

1 Apache::HookRun - Perl API for XXX

1.1 Synopsis

```
use Apache::HookRun ();
```

META: to be completed

1.2 Description

META: to be completed

1.3 API

Apache::HookRun provides the following functions and/or methods:

1.3.1 *die*

META: Autogenerated - needs to be reviewed/completed

Kill the current request

```
$r->die($type);
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **arg2: \$type (XXX)**

Why the request is dieing

- **ret: no return value**

1.3.2 *invoke_handler*

META: Autogenerated - needs to be reviewed/completed

Run the handler phase of each module until a module accepts the responsibility of serving the request

```
$ret = $r->invoke_handler();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

The status of the current request

1.3.3 run_access_checker

META: Autogenerated - needs to be reviewed/completed

This hook is used to apply additional access control to this resource. It runs **before** a user is authenticated, so this hook is really to apply additional restrictions independent of a user. It also runs independent of 'Require' directive usage.

```
$ret = $r->run_access_checker();
```

- **arg1: \$r (Apache::RequestRec)**

the current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.4 run_auth_checker

META: Autogenerated - needs to be reviewed/completed

This hook is used to check to see if the resource being requested is available for the authenticated user (`r->user` and `r->ap_auth_type`). It runs after the `access_checker` and `check_user_id` hooks. Note that it will **only** be called if Apache determines that access control has been applied to this resource (through a 'Require' directive).

```
$ret = $r->run_auth_checker();
```

- **arg1: \$r (Apache::RequestRec)**

the current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.5 run_check_user_id

META: Autogenerated - needs to be reviewed/completed

This hook is used to analyze the request headers, authenticate the user, and set the user information in the request record (`r->user` and `r->ap_auth_type`). This hook is only run when Apache determines that authentication/authorization is required for this resource (as determined by the 'Require' directive). It runs after the `access_checker` hook, and before the `auth_checker` hook.

1.3.6 run_create_request

```
$ret = $r->run_check_user_id();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.6 run_create_request

META: Autogenerated - needs to be reviewed/completed

Gives modules a chance to create their request_config entry when the request is created.

```
$ret = $r->run_create_request();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.7 run_fixups

META: Autogenerated - needs to be reviewed/completed

Allows modules to perform module-specific fixing of header fields. This is invoked just before any content-handler

```
$ret = $r->run_fixups();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.8 run_handler

META: Autogenerated - needs to be reviewed/completed

Run the handler functions for each module

```
$ret = $r->run_handler();
```

- **arg1: \$r (Apache::RequestRec)**

The request_rec

- **ret: \$ret (integer)**

non-wildcard handlers should HOOK_MIDDLE, wildcard HOOK_LAST

1.3.9 run_header_parser

META: Autogenerated - needs to be reviewed/completed

Run the header parser functions for each module

```
$ret = $r->run_header_parser();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK or Apache::DECLINED

1.3.10 run_log_transaction

META: Autogenerated - needs to be reviewed/completed

This hook allows modules to perform any module-specific logging activities over and above the normal server things.

```
$ret = $r->run_log_transaction();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.11 run_map_to_storage

META: Autogenerated - needs to be reviewed/completed

This hook allow modules to set the per_dir_config based on their own context (such as <Proxy> sections) and responds to contextless requests such as TRACE that need no security or filesystem mapping. based on the filesystem.

```
$ret = $r->run_map_to_storage();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::DONE (or Apache::HTTP_) if this contextless request was just fulfilled (such as TRACE), OK if this is not a file, and Apache::DECLINED if this is a file. The core map_to_storage (HOOK_RUN_LAST) will directory_walk and file_walk the r->filename.

1.3.12 run_post_read_request

META: Autogenerated - needs to be reviewed/completed

post_read_request --- run right after read_request or internal_redirect, and not run during any subrequests. This hook allows modules to affect the request immediately after the request has been read, and before any other phases have been processes. This allows modules to make decisions based upon the input header fields

```
$ret = $r->run_post_read_request();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK or Apache::DECLINED

1.3.13 run_translate_name

META: Autogenerated - needs to be reviewed/completed

This hook allow modules an opportunity to translate the URI into an actual filename. If no modules do anything special, the server's default rules will be followed.

```
$ret = $r->run_translate_name();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.3.14 run_type_checker

META: Autogenerated - needs to be reviewed/completed

This routine is called to determine and/or set the various document type information bits, like Content-type (via `r->content_type`), language, etc.

```
$ret = $r->run_type_checker();
```

- **arg1: \$r (Apache::RequestRec)**

the current request

- **ret: \$ret (integer)**

Apache::OK, Apache::DECLINED, or Apache::HTTP_...

1.4 See Also

mod_perl 2.0 documentation.

1.5 Copyright

mod_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 1.1.

1.6 Authors

The mod_perl development team and numerous contributors.

Table of Contents:

1	Apache::HookRun - Perl API for XXX	1
1.1	Synopsis	2
1.2	Description	2
1.3	API	2
1.3.1	die	2
1.3.2	invoke_handler	2
1.3.3	run_access_checker	3
1.3.4	run_auth_checker	3
1.3.5	run_check_user_id	3
1.3.6	run_create_request	4
1.3.7	run_fixups	4
1.3.8	run_handler	4
1.3.9	run_header_parser	5
1.3.10	run_log_transaction	5
1.3.11	run_map_to_storage	6
1.3.12	run_post_read_request	6
1.3.13	run_translate_name	6
1.3.14	run_type_checker	7
1.4	See Also	7
1.5	Copyright	7
1.6	Authors	7