1 Apache::RequestRec - Perl API for Apache request record accessors

1.1 Synopsis

```
use Apache::RequestRec ();

sub handler{
    my $r = shift;
    ...
    my $auth_type = $r->auth_type;
    ...
    my $s = $r->server;
    my $dir_config = $r->dir_config;
}
```

META: to be completed

1.2 Description

`Apache::RequestRec` provides the Perl API for Apache request object.

1.3 API

`Apache::RequestRec` provides the following functions and/or methods:

1.3.1 `proxyreq`

META: Autogenerated - needs to be reviewed/completed

```
$ret = $r->proxyreq($val);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **arg2: `$val` (integer)**
- **ret: `$ret` (integer)**

1.3.2 `pool`

META: Autogenerated - needs to be reviewed/completed

The pool associated with the request

```
$p = $r->pool();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$p` (`APR::Pool`)**

1.3.3 *`connection`*

META: Autogenerated - needs to be reviewed/completed

The connection record to the client

```
$c = $r->connection();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$c` (`Apache::Connection`)**

1.3.4 *`server`*

Get the `Apache::Server` object for the server the request `$r` is running under.

```
$s = $r->server();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$s` (`Apache::Server`)**

1.3.5 *`next`*

META: Autogenerated - needs to be reviewed/completed

Pointer to the redirected request if this is an external redirect

```
$next_r = $r->next();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$next_r` (`Apache::RequestRec`)**

1.3.6 *`prev`*

META: Autogenerated - needs to be reviewed/completed

Pointer to the previous request if this is an internal redirect

```
$prev_r = $r->prev();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$prev_r` (`Apache::RequestRec`)**

1.3.7 *`main`*

Get the main request record

```
$main_r = $r->main();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$main_r` (`Apache::RequestRec`)**

  If the current request is a sub-request, this method returns a blessed reference to the main request structure. If the current request is the main request, then this method returns undef.

  To figure out whether you are inside a main request or a sub-request/internal redirect, use `$r->is_initial_req`.

1.3.8 `the_request`

META: Autogenerated - needs to be reviewed/completed

First line of request

```
$request = $r->the_request();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$request` (string)**

1.3.9 `assbackwards`

META: Autogenerated - needs to be reviewed/completed

HTTP/0.9, "simple" request (e.g. GET /foo\n w/no headers)

```
$status = $r->assbackwards($newval);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **arg2: `$newval` (integer)**
- **ret: `$status` (integer)**

1.3.10 `header_only`

META: Autogenerated - needs to be reviewed/completed

HEAD request, as opposed to GET

```
$status = $r->header_only();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$status` (integer)**

1.3.11 *`protocol`*

META: Autogenerated - needs to be reviewed/completed

Protocol string, as given to us, or HTTP/0.9

```
  $protocol = $r->protocol();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$protocl` (string)**

1.3.12 *`proto_num`*

META: Autogenerated - needs to be reviewed/completed

Protocol version number of protocol; 1.1 = 1001

```
  $proto_num = $r->proto_num();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$proto_num` (integer)**

1.3.13 *`hostname`*

META: Autogenerated - needs to be reviewed/completed

Host, as set by full URI or Host:

```
  $hostname = $r->hostname();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$hostname` (string)**

1.3.14 *`request_time`*

META: Autogenerated - needs to be reviewed/completed

Time when the request started

```
  $request_time = $r->request_time();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$request_time` (number)**

1.3.15 `status_line`

META: Autogenerated - needs to be reviewed/completed

Status line, if set by script

```
  $status_line = $r->status_line();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$status_line` (string)**

1.3.16 `status`

META: Autogenerated - needs to be reviewed/completed

Get/set status line

```
  $status = $r->status($new_status);
  $status = $r->status();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$new_status` (integer)**

  If `$new_status` is passed the new status is assigned.

- **ret: `$newval` (integer)**

  If `$new_status` is passed the old status is returned.

1.3.17 `method`

META: Autogenerated - needs to be reviewed/completed

Request method (eg. GET, HEAD, POST, etc.)

```
  $method = $r->method();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$method` (string)**

1.3.18 `method_number`

META: Autogenerated - needs to be reviewed/completed

Apache::M_GET, Apache::M_POST, etc.

```
$methno = $r->method_number();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$methno` (integer)**

1.3.19 `allowed`

META: Autogenerated - needs to be reviewed/completed

'allowed' is a bitvector of the allowed methods.

```
$allowed = $r->allowed();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$allowed` (number)**

A handler must ensure that the request method is one that it is capable of handling. Generally modules should DECLINE any request methods they do not handle. Prior to aborting the handler like this the handler should set r->allowed to the list of methods that it is willing to handle. This bitvector is used to construct the "Allow:" header required for OPTIONS requests, and HTTP_METHOD_NOT_ALLOWED and HTTP_NOT_IMPLEMENTED status codes.

Since the default_handler deals with OPTIONS, all modules can usually decline to deal with OPTIONS. TRACE is always allowed, modules don't need to set it explicitly.

Since the default_handler will always handle a GET, a module which does *not* implement GET should probably return HTTP_METHOD_NOT_ALLOWED. Unfortunately this means that a Script GET handler can't be installed by mod_actions.

1.3.20 `allowed_xmethods`

META: Autogenerated - needs to be reviewed/completed

Array of extension methods

```
$array = $r->allowed_xmethods();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$array` (*`APR::ArrayHeader`*)**

1.3.21 `allowed_methods`

META: Autogenerated - needs to be reviewed/completed

List of allowed methods

```
  $list = $r->allowed_methods();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$list` (*`Apache::MethodList`*)**

1.3.22 `bytes_sent`

META: Autogenerated - needs to be reviewed/completed

body byte count, for easy access

```
  $bytes_sent = $r->bytes_sent();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$bytes_sent` (integer)**

1.3.23 `mtime`

META: Autogenerated - needs to be reviewed/completed

Last modified time of the requested resource

```
  $mtime = $r->mtime($new_mtime);
  $mtime = $r->mtime();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$new_mtime` (number)**
- **ret: `$mtime` (number)**

1.3.24 `remaining`

META: Autogenerated - needs to be reviewed/completed

Remaining bytes left to read from the request body

```
  $bytes = $r->remaining();
```

META: I think this method is not needed if the deprecated client_block methods aren’t used, (and plain $r-<read() is used instead).

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$bytes` (integer)**

1.3.25 *`headers_in`*

META: Autogenerated - needs to be reviewed/completed

```
$headers_in = $r->headers_in();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$headers_in` (`APR::Table`)**

1.3.26 *`headers_out`*

META: Autogenerated - needs to be reviewed/completed

MIME header environment for the response

```
$headers_out = $r->headers_out();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$headers_out` (`APR::Table`)**

1.3.27 *`err_headers_out`*

META: Autogenerated - needs to be reviewed/completed

MIME header environment for the response, printed even on errors and persist across internal redirects

```
$err_headers_out = $r->err_headers_out();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **err: `$err_headers_out` (`APR::Table`)**

The difference between headers_out and err_headers_out is that the latter are printed even on error, and persist across internal redirects (so the headers printed for ErrorDocument handlers will have them).

1.3.28 *`notes`*

META: Autogenerated - needs to be reviewed/completed

Notes from one module to another

```
$notes = $r->notes();
$notes = $r->notes($new_notes);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$new_notes` (`APR::Table`)**
- **ret: `$notes` (`APR::Table`)**

The 'notes' is for notes from one module to another, with no other set purpose in mind...

1.3.29 handler

META: Autogenerated - needs to be reviewed/completed

```
$handler = $r->handler();
$handler = $r->handler($new_handler);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$new_handler` (string)**
- **ret: `$handler` (string)**

1.3.30 content_encoding

META: Autogenerated - needs to be reviewed/completed

```
$ce = $r->content_encoding();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$ce` (string)**

1.3.31 content_languages

META: Autogenerated - needs to be reviewed/completed

Array of strings representing the content languages

```
$array_header = $r->content_languages();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$array_header` (*`APR::ArrayHeader`*)**

1.3.32 user

META: Autogenerated - needs to be reviewed/completed

If an authentication check was made, this gets set to the user name.

```
$r->user($user);
$user = $r->user();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$user` (string)**
- **ret: `$user` (string)**

1.3.33 `ap_auth_type`

If an authentication check was made, get or set the *ap_auth_type* slot in the request record

```
$auth_type = $r->ap_auth_type();
$r->ap_auth_type($newval);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$newval` (string)**

  If this argument is passed then a new auth type is assigned. For example:

  ```
  $r->auth_type('Basic');
  ```

- **ret: `$auth_type` (string)**

  If `$newval` is passed, nothing is returned. Otherwise the current auth type is returned.

ap_auth_type holds the authentication type that has been negotiated between the client and server during the actual request. Generally, *ap_auth_type* is populated automatically when you call `$r->get_basic_auth_pw` so you don't really need to worry too much about it, but if you want to roll your own authentication mechanism then you will have to populate *ap_auth_type* yourself.

Note that `$r->ap_auth_type` was `$r->connection->auth_type` in the mod_perl 1.0 API.

1.3.34 `no_local_copy`

META: Autogenerated - needs to be reviewed/completed

There is no local copy of this response

```
$status = $r->no_local_copy();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$status` (integer)**

1.3.35 `unparsed_uri`

META: Autogenerated - needs to be reviewed/completed

The URI without any parsing performed

```
$unparsed_uri = $r->unparsed_uri();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$unparsed_uri` (string)**

1.3.36 *`uri`*

META: Autogenerated - needs to be reviewed/completed

The path portion of the URI

```
$uri = $r->uri();
$r->uri($uri);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$uri` (string)**
- **ret: `$uri` (string)**

1.3.37 *`filename`*

META: Autogenerated - needs to be reviewed/completed

The filename on disk corresponding to this response

```
$filename = $r->filename();
$r->filename($filename);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$filename` (string)**
- **ret: `$filename` (string)**

1.3.38 *`canonical_filename`*

META: Autogenerated - needs to be reviewed/completed

```
$canon_filename = $r->canonical_filename();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$canon_filename` (string)**

1.3.39 *`path_info`*

META: Autogenerated - needs to be reviewed/completed

The PATH_INFO extracted from this request

```
$path_info = $r->path_info();
$r->path_info($path_info);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$path_info` (string)**
- **ret: `$path_info` (string)**

1.3.40 ***`args`***

META: Autogenerated - needs to be reviewed/completed

The QUERY_ARGS extracted from this request

```
  $args = $r->args();
  $r->args($args);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **opt arg2: `$args` (string)**
- **ret: `$args` (string)**

1.3.41 ***`used_path_info`***

META: Autogenerated - needs to be reviewed/completed

Flag for the handler to accept or reject path_info on the current request. All modules should respect the AP_REQ_ACCEPT_PATH_INFO and AP_REQ_REJECT_PATH_INFO values, while AP_REQ_DEFAULT_PATH_INFO indicates they may follow existing conventions. This is set to the user's preference upon HOOK_VERY_FIRST of the fixups.

```
  $ret = $r->used_path_info($newval);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **arg2: `$newval` (integer)**

1.3.42 ***`per_dir_config`***

META: Autogenerated - needs to be reviewed/completed

These are config vectors, with one void* pointer for each module (the thing pointed to being the module's business). * Options set in config files, etc.

```
  $per_dir_config = $r->per_dir_config();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$per_dir_config` (*`Apache::ConfVector`*)**

1.3.43 ***`request_config`***

META: Autogenerated - needs to be reviewed/completed

Notes on *this* request

```
$ret = $r->request_config($newval);
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **arg2: `$newval` (*`Apache::ConfVector`*)**

1.3.44 `output_filters`

META: Autogenerated - needs to be reviewed/completed

A list of output filters to be used for this request

```
$output_filters = $r->output_filters();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$output_filters` (`Apache::Filter`)**

1.3.45 `input_filters`

META: Autogenerated - needs to be reviewed/completed

A list of input filters to be used for this request

```
$input_filters = $r->input_filters();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$input_filters` (`Apache::Filter`)**

1.3.46 `proto_output_filters`

META: Autogenerated - needs to be reviewed/completed

A list of protocol level output filters to be used for this request

```
$proto_output_filters = $r->proto_output_filters();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$proto_output_filters` (`Apache::Filter`)**

1.3.47 `proto_input_filters`

META: Autogenerated - needs to be reviewed/completed

A list of protocol level input filters to be used for this request

```
$proto_input_filters = $r->proto_input_filters();
```

- **arg1: `$r` (`Apache::RequestRec`)**
- **ret: `$proto_input_filters` (`Apache::Filter`)**

1.4 See Also

mod_perl 2.0 documentation.

1.5 Copyright

1.6 Authors

The mod_perl development team and numerous contributors.

Table of Contents: