4.6 Apache Domain and URL Consistency Best Practices

It is often desirable to have consistency on how your domain name brand looks on your website. Here are the practices we use based on evolving de facto standards.

With net new websites, it's easy to follow these practices. With legacy sites, I've starred disruptive changes ⭐ which you should consider phasing in.

Practice	Example	Why	How to Do It
Remove context root.	http://bonsaiframework.com/content/**bonsai**/index.html http://www.bonsaiframework.com/index.html	The context root is not relevant to customers. This is more important for Customer public with portals where linking is desirable.	Using Apache rewrite urls. Normal virtualhosts will be fine, but more enterprise CMS platforms will likely require.
Normalize to SSL	http://www.bonsaiframework.com http**s**://www.bonsaiframework.com	As for July 2018 Chrome browser version 68 started marking normal sites as "not secure".	Bonsai Framework covers how to normalize to SSL with your Apache Web Server.For large organizations this can be done with your CDN. Depending on your Customer base you may want to consider what version of TLS to use and if you need a touchpoint site for legacy protocols. ⭐ Legacy sites with integration points may break if you do a global SSL enforcement. In this scenario, a phased approach by using analytics and flipping key entry points instead may be the way to go. I'll do a write-up on this if there is enough interest.
Normalize to www over SSL	http://bonsaiframework.com http**s://www**.bonsaiframework.com Including full urls such as http://bonsaiframework.com/ubuntu/ http**s://www**.bonsaiframework.com/ubuntu/	Visually it's more appealing and ensures your domain name stands out more. Major sites, Google, Microsoft and Apple do this. Some people disagree with me on this and instead remove the www. That's ok, but stay consistent.	Bonsai Framework shows how to enforce www but for large organizations consider working with your CDN.
Do not requires index.html.	https://www.bonsaiframework.com/ubuntu/**index.html** https://www.bonsaiframework.com/ubunut/	Visually more appealing and less work for users.	Baked into default by Apache but you need to build your site using index.html files and not something like http://www.bonsaiframework.com/ubuntu/homepage.html as our starting page. For organizational search engine optimization reasons you may need to use special names. In that case ... (I gotta review the best method still here).
File names versus folders.	Folder approach, https://www.bonsaiframework.com/**ubunut/** https://www.bonsaiframework.com/ubuntu/**setup/** * config does not require trailing slash to work https://www.bonsaiframework.com/**ubunutu** https://www.bonsaiframework.com/ubuntu/**setup** File approach, https://www.bonsaiframework.com/ubunut/**ubuntu.html** https://www.bonsaiframework.com/ubuntu/**setup.html**	The argument for folder based approach is it looks more visually appealing.	In the folder approach you need to create folders and the default doc file name itself is called **index.html**. Trailing space may be omitted and this is stock Apache out of the box. File approach is self explanatory.

Use Sub Domains	www.**linux**.bonsaiframework.com www.bonsaiframework.com/**linux** However, there are arguable cases (fast type for often used distinct sub-site or restrictions (ie separate system). www.bonsaiframework.com - public site www.**wiki**.bonsaiframework.com - secure site used for authoring internal content.	One consideration is that for best SEO it is better to use one domain. However, migrating ie to a new CMS is much tougher. This may be mitigated using CDN's rewrite or Apache which has its own level of complexity.	
Internationalization and urls	There are 3 popular variations, often with the **home** country being the .com. In this case I'm Canadian, **Subdirectory** www.bonsaiframework.com/**ca/en** - Canada English www.bonsaiframework.com/**ca/fr** - Canada French www.bonsaiframework.com/**mx/es** - Mexico Spanish **Subdomain Extension** www.bonsaiframework.com.**com/en** - Canada English www.bonsaiframework.com.**com/fr** - Canada French www.bonsaiframework.com.**mx/es** - Mexico Spanish **Subdomain Prefix** www.bonsaiframework.com/en - Canada English www.bonsaiframework.com/fr - Canada French www.**mexico**.bonsaiframework.com/es - Mexico Spanish **Alternative Domain** www.bonsaiframework.com/en - Canada English www.bonsaiframework.com/fr - Canada French www.bonsaiframework.**mx**/es - Mexico Spanish	There is a no best one here. Two considerations are, 1. What is popular in the countries being served. 2. Mixing (to cater to country) versus standardizing for consistency. **Subdirectory** ... **Subdomain** Pros - possible to bind to different systems for ease of migration (say moving to a new CMS one site at a time). Cons - there are some ssl certificates overhead to maintain with more subdomains. Adam chime in here if modern certs make this overhead negligible.	ISO letter country code followed by ISO letter language code. For country code, use the 2 letter SO 3166-1 alpha-2 supplemented by three-letter ISO 3166-1 alpha-3 codes when no two-letter code is applicable. Similarly for language code, use the 2 letter ISO 6391 supplemented by three-letter ISO 6392 codes.
Canonicalize your duplicate pages.	...		
Keeping both 443 and 80 Open	Option A https://www.bonsaiframework.com is default, but if users hit http://www.bonsaiframework.com it will still work staying on port 80. Option B Port 80 redirects to https with with a support website specifically made to work with older technology and help users upgrade, http://www.**support**.bonsaiframework.com and https://www.**support**.bonsaiframework.com	The argument here is that public sites are the last line of defense for Customer information. If only 443 is supported, This would be for the channel to communicate.	

SSL TLS version support. Do we continue to support 1.0 and implications as we drop?	www.bonsaiframework.com https://www.bonsaiframework.com	Better rankings in search engines and confidence in users. Additionally prevents alleviates against man-in-the-middle attacks. Any collection of customer information should use TLS 1.2 or higher.	Setup TLS and cipher orders from best to worse. For example, start with TLS 1.2 first and then downgrade to TLS 1.1 ect... Redirect 80 to 443, but notify users if their browser does not support ciphers. The challenge for large organizations is how to redirect for older browsers that simply do not support the minimal TLS (this needs to be it's own article). Today (Oct 2017) google still supports TLS 1.0. The PCI Council originally wanted to declare TLS 1.0 obsolete by June 30, 2016 but since pushed back to June 30, 2018. Google and of the TLS enabled Canadian Banks Public websites, only BMO has dropped TLS 1.0 support. Interestingly enough initial tests of secured portion of Canadian Bankings sites still support TLS 1.0. Still to research how older browsers react major websites that have dropped TLS 1.0.
Redirects should be file based and generally aim to use 301 "Moved Permanently".		Moved permanently is best for SEO and passes between 90-99% of ranking power to the new page. Note transferring content becomes more complex when an entire site changes its domain or when content moves from one domain to another.	If you have a CDN look for an API which under the hoods uses a web server redirect. For example Akamai's Edge Redirector Cloudlet can be programmatically populated via their API. You may use your Web Server such as Apache mod_rewrite with the caveates, • Lose the speed of rewrites happening on the CDN edge servers. • May create significant load on your Web Servers.