

Core - Task # 54316

Status:	New	Priority:	Should have
Author:	Stefan Neufeind	Category:	
Created:	2013-12-11	Assigned To:	
Updated:	2014-03-04	Due date:	
TYPO3 Version:	6.2		
PHP Version:			
Complexity:			
Sprint Focus:			
Subject:	Provide a Nginx server configuration		
Description	<p>We ship with .htaccess-configuration (by default only as an example). Imho these we should also provide an equivalent for nginx-syntax.</p>		
Related issues:			
related to TYPO3.Flow - Task # 8923: Provide a Nginx Server Configuration for...		Under Review	2010-07-18
related to Core - Task # 56553: Move files from toplevel to docs-directory		Rejected	2014-03-04

History

#1 - 2013-12-11 03:40 - Stefan Neufeind

Because nginx won't magically pull in configuration because it exists in a file (like .htaccess) that would more be a "documentation". But imho we should still ship it side-by-side with the example .htaccess and not just hide it somewhere in a wiki or even the TYPO3-documentation.

#2 - 2014-03-04 20:08 - Pascal Dürsteler

I am not sure where I should put this, otherwise I'd have made a commit. I adapted the .htaccess for apache for nginx and it seems to work quite nice. I am still testing it out, though.

A few points to mention:

- I didn't copy over the comments from .htaccess 1:1, but made some changes to them to fit nginx
- I modified a few regexes to include more things, mostly for the sake of security
- I added a "security" block to provide some basic security-related rules. I felt like this would be the right place.
- There is no php-cgi or php-fpm block included, as I assume a sane environment with php already working, when someone is deploying a TYPO3 setup on nginx. However, I'd suggest to put a second snippet into the documentations folder about the proper setup of php-fpm, since most of the tutorials are vulnerable to arbitrary code execution (see http://wiki.nginx.org/Pitfalls#Passing_Uncontrolled_Requests_to_PHP).
- I am not yet satisfied with the versioned static files rules, as an IF gets evaluated on EVERY request, which is a bit of a performance sucker. This may be solvable with try_files.

In addition to the previous points: I've put this configuration into conf.d/typo3.conf next to a php5.conf which contains the said php5-fpm settings. This allows very flexible host setups, as you just need to provide a server-block with root and hostname, and then include the desired configurations. So, my host config file is as simple as:

```
server {
    server_name ~(*.*)?mydomain\.*;
    root /var/www/mydomain/public;

    include conf.d/php5.conf;
    include conf.d/typo3.conf;
}
```

The actual typo3.conf:

```
#####  
#  
# Example configuration file for TYPO3 CMS - for use with NGINX Webserver.  
#  
# This file includes settings for the following configuration options:  
#  
# - Compression via TYPO3  
# - Security  
# - Settings for URL rewriting  
#  
# If you want to use it, you have to include the following directives into your "server" block,  
# either by manually pasting it there or by using the "include" directive.  
#  
# IMPORTANT: You may need to change this file depending on your TYPO3 installation!  
#  
# Lines starting with a # are treated as comment and ignored by the web server.  
#  
# Questions about this file go to the matching Install mailing list, see  
# http://typo3.org/documentation/mailling-lists/  
#  
#####  
  
### Begin: Compression via TYPO3 ###  
#  
# Compressing resource files will save bandwidth and so improve loading speed especially for users  
# with slower internet connections. TYPO3 can compress the .js and .css files for you.  
# *) Uncomment the following lines and  
# *) Set $TYPO3_CONF_VARS['BE']['compressionLevel'] = '9' for the Backend  
# *) Set $TYPO3_CONF_VARS['FE']['compressionLevel'] = '9' together with the TypoScript properties  
#   config.compressJs and config.compressCss for GZIP compression of Frontend JS and CSS files.  
  
# Enable gzip compression  
#gzip on;  
  
# Disable gzip compression for browsers that don't support it (in this case MS Internet Explorer  
# before version 6 SV1).  
#gzip_disable "MSIE [1-6]\.(?!.*SV1)";  
  
# Set the response header Vary: Accept-Encoding.  
# Some proxies have a bug in that they serve compressed content to browsers that don't support it.  
# By setting the Vary: Accept-Encoding header, you instruct proxies to store both a compressed and  
# uncompressed version of the content.  
#gzip_vary on;  
  
# Enables or disables gzipping of responses for proxied requests depending on the request and response.  
#gzip_proxied any;  
  
# This tells nginx what file types to compress (text/html is always compressed)  
#gzip_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript  
application/javascript;
```

```
# Add mime-type for compressed js files.
```

```
#location ~ /\.js\.gzip {  
# types {  
#   text/javascript gzip;  
# }  
#}
```

```
# Add mime-type for compressed css files.
```

```
#location ~ /\.css\.gzip {  
# types {  
#   text/css css;  
# }  
#}
```

```
### End: Compression via TYPO3 ###
```

```
### Begin: Browser caching of resource files ###
```

```
# Enable long browser caching for assets. This affects Frontend and Backend and increases performance.
```

```
location \.(css|js|gif|png|jpg|svg)$ {  
# etag is supported on nginx >= 1.3.3  
# etag on;  
expires max;  
}
```

```
### End: Browser caching of resource files ###
```

```
### Begin: Security ###
```

```
#  
# Prevent information disclosure by blocking files possibly containing sensitive information.
```

```
# Block access to hidden" directories or files.
```

```
location ~ /\. {  
deny all;  
access_log off;  
log_not_found off;  
}
```

```
# Block access files accidentally left on the server.
```

```
location (\.(bak|config|sql|\.zip|\.gz|\.bz2)?|ini|log|sh|inc|swp|t3d|~)$ {  
deny all;  
access_log off;  
log_not_found off;  
}
```

```
# Restrict access to deleted files in Recycler directories
```

```
location ~ ^/fileadmin/(.*)?_recycler_/ {  
deny all;  
access_log off;  
log_not_found off;  
}
```

```

# Restrict access to TypoScript files in default templates directories
location ~ ^/fileadmin/templates/.*(\.txt|\.ts)$ {
    deny all;
    access_log off;
    log_not_found off;
}

# Restrict access to Private extension directories
location ~ ^/typo3conf/ext/[^/]+/Resources/Private/ {
    deny all;
    access_log off;
    log_not_found off;
}

### End: Security ###

### Begin: Settings for url rewriting ###
#
# You need rewriting, if you use a URL-Rewriting extension like realurl or cooluri.

# Rule for versioned static files, configured through:
# - $TYPO3_CONF_VARS['BE']['versionNumberInFilename']
# - $TYPO3_CONF_VARS['FE']['versionNumberInFilename']
# if (!-e $request_filename) {
# rewrite ^(.(+)\.(d+)).(php|js|css|png|jpg|gif|gzip)$ $1.$3 last;
#}

# Main URL rewriting
location / {
    try_files $uri $uri/ /index.php$is_args$args;
}

### End: Settings for url rewriting ###

```