

# Nginx redirect all requests from subdirectory to another subdirectory root

Asked 11 years, 7 months ago Modified 7 years, 10 months ago Viewed 54k times  Part of PHP Collective



I'm quite new to Nginx so please bear with me.

19

I'm trying to redirect all requests from one subdirectory (store) to the root of another subdirectory (trade). See my progress below. The site in the target subdirectory (trade) is a magento site so that is what most of the current rules are for.



```
server {
    server_name example.com *.example.com;
    root /usr/share/nginx/html/example.com/public_html;
    index index.php index.html index.htm;

    access_log /var/log/nginx/example.access.log;
    error_log /var/log/nginx/example.error.log;

    location / {
        try_files $uri $uri/ /index.html;
    }

    location /trade/ {
        index index.html index.php;
        try_files $uri $uri/ @handler;
        expires 30d;
    }

    location ~ /store {
        rewrite /trade permanent;
    }

    location ~
    ^/trade/(app|includes|lib|media/downloadable|pkginfo|report/config.xml|var)/ {
        internal; }
    location /trade/var/export/ { internal; }
    location /. { return 404; }
    location @handler { rewrite / /trade/index.php; }

    error_page 404 /404.html;

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

```
}
}
}
```

The section I am using to redirect is the following:

```
location ~ /store {
    rewrite /trade permanent;
}
```

This works for `example.com/store` but not `example.com/store/index.php` or any other uri with args. I have a feeling that the php file section at the bottom is overriding the processing. That is why I have put the `~` in front of the store location as the documentation [here](#) states this will be processed first. Does the processing stop or continue on?

I have read about nesting a php rule but I have tried this to no avail.

I would greatly appreciate any help.

[PHP](#) [php](#) [http-redirect](#) [nginx](#)

Share Improve this question Follow

asked Mar 6, 2014 at 12:23



[user1214769](#)

371 ● 1 ● 2 ● 6

An example url would be [example.com/store/...](#) – [user1214769](#) Mar 7, 2014 at 9:12

Does this answer your question? [Nginx redirect all contents of a subdirectory to another subdirectory](#)  
– [Bobort](#) Feb 2, 2023 at 16:28

### 3 Answers

Sorted by: Highest score (default)



ok try something like this

45

```
location ^~ /store(.*) {
    return 301 $scheme://$http_host/trade$1$is_args$query_string;
}
```



Trying to avoid hardcoded stuff as much as possible and using return because [it's preferred over permanent rewrites](#)



Share Improve this answer Follow

answered Mar 7, 2014 at 21:45



! Sign up to request clarification or add additional context in comments. ✕

## 5 Comments ▼

Add a comment



[user1214769](#) Over a year ago

Thanks for the reply, I would vote you up but I haven't got the reputation!

▲ 4 Reply ⋮



[Mohammad AbuShady](#) Over a year ago

It's ok don't worry :-D

▲ 1 Reply ⋮



[fideloper](#) Over a year ago

I had to use `~*` to get this to work, e.g. `location ~* /store(.*) { ... }`

▲ 1 Reply ⋮



[Lemonade](#) Over a year ago

It needs to be `: location ~ ^/store/(.*) { ... }`

▲ 2 Reply ⋮



[Mohammad AbuShady](#) Over a year ago

@Lemonade you can check the documentation, I would say both work, but I'm not sure if one is better optimized than the other [nginx.org/en/docs/http/nginx\\_http\\_core\\_module.html#location](https://nginx.org/en/docs/http/nginx_http_core_module.html#location)

▲ 0 Reply ⋮



Ok,

7

Coming back to this I can see the issue.



In Nginx when you prepend a location directive with `~` this means that you want to process regular expressions in your directive (case sensitive, `~*` for case insensitive). I believe that all regex directives will process before any others but I stand to be corrected.



So when I am using:

```
location ~/store {
    rewrite /trade permanent;
}
```

There is no regex there. Its is simply matching `/store` and redirecting to `trade`.

After some investigation (and polishing up on my regex, which is rubbish), I came back to it and have come up with a working solution.

```
location ~ ^/store/(.*) {
    rewrite ^/store(.*) /trade permanent;
}
```

Here I am asking the directive to process the regex by entering ~ then match any url with /store/ in it.

Then, according to the docs, the rewrite syntax is:

```
rewrite regex replacement [ flag ]
```

so I am matching all urls with store in it and permanently redirecting them to the new subfolder.

Pretty easy really, embarrassingly so actually but hey, every day is a school day. I'm open to correction on all of this and hope it helps someone.

Share Improve this answer Follow

answered Mar 7, 2014 at 10:09



[user1214769](#)  
371 ● 1 ● 2 ● 6

## Comments

Add a comment



1



You need to ensure that your `location ~ /\.php$` handler does not take any URLs below the old folder. Indeed, precedence rules are clearly documented within <http://nginx.org/r/location>, and you can either use regular expressions, or, better yet, use prefix-based matching with the `^~` modifier to instruct that the search must stop without trying to see if that regex-based `\.php$`

`location` would match:

```
location ^~ /old/long/path/ { # will match /old/long/path/index.php, too
    rewrite ^/old/long/path/(.*)$ /new/$1 permanent;
}
```

The above snippet is likely the most efficient way of doing this, but here is another way of doing the same:

```
location ~ /old/long/path/(.*) {
    return 301 /new/$1$is_args$args;
}
```

Why does one example has `$is_args$args` and the other one doesn't? Good question! Note that `location` directive as well as the first parameter of the `rewrite` directive both operate based on the contents of the `$uri` variable, as opposed to `$request_uri`. Long story short, but `$uri` does not contain `$args`, so, in both cases, `$1` will not contain any `args`; however, in the case of `rewrite`, the case is deemed so common that `$args` are automatically added back by nginx, unless the new string ends with a `?` character, see <http://nginx.org/r/rewrite>.

Share Improve this answer Follow

answered Dec 10, 2017 at 4:38



cnst

27.5k ● 7 ● 100 ● 131

## Comments

Add a comment

### Start asking to get answers

Find the answer to your question by asking.

Ask question

### Explore related questions

php http-redirect nginx

See similar questions with these tags.