

Pre-Dogfood Server

Managed by the [Infrastructure Team](#) and [Dogfood Team](#), this is to have a daily test upgrade of most of our *.tiki.org sites. So anyone can check anytime how stable trunk (the next version) is with real world data. The name relates to our *eat our own dogfood* policy.

The main sites

Production	https://themes.tiki.org	https://doc.tiki.org	https://dev.tiki.org	https://tiki.org
Copy for developers to troubleshoot without touching production	https://copythemes.tiki.org	https://copydoc.tiki.org	https://copydev.tiki.org	
Trunk	https://trunkthemes.tiki.org/ (PHP 8.2)	https://trunkdoc.tiki.org/ (PHP 8.2)	https://trunkdev.tiki.org/ (PHP 8.2)	http://next.tiki.org/
Testing for Converting from Tiki syntax to Markdown	https://markdownthemes.tiki.org	https://markdowndoc.tiki.org	https://markdowndev.tiki.org	
Special	https://24xthemes.tiki.org will exist for quite a while as a reference for Bootstrap 4 themes			

Challenges

- [Dogfooding](#) Tiki on our own sites is great to find bugs, but it makes it difficult to work.
- Say I find a bug on a tiki.org site, is it fixed in the next version, or should I report/try to fix bug?
 - This is very time consuming.
- Detecting and fixing bugs is much less efficient without real data. We would like to give developers access to a site where they can test a fix, before committing (or very quickly after)

Which is why we have set up a continuous Pre-Dogfood Server. *Yesterday's data with tomorrow's code* (code in trunk). Ideally, this would be part of [TRIM](#) so everyone in the community could easily set this up for themselves, but it's not at this stage.

Warnings

- Make sure both db/local.php have same settings so you don't get an issue like <http://dev.tiki.org/item5300>

Make your own pre-dogfood server

The future-proof solution is Clone and Upgrade in [Tiki Manager](#)

Alternatively, some other code is here:

<https://gitlab.com/tikiwiki/tiki/-/blob/master/doc/devtools/update.dogfoodserver.sh>

System Configuration

Use the following as per [System Configuration](#). See also [Divergent Preferences in Staging Development Production](#)

Create a tiki.ini file with some or all of the following content



```
; To avoid bots indexing the site and losing SEO over duplicate content
preference.metatag_robots = &quot;noindex, nofollow&quot;

; Testing changes shouldn't spam people
preference.feature_user_watches = n

; Make it obvious to everyone that this is not real data site
preference.sitesubtitle = &quot;Pre-dogfood test upgrade server. Data refreshed
daily.&quot;

; Set error reporting to all (useful for debugging)
preference.error_reporting_level = -1

; Make error reporting available to all users, not just admins
preference.error_reporting_adminonly = n

; Here are some less commonly used options. Just uncomment the prefs to be like above by
removing the ;

; If you are using memcache in production, you probably want to deactivate for the pre-
dogfood server, or at least change the memcache_prefix so the production and pre-dogfood
server don't collide
; preference.memcache_enabled = n

; If you are using a Content Delivery Network (CDN), you want to make sure values are
not the same in production vs pre-dogfood server
; tiki_cdn = &quot;&quot;
; tiki_cdn_ssl = &quot;&quot;

; If you want to prevent user registrations, uncomment below
; preference.allowRegister = n

; To close the site. Only users with tiki_p_admin permissions can log in.
; preference.site_closed = y

; Closed site message
; preference.site_closed_msg = &quot;This is a development site, not open to the
public&quot;
```

Is enough metatag_robots enough? Or is modifying <https://next.tiki.org/robots.txt> also necessary?

Benefits

- Upgrade procedure is tested very frequently
- Makes it very easy to compare old & new interfaces (and people can say what they liked/want to keep about the old one)
- Make it easier for Tiki community to make a judgment on when to start dogfooding next version on community site.

Potential users

- All devs: To test before an upgrade. Ex.: They could load test daily and report any feature or performance regressions.

Ideas to improve

- [Continuous Integration Team](#) could set up some tests on pre-dogfood servers
 - A way to receive alerts on major breaking
 - SQL error, PHP error
 - Would need a crawler for early detection of breakage
- all pre-dogfood servers should indicate in how many hours is next data refresh
- To make *.tiki.org upgrades easier, to catch regressions sooner, dev.tiki.org could run on monthly updates of trunk code, while nextdev.tiki.org is daily trunk. Before each monthly upgrade, nextdev.tiki.org is checked to catch and fix regressions.
- ~~Change the domain name from trunk to next, because it's not always trunk, but always next done~~
- [Tracker item: #4200 - - Compare generated HTML for all *.tiki.org sites between versions \(to check for regressions/changes of behavior\)](#)
- Handle `_htaccess`
 - Use unix file link from `.htaccess` to `_htaccess`
- Find a solution for RewriteBase to avoid " The RewriteBase directive seems not to be set up correctly. This is required for sefurl to function correctly. The current value in `.htaccess` is `/` but the base url for this site is `/trunk/` "
- ~~Clear all cache~~ Now part of the shell installer
- To be able to apply a profile or a [System Configuration](#)
- To do the same thing, but for various [profiles](#).

Related

- [Continuous Integration](#)

alias

- [Pre-Dogfood servers](#)