Responses, redirects, errors and JSON

How a module view answers - the $Result array instead of a Response object, the pagelayout around it, redirects with redirectTo and safe redirect targets, error pages with handleError and eZError codes, JSON and other...

Source file: `doc/eng-us/help/web-framework-developers/responses-redirects-and-errors.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/web-framework-developers/responses-redirects-and-errors.md

This page is for developers who return `Response`, `JsonResponse`, `redirect()` or `abort(404)` from a controller. A module view returns an **array**, and the kernel turns it into the page. Redirects and errors go through the module object; non-HTML answers are written directly and end the request.

Contents

1. The result array
2. The pagelayout
3. Redirects
4. Errors
5. JSON and other formats
6. Headers and status codes
7. Translation table

The result array

Key	Meaning
`content`	the HTML of the view, usually `$tpl->fetch( 'design:<path>.tpl' )`
`path`	the breadcrumb: items `array( 'text' => ..., 'url' => ... )`, the last one with `'url' => false`
`title_path`	the page title's path, when it should differ from `path`
`pagelayout`	another pagelayout template for this view, or `false` for none
`navigation_part`, `ui_context`	how the admin designs frame the page

The full list, with the keys the kernel and content views add, is the module result reference.

The pagelayout

The siteaccess's `pagelayout.tpl` is the base layout. It receives `$module_result` and prints `{$module_result.content}` where the view's HTML goes. A view chooses another layout by naming its template (the user views set `$Result['pagelayout'] = 'loginpagelayout.tpl'`), or none with `false` (View results, templates and the pagelayout).

Redirects

Need	Call
Redirect to an address	`return $Module->redirectTo( '/expdemo/list' );`
Redirect to another view of the same module	`return $Module->redirectToView( 'list', array() );`
Redirect to an address taken from the request	check it first: `eZRedirectManager::safeURI( $http->postVariable( 'RedirectURI' ) )` returns the target or `false`
"Back to where you came from"	`eZRedirectManager::returnURI( $module, '/expdemo/list', eZRedirectManager::formReturnURIs() )`

Use POST, redirect, GET after every successful form submission. The safe redirect helpers only follow addresses of the site (Safe redirects).

Errors

PHP

```php
return $Module->handleError( eZError::KERNEL_NOT_FOUND, 'kernel' );       // not found
return $Module->handleError( eZError::KERNEL_ACCESS_DENIED, 'kernel' );   // access denied; anony
mous users get the login page
```

`eZError` defines further codes, among them `KERNEL_NOT_AVAILABLE`, `KERNEL_MOVED`, `KERNEL_MODULE_NOT_FOUND` and `KERNEL_FORM_TOKEN_REFUSED`. The error page is rendered by the `error` module with the siteaccess's design, so it can be styled like any other template. Errors that users report can be found by their reference in the logs (Finding an error by its reference).

Exceptions that escape a view are not turned into a 500 page by an exception listener you configure, as in Symfony or Laravel: handle expected failures in the view and return an error result; log unexpected ones with `eZDebug::writeError()` (Debugging, logging and errors).

JSON and other formats

For a small JSON answer from your own view, send it and end the request without the pagelayout:

PHP

```php
header( 'Content-Type: application/json; charset=utf-8' );
echo json_encode( array( 'ok' => true, 'data' => $data ) );
eZExecution::cleanExit();
```

Under Velocity, `cleanExit()` ends the request by an exception that the server handles; never wrap it in a `catch ( Exception $e )`, or the page is appended to your output. For APIs, prefer the existing layers: ezjscore server functions for your pages' JavaScript, the REST API for other programs (REST and JSON APIs).

Headers and status codes

- Set headers with PHP's `header()` before output; read request headers through `eZSys` or `$_SERVER` as prepared by the kernel.

- Page caching headers for anonymous visitors are settings, not code: `site.ini [HTTPHeaderSettings]` (HTTP caching).
- Request rules can answer with not found, forbidden, login or a redirect before your view runs (Middleware).

Translation table

Framework	Exponential
`return $this->render( 'item.html.twig', [...] )` / `return view( 'item', [...] )`	`$tpl->setVariable(); $Result['content'] = $tpl->fetch( 'design:...' );`
`return $this->redirectToRoute( 'list' )` / `return redirect()->route( 'list' )`	`return $Module->redirectTo( '/expdemo/list' );`
`throw $this->createNotFoundException()` / `abort( 404 )` / `raise Http404`	`return $Module->handleError( eZError::KERNEL_NOT_FOUND, 'kernel' );`
`throw new AccessDeniedException()` / `abort( 403 )`	`return $Module->handleError( eZError::KERNEL_ACCESS_DENIED, 'kernel' );`
`return new JsonResponse( $data )` / `response()->json( $data )`	`header(); echo json_encode(); eZExecution::cleanExit();`

Related pages

- Controllers: module views
- Custom modules and views
- Forms and CSRF protection
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.