LoxBerry::Web::loglist_html

$html = loglist_html(%params);

Returns HTML of the logfile list to directly show the overview of your logfiles in your plugin. It is thought to show the list on navigating a Logfiles tab ($navbar), but you can integrate the HTML as ever you like. .

> ⚠ **LoxBerry Compatibility**
>
> This function is available starting with LoxBerry V1.2.5. If you use that function, set this minimum version in your plugin.cfg.

This feature only works with logfiles creates with the module LoxBerry::Log (Perl) or LBLog (PHP). It does not work for logfiles created with own logging features (as the list is read from the LoxBerry logfile database).

The function returns a preformatted html string of the list of your logfiles. Additionally, with the NAME parameter you can filter to a specific logfile group name (e.g. 'Daemon') to only show that logfiles.

To give parameters to the function, use named parameters.

> ⓘ **get_logs**
>
> If you want to render your logfiles with your own html style and template, you instead can use LoxBerry::Log::get_logs that returns an array of hashes, including the "raw" data from the log database. This array of hashes directly can be used with HTML::Template.

Parameter

> ⓘ **Case sensitivity**
>
> All parameter names are case-sensitive and written in capital letters.

Calling loglist_html in a plugin without parameters will return the html with all logs of the current plugin (=package).

Parameter	Optional	Usage
NAME	x	NAME is the log group name used with LoxBerry::Log. If you have created a logfile with LoxBerry::Log::new(name => 'Daemon'), use NAME => 'Daemon' to list only logfiles of that group. If you omit the NAME parameter, the list includes all logfiles of your plugin, independent of the log group name.
PACKAGE	x	Package is not required in the case of a plugin. By default, this is (like with LoxBerry::Log) the $lbpplugindir-Variable. *Core-Developers* have to use the PACKAGE parameter.

Return value

Returns a string that is the HTML code of the logfile list. Print the result of the function to your webpage, or send it to your HTML::Template.

Returns *undef*, if the call failed.

Usage

```
use LoxBerry::Web;

my $html = loglist_html ( NAME => 'daemon' );
print $html;

# Or send it to HTML::Template
my $html = loglist_html ( NAME => 'daemon' );
$template->param('loglist', $html);

# Core-developers: Also use the PACKAGE parameter
my $html = loglist_url( PACKAGE => 'Plugins Update', NAME => 'check' );
print $html;
```