# Apache Module mod\_proxy\_balancer

|  |  |
| --- | --- |
| [**Description:**](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOJC%2BZFVPTXaCF0YV9hM%3D&b=1#Description) | [mod\_proxy](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1) extension for load balancing |
| [**Status:**](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOJC%2BZFVPTXaCF0YV9hM%3D&b=1#Status) | Extension |
| [**Module Identifier:**](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOJC%2BZFVPTXaCF0YV9hM%3D&b=1#ModuleIdentifier) | proxy\_balancer\_module |
| [**Source File:**](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOJC%2BZFVPTXaCF0YV9hM%3D&b=1#SourceFile) | mod\_proxy\_balancer.c |
| [**Compatibility:**](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOJC%2BZFVPTXaCF0YV9hM%3D&b=1#Compatibility) | Available in version 2.1 and later |

### Summary

This module *requires* the service of [mod\_proxy](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1) and it provides load balancing for all the supported protocols. The most important ones are:

* HTTP, using mod\_proxy\_http
* FTP, using mod\_proxy\_ftp
* AJP13, using mod\_proxy\_ajp
* WebSocket, using mod\_proxy\_wstunnel

The Load balancing scheduler algorithm is not provided by this module but from other ones such as:

* mod\_lbmethod\_byrequests
* mod\_lbmethod\_bytraffic
* mod\_lbmethod\_bybusyness
* mod\_lbmethod\_heartbeat

Thus, in order to get the ability of load balancing, [mod\_proxy](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1), [mod\_proxy\_balancer](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXUqUWEIA9Rw50m218pER&b=1) and at least one of load balancing scheduler algorithm modules have to be present in the server.

### Warning

Do not enable proxying until you have [secured your server](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#access). Open proxy servers are dangerous both to your network and to the Internet at large.Load balancer scheduler algorithm

At present, there are 4 load balancer scheduler algorithms available for use: Request Counting (mod\_lbmethod\_byrequests), Weighted Traffic Counting (mod\_lbmethod\_bytraffic), Pending Request Counting (mod\_lbmethod\_bybusyness) and Heartbeat Traffic Counting (mod\_lbmethod\_heartbeat). These are controlled via the lbmethod value of the Balancer definition. See the [ProxyPass](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxypass) directive for more information, especially regarding how to configure the Balancer and BalancerMembers.

## Load balancer stickyness

The balancer supports stickyness. When a request is proxied to some back-end, then all following requests from the same user should be proxied to the same back-end. Many load balancers implement this feature via a table that maps client IP addresses to back-ends. This approach is transparent to clients and back-ends, but suffers from some problems: unequal load distribution if clients are themselves hidden behind proxies, stickyness errors when a client uses a dynamic IP address that changes during a session and loss of stickyness, if the mapping table overflows.

The module [mod\_proxy\_balancer](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXUqUWEIA9Rw50m218pER&b=1) implements stickyness on top of two alternative means: cookies and URL encoding. Providing the cookie can be either done by the back-end or by the Apache web server itself. The URL encoding is usually done on the back-end.

## Examples of a balancer configuration

Before we dive into the technical details, here's an example of how you might use [mod\_proxy\_balancer](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXUqUWEIA9Rw50m218pER&b=1) to provide load balancing between two back-end servers:

<**Proxy** "balancer://mycluster">

**BalancerMember** "<http://192.168.1.50:80>"

**BalancerMember** "<http://192.168.1.51:80>"

</**Proxy**>

**ProxyPass** "/test" "balancer://mycluster"

**ProxyPassReverse** "/test" "balancer://mycluster"

Another example of how to provide load balancing with stickyness using [mod\_headers](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLq6ZBlPQWeFF0YV9hM%3D&b=1), even if the back-end server does not set a suitable session cookie:

**Header** add Set-Cookie "ROUTEID=.%{BALANCER\_WORKER\_ROUTE}e;

path=/" env=BALANCER\_ROUTE\_CHANGED

<**Proxy** "balancer://mycluster">

**BalancerMember** "<http://192.168.1.50:80>" route=1

**BalancerMember** "<http://192.168.1.51:80>" route=2

**ProxySet** stickysession=ROUTEID

</**Proxy**>

**ProxyPass** "/test" "balancer://mycluster"

**ProxyPassReverse** "/test" "balancer://mycluster"

## Exported Environment Variables

At present there are 6 environment variables exported:

**BALANCER\_SESSION\_STICKY**

This is assigned the stickysession value used for the current request. It is the name of the cookie or request parameter used for sticky sessions

**BALANCER\_SESSION\_ROUTE**

This is assigned the route parsed from the current request.

**BALANCER\_NAME**

This is assigned the name of the balancer used for the current request. The value is something like balancer://foo.

**BALANCER\_WORKER\_NAME**

This is assigned the name of the worker used for the current request. The value is something like <http://hostA:1234>.

**BALANCER\_WORKER\_ROUTE**

This is assigned the route of the worker that will be used for the current request.

**BALANCER\_ROUTE\_CHANGED**

This is set to 1 if the session route does not match the worker route (BALANCER\_SESSION\_ROUTE != BALANCER\_WORKER\_ROUTE) or the session does not yet have an established route. This can be used to determine when/if the client needs to be sent an updated route when sticky sessions are used.

## Enabling Balancer Manager Support

This module *requires* the service of [mod\_status](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqhdRlfUWbYUVoM9w%3D%3D&b=1). Balancer manager enables dynamic update of balancer members. You can use balancer manager to change the balance factor of a particular member, or put it in the off line mode.

Thus, in order to get the ability of load balancer management, [mod\_status](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqhdRlfUWbYUVoM9w%3D%3D&b=1) and [mod\_proxy\_balancer](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXUqUWEIA9Rw50m218pER&b=1) have to be present in the server.

To enable load balancer management for browsers from the example.com domain add this code to your httpd.conf configuration file

<**Location** "/balancer-manager">

**SetHandler** balancer-manager

**Require** host example.com

</**Location**>

You can now access load balancer manager by using a Web browser to access the page <http://your.server.name/balancer-manager>. Please note that only Balancers defined outside of <Location ...> containers can be dynamically controlled by the Manager.

## Details on load balancer stickyness

When using cookie based stickyness, you need to configure the name of the cookie that contains the information about which back-end to use. This is done via the stickysession attribute added to either [ProxyPass](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxypass) or [ProxySet](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxyset). The name of the cookie is case-sensitive. The balancer extracts the value of the cookie and looks for a member worker with route equal to that value. The route must also be set in either [ProxyPass](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxypass) or [ProxySet](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxyset). The cookie can either be set by the back-end, or as shown in the above [example](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXUqUWEIA9Rw50m218pER&b=1&f=norefer) by the Apache web server itself.

Some back-ends use a slightly different form of stickyness cookie, for instance Apache Tomcat. Tomcat adds the name of the Tomcat instance to the end of its session id cookie, separated with a dot (.) from the session id. Thus if the Apache web server finds a dot in the value of the stickyness cookie, it only uses the part behind the dot to search for the route. In order to let Tomcat know about its instance name, you need to set the attribute jvmRoute inside the Tomcat configuration file conf/server.xml to the value of the route of the worker that connects to the respective Tomcat. The name of the session cookie used by Tomcat (and more generally by Java web applications based on servlets) is JSESSIONID (upper case) but can be configured to something else.

The second way of implementing stickyness is URL encoding. The web server searches for a query parameter in the URL of the request. The name of the parameter is specified again using stickysession. The value of the parameter is used to lookup a member worker with route equal to that value. Since it is not easy to extract and manipulate all URL links contained in responses, generally the work of adding the parameters to each link is done by the back-end generating the content. In some cases it might be feasible doing this via the web server using [mod\_substitute](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqhdBpYUHyCTFoEtRcozS8%3D&b=1) or [mod\_sed](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqhZBwFTGGbVQ%3D%3D&b=1). This can have negative impact on performance though.

The Java standards implement URL encoding slightly different. They use a path info appended to the URL using a semicolon (;) as the separator and add the session id behind. As in the cookie case, Apache Tomcat can include the configured jvmRoute in this path info. To let Apache find this sort of path info, you neet to set scolonpathdelim to On in [ProxyPass](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxypass) or [ProxySet](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLqicxdTXTueTUMN&b=1#proxyset).

Finally you can support cookies and URL encoding at the same time, by configuring the name of the cookie and the name of the URL parameter separated by a vertical bar (|) as in the following example:

**ProxyPass** "/test" "balancer://mycluster" stickysession=JSESSIONID|jsessionid scolonpathdelim=On

<**Proxy** "balancer://mycluster">

**BalancerMember** "<http://192.168.1.50:80>" route=node1

**BalancerMember** "<http://192.168.1.51:80>" route=node2

</**Proxy**>

If the cookie and the request parameter both provide routing information for the same request, the information from the request parameter is used.

## Troubleshooting load balancer stickyness

If you experience stickyness errors, e.g. users lose their application sessions and need to login again, you first want to check whether this is because the back-ends are sometimes unavailable or whether your configuration is wrong. To find out about possible stability problems with the back-ends, check your Apache error log for proxy error messages.

To verify your configuration, first check, whether the stickyness is based on a cookie or on URL encoding. Next step would be logging the appropriate data in the access log by using an enhanced [LogFormat](https://eu0.proxysite.com/process.php?d=B1tRzFe3NmbztMLZBIj4Cfw9wXCegmK3rpQvlteeaRno2c7iOLq%2Bbh90R3qYX0cGtRcozS8%3D&b=1#logformat). The following fields are useful:

**%{MYCOOKIE}C**

The value contained in the cookie with name MYCOOKIE. The name should be the same given in the stickysession attribute.

**%{Set-Cookie}o**

This logs any cookie set by the back-end. You can track, whether the back-end sets the session cookie you expect, and to which value it is set.

**%{BALANCER\_SESSION\_STICKY}e**

The name of the cookie or request parameter used to lookup the routing information.

**%{BALANCER\_SESSION\_ROUTE}e**

The route information found in the request.

**%{BALANCER\_WORKER\_ROUTE}e**

The route of the worker chosen.

**%{BALANCER\_ROUTE\_CHANGED}e**

Set to 1 if the route in the request is different from the route of the worker, i.e. the request couldn't be handled sticky.

Common reasons for loss of session are session timeouts, which are usually configurable on the back-end server.

The balancer also logs detailed information about handling stickyness to the error log, if the log level is set to debug or higher. This is an easy way to troubleshoot stickyness problems, but the log volume might be to high for production servers under high load.