

# How to disable proxy cache when query string is empty?

Asked 8 years, 8 months ago   Modified 8 years, 8 months ago   Viewed 10k times



With nginx I have

2



```
server {  
    listen 1.2.3.4:80  
    proxy_cache_valid      200 302 5m;  
    location / {  
        try_files $uri @upstream;  
        root $root;  
    }  
}
```

When I go `http://example.com/foobar` it generates a redirect to `http://example.com/foobar?filter_distance=50&...` which is visitor dependent so I would like to not cache this redirect. I need to bypass cache when the query string is empty. I am a bit lost because `location /foobar` will match both.

nginx

Share   Improve this question   Follow

edited Aug 19, 2014 at 10:46

asked Aug 19, 2014 at 10:13



chx

1,695 ● 2 ● 16 ● 25

## 2 Answers

Sorted by: Highest score (default)



I added

2

map \$request\_uri \$nocache {  
 /foobar 1;  
}

to the http section and

```
proxy_cache_bypass $nocache;  
proxy_no_cache $nocache;
```

to the server section. This seems to be working.

Share Improve this answer Follow

edited Aug 20, 2014 at 1:49

answered Aug 20, 2014 at 1:30



chx

1,695 ● 2 ● 16 ● 25

Nice, I haven't used the `map` feature at all, so I didn't think of it. – Tero Kilkanen Aug 20, 2014 at 9:20

It's hard to get into the nginx config headspace but once you are there, it's very neat: the default for `map` is the empty string and `proxy_no_cache` doesn't fire for empty strings. – chx Aug 20, 2014 at 10:43

You should use the `proxy_cache_bypass` and `proxy_no_cache` directives like this:

```
1 set $nocache 0;  
  
if ($arg_filter_distance = "") {  
    set $nocache 1;  
}  
  
proxy_cache_bypass $nocache;  
proxy_no_cache $nocache;
```

Definition of `proxy_no_cache` from [nginx documentation](#)

Defines conditions under which the response will not be saved to a cache. If at least one value of the string parameters is not empty and is not equal to "0" then the response will not be saved:

Here we test if the `filter_distance` GET parameter is empty. If it is, we set `$nocache` to 1, and then the `proxy_cache_bypass` and `proxy_no_cache` directives will get active.

You can add other GET parameters similarly, for example `$arg_filter_type`, if you have a `filter_type` GET parameter.

Share Improve this answer Follow

edited Aug 19, 2014 at 10:51

answered Aug 19, 2014 at 10:24



Tero Kilkanen

36.3k ● 3 ● 40 ● 62

---

Sorry. You got it backwards. I need a `proxy_no_cache` if `$arg_filter_distance` is empty. The title of the question says so but I edited my question to emphasize.  
– [chx](#) Aug 19, 2014 at 10:46

---

See updated answer. – [Tero Kilkanen](#) Aug 19, 2014 at 10:51

---

Sorry for the many questions... but I wanted to avoid `if` since [wiki.nginx.org/lflsEvil](http://wiki.nginx.org/lflsEvil) but I guess here it's OK? – [chx](#) Aug 19, 2014 at 10:57

---

It is a necessary evil in this spot. – [Tero Kilkanen](#) Aug 19, 2014 at 12:00

---

`if` breaks `try_files`, however. – [chx](#) Aug 20, 2014 at 1:19

---

# nginx "starts with" regex

Asked 7 years, 8 months ago   Modified 7 years, 8 months ago   Viewed 5k times

I'm trying to block/allow access to a particular vhost based on string matching of the HTTP user agent name.

1 The version number of the application in the HTTP user agent changes, e.g.

```
My%20App/1.55.01 CFNetwork/711.5.6 Darwin/14.0.0
My%20App/1.49.03 CFNetwork/711.5.6 Darwin/14.0.0
My%20App/1.35.02 CFNetwork/711.5.6 Darwin/14.0.0
```

I want to match everything starting with `My%20App/` - anything that matches should be allowed access, anything else should get HTTP 403.

I don't want to have to update the nginx config every time a new version of the app comes along.

The following works fine:

```
if ($http_user_agent !~* "My%20App/1.55.01 CFNetwork/711.5.6 Darwin/14.0.0") {
    return 403;
}
```

I figure I need a solution that involves hat (^) for "starts with" but I can't get nginx to accept anything I've tried.

Thanks in advance.

nginx

regex

Share Improve this question Follow

asked Sep 6, 2015 at 14:42



tomfanning

3,368 ● 7 ● 34 ● 34

What's wrong with simple `if ($http_user_agent !~ "^My App/") { return 403; }` – Alexey Ten Sep 7, 2015 at 7:31

## 1 Answer

Sorted by: Highest score (default)



Turns out that the following works.

3



```
if ($http_user_agent !~* "My%20App/*") {  
    return 403;  
}
```



Share Improve this answer Follow

answered Sep 7, 2015 at 11:27



tomfanning

3,368 ● 7 ● 34 ● 34