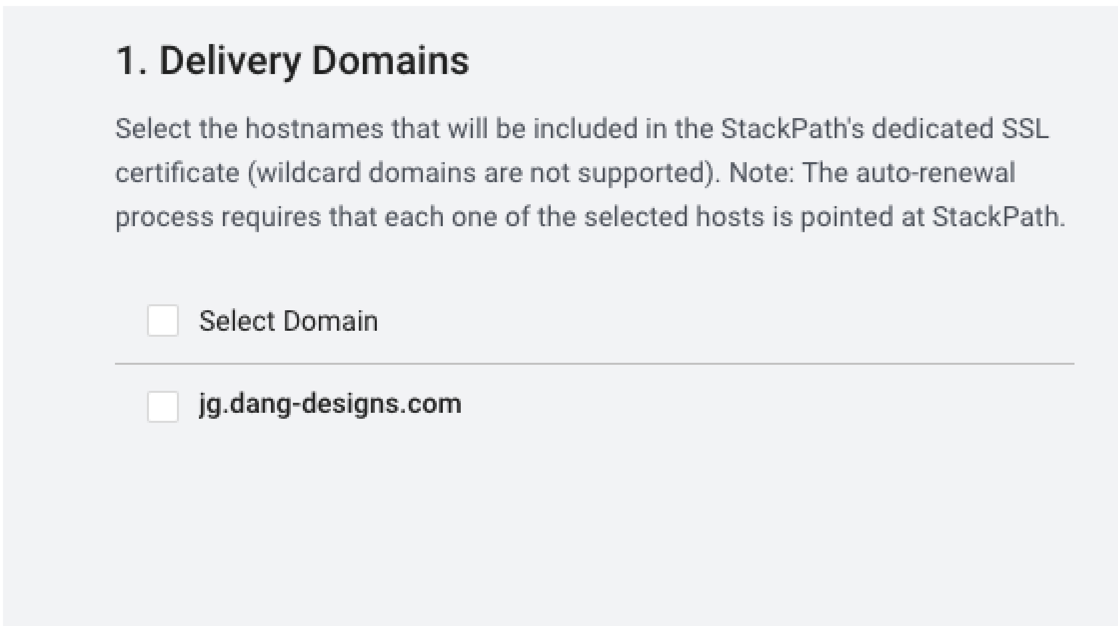
i. Select the option to generate a free SSL by clicking '*Generate*'

   i. 



j. Select all domains and select ‘*Continue to Validation*’

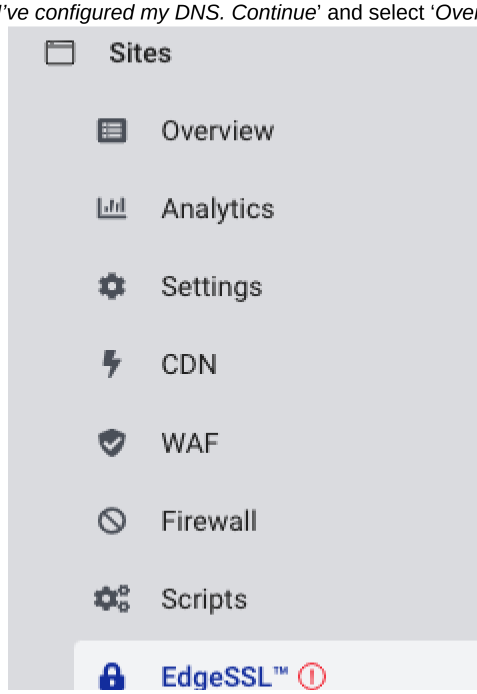
i.



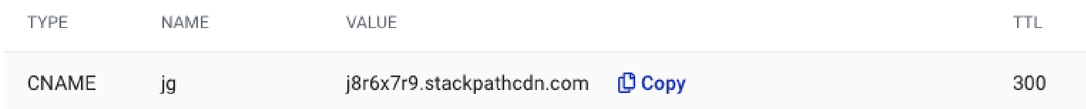
k. Select ‘*HTTP Request*’ radio button and select ‘*Create Certificate*’

l. Select ‘*I’ve configured my DNS. Continue*’ and select ‘*Overview*’ the left-hand panel

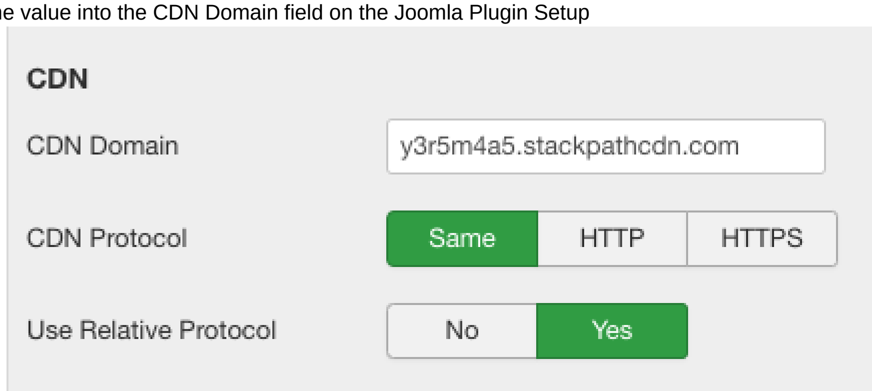
i.



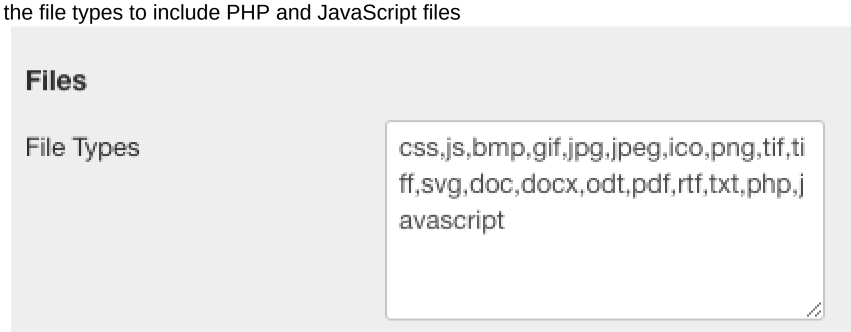
    m. Keep this page up for when we will need it for the plugin. **}**
2. **Install the CDN for Joomla! plugin and set it up > {**
    a. Download the CDN for Joomla! plugin on the shared drive
        i. 01-DANG-Work > 00-Stock-&-Software-Related > Joomla - Extensions > Plug - ins > cdnforjoomla-v6.1.5-pro.zip
        ii. https://drive.google.com/drive/u/0/folders/1tijDTVJEjvZM5SpdbvrhieTwV31Z17nI
    b. Log into the Joomla site to install the plugin
    c. Select Extensions > Manage > Install
    d. Under 'Upload Package File' select 'Or Browse for file' and upload the CDN for Joomla plugin
    e. Now that the plugin is installed, select Extensions > Plugins
    f. Pull up the plugin and select it
        i. Should be named 'System - Regular Labs - CDN for Joomla!'
        ii. *Once inside the plugin, make sure to switch from the 'Plugin' page to the 'Setup' page*
    g. From the StackPath Overview page, copy the StackPath site link
        i. Should look something like this:
        ii.



    h. Paste the value into the CDN Domain field on the Joomla Plugin Setup
        i.



    i. Update the file types to include PHP and JavaScript files
        i.



    j. Leave all other defaults the same and select 'Save & Close' at the top of the page **}**
3. **Delete the cache > {**
    a. On the Joomla site, select System > Clear Cache
    b. Select 'Delete All' **}**
4. **You're Done!**

Setting Up WordPress/Joomla Site On Alternative Domain

This page will provide steps on how to set up a WordPress site on an alternative domain (local isn't working effectively) and will discuss how to navigate around issues encountered.

1. **Open the C-Panel for the live site**
2. **Download the tables from the live database > {**
    a. Once inside phpMyAdmin, select the database by clicking on the database
    b. With the database selected — and the tables showing — select export, select custom, ensure the compression is zip, and press 'Go'.
    c. Save the files to a familiar location **}**
3. **Download a copy of the files from File Manager > {**
    a. Log into the C-Panel of the live site.

    b. Enter the Public_HTML folder
    c. Select all files and compress them
    d. Rename the file to the site name (or something to help understand exactly what it is)
    e. Download the file to use for later
    f. Delete the zip file just created from the File Manager
        i. Ensure that the file does not go into .Trash and purge it from the File Manager. **}**
4. **On the alt domain, create a database and user for the tables from the live database and add them > {**
    a. Log into the C-Panel for the alt domain
    b. Open the MySQL Databases
    c. Underneath 'Create New Database' create a database to hold the tables from the live site's database
        i. To make things easier, use the name of the database from the live site
            1. This won't cause issues as the name of the database will be concatentated with the prefix for the domain.
    d. Select Create Database
    e. Further down the page, create a user to access the database
        i. To make things easier, use the name of the database user from the live site
    f. Add the user, password, and select Create User
    g. Under Add User to Database, select the user just created and the database; select Add
    h. Open the phpMyAdmin and select the new database create
    i. With the database selected, choose Import and press 'Go' **}**
5. **On the alt domain, upload the files from the live site to the alt domain File Manager > {**
    a. Log into the C-Panel for the alt domain (if not already still within it)
    b. Open the File Manager and navigate to the Public_HTML folder
    c. If there are files present, ensure that the site isn't being tested already by checking the DANG-Sites sheet
        i. https://docs.google.com/spreadsheets/d/1WT_AT7jVpJ_4Mv3AACkmL4h0_PAU8_2qI5kD4Zkayxw/edit#gid=1124190019
        ii. If the files were not removed after a testing situation has been completed, simply delete the contents of the Public_HTML files
    d. Upload the zip file of the File Manager contents from the live site
        i. Extract the files and remove the zip file from the directory **}**
6. **Modify the database and files to match the current domain > {**
    a. **For Joomla, follow these steps**
        i. Update the configuration files to match the new location
            1. Under Public_HTML, open configuration.php
                a. This file stores the database connection credentials to update
                b. The data to update should be set to variables in a group called $user, $password, $db
                    i. *username, password, database_name (from left to right)*
                c. Update the log_path and tmp_path variables (*only necessary if you intend to log into backend as administrator user*)
                    i. Settings will have this look like the following:
                    ii.

```
        public $log_path = '/home/regional/public_html/logs';
        public $tmp_path = '/home/regional/public_html/tmp';
```

                    iii. Change that to this
                    iv.

```
        public $log_path = '/home/[NAME_OF_SITE]/public_html/logs';
        public $tmp_path = '/home/[NAME_OF_SITE]/public_html/tmp';
```

                    v. **NOTE:** It's best to use the home reference in the File Manager as a reference to see precisely what this should be.
                d. In some rare cases, the .htaccess file will need to be updated. To see what to look out for, check the following link: (Check the table at the bottom of the page referenced)
                    i. https://dangdesigns.atlassian.net/wiki/pages/resumedraft.action?draftId=179830812
    b. **For WordPress, follow these steps**
        i. Update the database files to line up with the new site
            1. Open the file within the database (inside of phpMyAdmin) called wp_options
                a. This may be named slightly differently if the WordPress site has a different prefix for the database files. However, the concept is still the same (XX_options) etc.
            2. The first two options should refer to the previous domain

        a. Update these two fields to match the domain that the testing site is on.
            i. EX: https://joshtestdomain.com
    ii. Update the files in File Manager to match the new site
        1. Within Public_HTML, open the **wp-config.php** file
            a. Inside of this file, update the database, username, and password to match what was created in Step 4. Save the file
            b. With some websites, the .htaccess file holds redirects to the live site. Since these behavior less naturally then the other options, test these and remove them only as needed.
                i. It's also possible that the redirects may be the result of the domain autoredirecting by default. If this is the case, use the solution listed below.
    iii. Update the .htaccess file to deny all traffic except for us
        1. Inside the root directory, open the .htaccess file
        2. Within the file, add the following code

a.
```
Order Deny,Allow
Deny from all
Allow from 47.205.11.142
```

7. **You're Done!**

Issues You May Encounter	
Issue	**Solution**

Domain is redirecting to live site	Go to **wp-config.php** file and include the statements below towards the top of the page.
	`define('WP_SITEURL', 'https://domain.com');`
	`define('WP_HOME', 'https://domain.com');`

Setup PayPal Auto Return for Website Payments

Settings location within the PayPal Account:
Account Settings > Website Payments > Website Preferences

Auto return for website payments

Auto return for website payments brings your buyers back to your website immediately after completing a payment. Auto return applies to PayPal website payments, including "buy now" payments, donations, subscriptions, and shopping cart payments.

Return URL requirements:

Enter the URL that will be used to redirect your customers after payment. This URL must meet the guidelines detailed below.

- According to our User Agreement, you must explain to the buyer on the page displayed by the return URL that the payment has been made and the transaction completed.
- You must explain on the page displayed by the Return URL that payment transaction details will be emailed to the buyer.
- Example: Thank you for your payment. Your transaction has been completed, and a receipt for your purchase has been emailed to you. Log into your PayPal account to view transaction details.

Auto return

ON
EXAMPLE: https://domain-name.com/checkout/order-received/?utm_nooverride=1

OFF
Auto Return will disable the Payment Data Transfer feature.

Payment data transfer (optional)

Payment data transfer allows you to receive notifications of successful payments as they are made. The use of payment data transfer depends on your system configuration and your Return URL. Please note that in order to use payment data transfer, you must turn on auto return.

Identity Token: xxxxxxxxxxxxxxxxxxxxxxx

Note: Most plugins will require your identity token to reference your account settings

Setting Up Windows Server 2019

This page is intended to give instructions on how to set up a Windows Server instance in Google Cloud Platforms. While the title is very broad, the information covered will support navigation for the instance and chrome Remote desktop installation only.

1. **Create the Windows Server Instance > {**
    a. [Provide a suitable name for the instance]
    b. [Choose the N2 series in the machine configuration input]
        i. *Other machine instances can be selected but offer a higher cost than necessary to complete the task at hand.*
        ii. *E2 instances are no longer supported for the option to add a GPU or display device.*
    c. [Select the machine type in the machine configuration input]
        i. *The standard default (e2-standard-2 (2 vCPU, 8 GB memory) is powerful enough for Fishbowl Server configuration, but higher power options are available*
    d. [Under the Boot Disk option select Change]
    e. [Under operating system, select "*Windows Server*"]
    f. [Under version select "*Windows Server 2019 Datacenter*"]
    g. [Leave the Boot disk type and Size (GB) fields as default and press "*Select*"]
    h. [Select the "*Create*" button at the bottom of the page] }
2. **Set the Password for the Windows Server Instance > {**
    a. [Go to **VM Instances** page]
    b. [Click on the Windows VM to change the password on]
    c. [On the **VM Instance Details** page, click "*Set Windows password*"]
    d. [In the "*Username*" field, enter the username to change the password for, or enter a new username to create a new user.]
    e. [Click "*Set*"] }
3. **Install the GCloud Command Line Tool OR Use Web-Directed Client > {**
    a. [First check that python library meets requirements by opening a terminal and typing "python -v"]
        i. *The supported versions require 3.5 to 3.7, and 2.7.9 or higher on the local machine*
        ii. *Google documentation states that there will be a move to Python 3 libraries exclusively soon*
        iii. https://ssh.cloud.google.com/cloudshell/editor?page=editor&_ga=2.240157888.296902771.1592834447-652029738.1591883601
        iv. *The GCloud Command line tool can also be accessed by going to VM Instances under the "*Connect*" column, then selecting "*View gcloud command*" in the SSH dropdown selection.
    b. [Enter the following command to download SDK tools to path]

i.
```
./google-cloud-sdk/install.sh
```

1. *Help information can be found using the following command.*

2.
```
./google-cloud-sdk/install.sh --help
```

    c. [Run the gcloud Init to initialize the SDK]

i.
```
./google-cloud-sdk/bin/gcloud init
```

        ii. *For additional assistance after this step, check out the how-to guides at the link below.*
        iii. https://cloud.google.com/sdk/docs/how-to }.
4. **Using GCloud Command Line, Configure Routes and Firewall Rules > {**
    a. [Create the allow egress firewall rule with the highest priority]

i.
```
gcloud compute firewall-rules create [RULE_NAME] \
```

1. *Replace RULE_NAME with the name of the firewall rule; for this instance, it would make sense to name the rule allow_egress or something similar.*

ii.
```
--direction=EGRESS \
```

iii. `--network=[NETWORK] \`

1. *Replace NETWORK with the name of the VPC network.*

iv. `--action=ALLOW \`

v. `--rules=tcp:1688 \`

vi. `--destination-ranges=35.190.247.13/32 \`

vii. `--priority=0`

1. *This sets the priority to the highest by placing it at the top of the list.*

b. [Verify that the instance has officially started]
   i. *Windows uses the sysprep process and may take a bit longer to start up; however, you can check whether the process has started by checking the port output.*

ii.
```
gcloud compute instances get-serial-port-output
[INSTANCE_NAME]
```

1. *Replace INSTANCE_NAME with the name of the instance created.*
2. *The output should be something like the following:*

a.
```
Instance setup finished.  [INSTANCE_NAME] is
ready to use.
```

c. [Exit from the GCloud terminal]
   i. *Simply typing "exit" will exit the GCloud Command Line }*

5. **Connect to the VM Instance via Chrome Remote Desktop > {**
   a. [On the **VM Instance** page, select the instance name of the Windows Server VM instance]
   b. [Click the **RDP** button]
   c. [When prompted, enter the password created for the account and click "OK"]
   d. [Click "Continue"]
      i. *This accepts the server certificate and logs into the remote windows instance.*
   e. [When prompted to allow your PC to be discoverable by other devices, choose "No"]
   f. [Close the **Server Manager Dashboard** if open] }
6. **Open PowerShell to Install Chrome and Chrome Remote Desktop > {**
   a. [Click the "Start" button and type "PowerShell" and select it]
   b. [Enable HTTPS requests through the PowerShell command:]

i.
```
[Net.ServicePointManager]::SecurityProtocol = "tls12,
tls11, tls"
```

c. [Download and run the Chrome remote desktop host installer]

```
i. & {$P = $env:TEMP + '\chromeremotedesktophost.msi';
   Invoke-WebRequest 'https://dl.google.com/edgedl/chrome-
   remote-desktop/chromeremotedesktophost.msi' -OutFile $P;
   Start-Process $P -Wait; Remove-Item $P}
```

ii. [Confirm that you want to make changes when prompted]

d. [Download and run Chrome installer]

```
i. & {$P = $env:TEMP + '\chrome_installer.exe'; Invoke-
   WebRequest 'https://dl.google.com/chrome/install/latest
   /chrome_installer.exe' -OutFile $P; Start-Process -
   FilePath $P -Args '/install' -Verb RunAs -Wait; Remove-
   Item $P}
```

ii. [Confirm that you want to make changes when prompted] }

7. **Set up the Chrome Remote Desktop Service > {**
    a. [Go to the following web address]
        i. `https://remotedesktop.google.com/access/`
    b. [Log into the account to use to access this machine]
        i. *This would be the* support@dang-designs.com *account*.
    c. [In the set up remote access dialog, click Download to reach web store page]
    d. [To install the extension, click Add to Chrome and Add Extension]
    e. [Accept the computer name and set pin]
        i. *Confirm that you want to make changes*.
    f. [Close Chrome windows, PowerShell, and the Remote Desktop window]
    g. [You can now connect through Chrome Remote Desktop!] }

References	Notes
https://cloud.google.com/compute/docs/instances/windows/creating-managing-windows-instances	Initial Windows Server configuration setup
https://cloud.google.com/compute/docs/instances/windows/creating-passwords-for-windows-instances	How to create password for instance to access machine
https://cloud.google.com/sdk/docs/how-to	How-to documents for GCloud terminal
https://ssh.cloud.google.com/cloudshell/editor?page=editor&_ga=2.240157888.296902771.1592834447-652029738.1591883601	Opens a web-based GCloud terminal shell instance
https://cloud.google.com/solutions/chrome-desktop-remote-windows-compute-engine	Describes how to install Google Chrome Remote Desktop utility for instance

Setting Up Fishbowl/MySQL Workbench on Windows Server

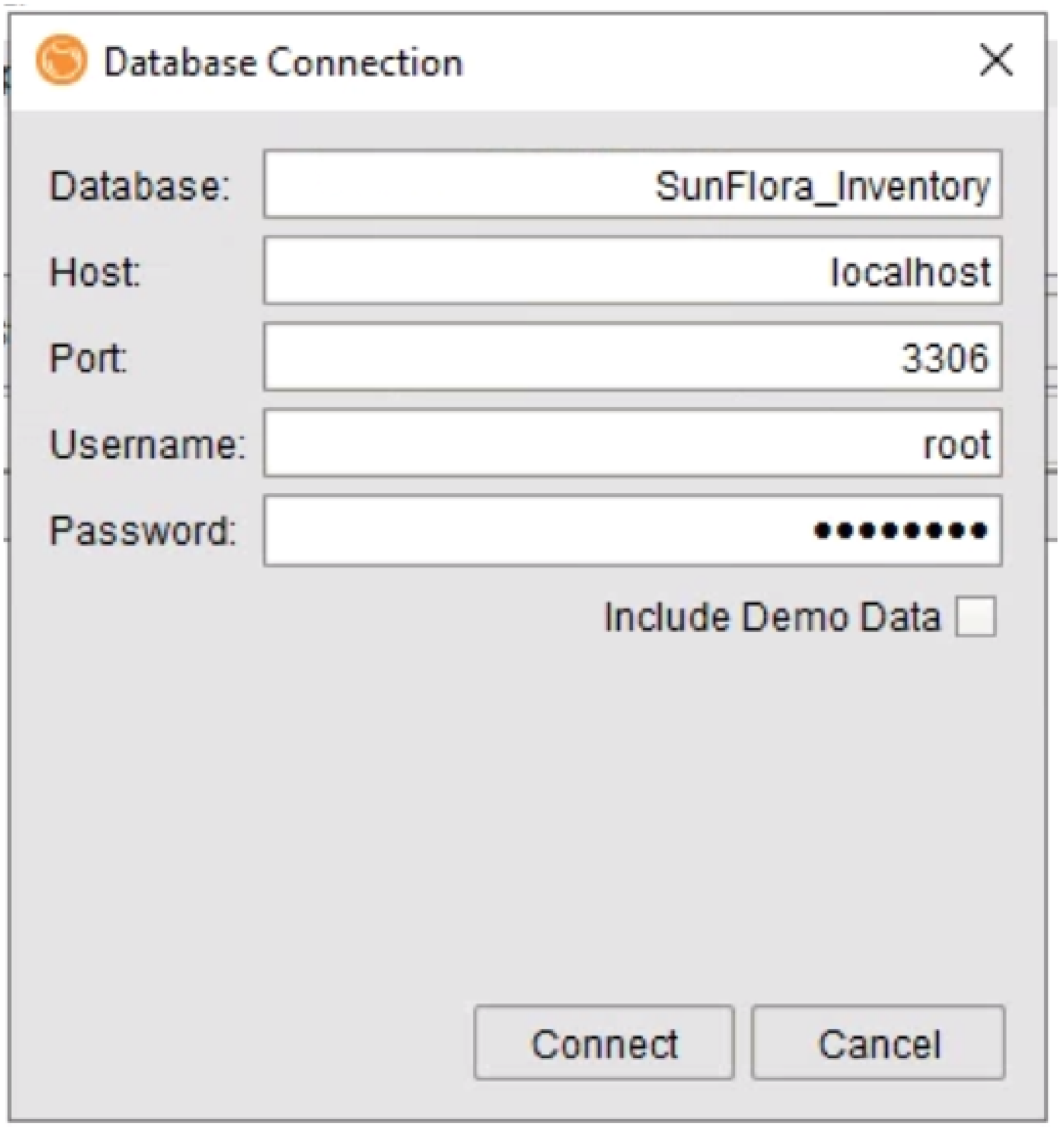
This page will cover some of the specifics to setting up the Workbench and Fishbowl on the Windows server that may not necessarily be explicitly derived.

WORKBENCH

1. **Go to the MySQL downloads page and download MySQL Workbench**
    a. https://dev.mysql.com/downloads/file/?id=497105
        i. This link may be outdated by the time of using it, but the process is intuitive.
2. **Download the MSI file.**
3. **Open the installer and follow the prompts**
    a. Windows offers a much more intuitive approach when installing this software. Make sure to capture all the dependencies for the workbench and press try again when installations pieces fail; sometimes they get caught while checking for library components newly installed.
4. **Set a password for the root account and ensure that the Workbench can open with a solid connection**
    a. This may sound redundant but should be performed to ensure no issues when completing.

5. **You're Done!**

FISHBOWL

1. **Navigate to the Fishbowl downloads page to download either the most recent version of Fishbowl or the one in current use as of writing this page (2019.11)**
    a. https://www.fishbowlinventory.com/support/updates/download/
    b. If performing a Google search, ensure to select the option with the header 'Download Updates'
    c. *If attempting to download both the server and the client, a new license key will be needed for testing or retaining the active session*
        i. Only 3 testing licenses are supported per year and this can be overruled by contacting the manager of the Fishbowl account (not support).
        ii. Only 1 permanent license is allowed at a time.
2. **Restore the database from a recent database copy**
    a. The Fishbowl server may come with demo data and a created connection. Remove this connection and navigate to the restore page on the server options.
    b. Database name: Sunflora_Inventory (This should auto assign either way)
    c. Port 3306
    d. Select a copy of the file to restore from the backup which must be removed from the zip file
        i. The backup process can take up to 25 - 30 minutes based on the size of the data file overall.
    e. Return to the Fishbowl Server Administration tool (Fishbowl Server > Tools > options) to add the database connection
        i. Database after fresh Fishbowl installation is still pointed to the Fishbowl Demo connection
        ii. Click the '+' icon on the right
        iii. Add the database credentials
            1. 

        iv. Click Connect and a message should appear that the database connection was established.
3. **With Fishbowl installed, place the necessary jar files in the plugins folder**
    a. The following JARs needed are located in the Google Drive under 01-DANG-Work > 00-Stock-&-Software-related > JAR Files
        i. advanced-shipstation-19.11.6.jar

        ii. fishbowl-go-18.12.01.jar
        iii. magento-cart-19.11.3.jar
    b. All of these JAR files need to be included in the Fishbowl > plugins folder
    c. After placing them in the plugins folder, pull up the client locally and download advanced shipstation from the plugins page, exit the server, and restart it.
4. **Deactivate all scheduled tasks**
    a. This is important so that orders are not shipped from within the wrong environment.
        i. Select the task and press the edit icon on the right side panel
        ii. Deselect the active checkbox and update the task as normal with the bottom buttons.
            1. **NOTE:** Any changes made within the plugins will cause the scheduled tasks to attempt to repost.
    b. As needed, also switch the environment over the the dev database.
        i. Under integrations, inside of Store Settings for Magento, change the URL to https://dev.sunflora.org
5. **Add the batch files and schedule the task for daily transfers**
    a. Batch files are located in 01-DANG-Work > CBD Store > Fishbowl > Batch Files
    b. Items are stored in the zip file to simplify the process
    c. Steps for this process can be found here: Setting Up A Batch File To Transfer Backups And Delete Them
6. **When reactivating tasks, make sure to edit the directory path for split payments and database backup tasks to ensure that they work successfully***
7. **You're Done!**

Website @media Responsive Queries

Always Design for Mobile First

Mobile First means designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices).

Instead of changing styles when the width gets *smaller* than 768px, we should change the design when the width gets *larger* than 768px. This will make our design Mobile First:

```
/* For mobile phones: */
[class*="col-"] {
width: 100%;
}

@media only screen and (min-width: 768px) {

  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

Typical Device Breakpoints Examples:

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}
```

```
/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}
```

```
/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

Change Font Size With Media Queries

```
/* If the screen size is 601px or more, set the font-size of <div> to
80px */
@media only screen and (min-width: 601px) {
  div.example {
    font-size: 80px;
  }
}
```

```
/* If the screen size is 600px or less, set the font-size of <div> to
30px */
@media only screen and (max-width: 600px) {
  div.example {
    font-size: 30px;
  }
}
```

Hide Elements With Media Queries

```
/* If the screen size is 600px wide or less, hide the element */
@media only screen and (max-width: 600px) {
```

```
  div.example {
    display: none;
  }
}
```

Media Queries For Columns

```
/* Create four equal columns that floats next to each other */
.column {
  float: left;
  width: 25%;
}
```

```
/* On screens that are 992px wide or less, go from four columns to two
columns */
@media screen and (max-width: 992px) {
  .column {
    width: 50%;
  }
}
```

```
/* On screens that are 600px wide or less,
 * make the columns stack on top of each other
 * instead of next to each other */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
  }
}
```

Min Width to Max Width

```
media screen and (max-width: 900px) and (min-width: 600px) {
  div.example {
    font-size: 50px;
    padding: 50px;
    border: 8px solid black;
    background: yellow;
  }
}
```

Responsive

This component provides responsive classes to hide or display elements on different viewports.

Hidden Add one of the `.uk-hidden-*` classes to hide the element from screens larger than a specified width.	
CLASS	**DESCRIPTION**
`uk-hidden@s`	Only affects device widths of *640px* and larger.
`uk-hidden@m`	Only affects device widths of *960px* and larger.
`uk-hidden@l`	Only affects device widths of *1200px* and larger.
`uk-hidden@xl`	Only affects device widths of *1600px* and larger.

```
<!-- Hidden on tablets and larger -->

<div class="uk-hidden@m"></div>
```

Where To Update IP Addresses On Change

This page will collect all of the locations where, when the IP address changes for the workplace, these IP addresses need to be updated. The goal of this list is to be as comprehensive as possible for when these changes occur to swiftly update every instance.

Locations

- Google Cloud VM Firewall Rules
  - Firewall rules set for gcloudhost, windows servers (1, 2, and 3) (No longer active)
    - **Corrected with CIDR notation*
- Dev sites htaccess File
  - 360dev.dang-designs.com
  - 360new.dang-designs.com
  - banner.dang-designs.com
  - sandbox.cbdrx4u.com
  - develop.cbdrx4u.com
  - demo-wp.dang-designs.com
  - demo.dang-designs.com
  - demo-joom.dang-designs.com
  - hoffroad.dang-designs.com
  - kristidev.dang-designs.com
  - sandbox.dang-designs.com
  - riversidev.dang-designs.com
  - segno.dang-designs.com
  - m2dev.capitalcityshoes.com
  - sandbox.capitalcityshoes.com
  - sandbox.sunflora.org

For IPs in the .htaccess files, look for the piece that looks similar to this code below:

```
Order Deny,Allow
Deny from all
Allow from 47.205.11.214
```