

Nginx Sticky Module

modified and extended version; see Changelog.txt

Description

A nginx module to add a sticky cookie to be always forwarded to the same upstream server.

When dealing with several backend servers, it's sometimes useful that one client (browser) is always served by the same backend server (for session persistence for example).

Using a persistence by IP (with the `ip_hash` upstream module) is maybe not a good idea because there could be situations where a lot of different browsers are coming with the same IP address (behind proxies) and the load balancing system won't be fair.

Using a cookie to track the upstream server makes each browser unique.

When the sticky module can't apply, it switches back to the classic Round Robin Upstream or returns a "Bad Gateway" (depending on the `no_fallback` flag).

Sticky module can't apply when cookies are not supported by the browser

> Sticky module is based on a "best effort" algorithm. Its aim is not to handle
> security somehow. It's been made to ensure that normal users are always
> redirected to the same backend server: that's all!

Installation

You'll need to re-compile Nginx from source to include this module. Modify your compile of Nginx by adding the following directive (modified to suit your path of course):

```
./configure ... --add-module=/absolute/path/to/nginx-sticky-module-ng
make
make install
```

Usage

```
upstream {
    sticky;
    server 127.0.0.1:9000;
    server 127.0.0.1:9001;
    server 127.0.0.1:9002;
}

sticky [name=route] [domain=.foo.bar] [path=/] [expires=1h]
[hash=index|md5|sha1] [no_fallback] [secure] [httponly];
```

- name: the name of the cookies used to track the persistent upstream srv;
default: route
- domain: the domain in which the cookie will be valid
default: nothing. Let the browser handle this.
- path: the path in which the cookie will be valid
default: /
- expires: the validity duration of the cookie
default: nothing. It's a session cookie.
restriction: must be a duration greater than one second
- hash: the hash mechanism to encode upstream server. It can't be used with hmac.
default: md5
 - md5|sha1: well known hash

- index: it's not hashed, an in-memory index is used instead, it's quicker and the overhead is shorter
Warning: the matching against upstream servers list is inconsistent. So, at reload, if upstreams servers has changed, index values are not guaranteed to correspond to the same server as before!
USE IT WITH CAUTION and only if you need to!
 - hmac: the HMAC hash mechanism to encode upstream server
It's like the hash mechanism but it uses hmac_key to secure the hashing. It can't be used with hash.
md5|sha1: well known hash
default: none. see hash.
 - hmac_key: the key to use with hmac. It's mandatory when hmac is set
default: nothing.
 - no_fallback: when this flag is set, nginx will return a 502 (Bad Gateway or Proxy Error) if a request comes with a cookie and the corresponding backend is unavailable.
 - secure enable secure cookies; transferred only via https
 - httponly enable cookies not to be leaked via js
- # Detail Mechanism
- see docs/sticky.{vsd,pdf}
- # Issues and Warnings:
- when using different upstream-configs with stickyness that use the same domain but refer to different location - configs it might be wise to set a different path / route
- option on each of this upstream-configs like described here:
<https://bitbucket.org/nginx-goodies/nginx-sticky-module-ng/issue/7/leaving-cookie-path-empty-in-module>
 - sticky module does not work with the "backup" option of the "server" configuration item
 - sticky module might work with the nginx_http_upstream_check_module (up from version 1.2.3)
 - sticky module may require to configure nginx with SSL support (when using "secure" option)
- # Contributing
- please send/suggest patches as diffs
 - tickets and issues here: <https://bitbucket.org/nginx-goodies/nginx-sticky-session-ng>
- # Downloads
- tarballs are available via tags from the repo: <https://bitbucket.org/nginx-goodies/nginx-sticky-module-ng/downloads>
- # TODO
- see Todo.md
- # Authors & Credits
- Jerome Loyet, initial module
 - Markus Linnala, httponly/secure-cookies-patch
 - Peter Bowey, Nginx 1.5.8 API-Change
 - Michael Chernyak for Max-Age-Patch
 - anybody who suggested a patch, created an issue on bitbucket or helped improving this

module

Copyright & License

This module is licenced under the BSD license.

Copyright (C) 2010 Jerome Loyet (jerome at loyet dot net)
Copyright (C) 2014 Markus Manzke (goodman at nginx-goodies dot com)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.