Module result reference

Every key of $Result / $module_result found in the kernel - content, path, title_path, content_info and its 25 keys, view_parameters, pagelayout, navigation_part, left_menu, errorType and the rest - with the code that...

Source file: `doc/eng-us/help/template-language/module-result-reference.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/template-language/module-result-reference.md

A module view returns an array, `$Result`; the kernel adds a few keys and hands it to the pagelayout as `$module_result` (see Module view templates). This page lists every key the kernel's views, `eZModule::run()` and `ezpKernelWeb` set, found by searching the kernel for `$Result['...']` and `$moduleResult['...']`. The values in the examples are what `content/view/full/96` returned on a development installation.

Contents

1. Keys every view can set
2. Keys the kernel adds
3. Keys of content views
4. content_info
5. Keys of error results
6. Reading it in the pagelayout
7. Related pages

Keys every view can set

Key	Meaning	Set by
`content`	the HTML of the view, placed by the pagelayout with `{$module_result.content}`	every view
`path`	the breadcrumb: a list of `array( 'text' => ..., 'url' => ... )`, the last with `url` false; content views add `url_alias` and `node_id`	every view; `false` when a view sets none (`ezpKernelWeb`)
`title_path`	the same for the page title, when it should differ from `path`	content views, some others
`pagelayout`	another pagelayout template, or `false` for none	e.g. the user login views (`loginpagelayout.tpl`), `layout/set`, `content/treemenu` (`false`)
`navigation_part`	identifier of the admin navigation part (top menu tab)	views; else `default_navigation_part` of `module.php`
`left_menu`	a template for the left menu of the admin design	the setup logs views (`design:setup/logs_menu.tpl`)

Key	Meaning	Set by
`ui_context`	`edit`, `view`, `browse`, `administration`, `ajax`, `authentication` …; also taken from `module.php`	views or `module.php`
`requested_uri_string`	the URI to report as requested, when the view ran under another one	views that rerun
`rerun_uri`	makes the kernel run another URI instead (with `eZModule::STATUS_RERUN`)	views
`external_css`	whether the debug output uses an external stylesheet	views
`title`	a title for an error page (form token refusals)	the error view

Keys the kernel adds

Key	Value	Where
`is_default_navigation_part`	`true` when `navigation_part` came from `module.php`, else `false`	`eZModule::run()`, `lib/ezutils/classes/ezmodule.php`
`ui_context`	the module's UI context, when the view set none	`ezpKernelWeb::run()`
`ui_component`	the module's UI component (by default the module name)	`ezpKernelWeb::run()`
`is_mobile_device`, `mobile_device_alias`	the mobile device detection of the request	`ezpKernelWeb::run()`
`uri`	the request URI (`eZSys::requestURI()`)	`ezpKernelWeb::run()`, only when a pagelayout is rendered

Keys of content views

`eZNodeviewfunctions::generateNodeViewData()` (`kernel/classes/eznodeviewfunctions.php`) builds these for `content/view`:

Key	Value on a development installation (node 96)
`view_parameters`	`offset`, `year`, `month`, `day`, `namefilter`, plus every user parameter (see View parameters)
`section_id`	1
`node_id`	96
`content_info`	see below
`template_list`	the templates used to render the view (11 on a development installation), stored with the view cache so a changed template expires it
`cache_ttl`	`-1`, or the value of a template variable `$cache_ttl`; `0` adds `no_cache`
`no_cache`	`true` when a template set `cache_ttl` to 0

content_info

`content_info` lets the pagelayout know what the page shows without a fetch. All 25 keys, as a development installation returned them for node 96:

Key	Value on a development installation	Meaning
`object_id`	`94`	content object ID
`node_id`	`96`	node ID
`parent_node_id`	`92`	parent node ID
`class_id`	`44`	class ID
`class_identifier`	`'ng_article'`	class identifier
`remote_id`	`'media-o-958'`	object remote ID
`node_remote_id`	`'media-n-729'`	node remote ID
`offset`	`false`	the (`offset`) view parameter as an integer, or false
`viewmode`	`'full'`	view mode
`navigation_part_identifier`	`'ezcontentnavigationpart'`	navigation part
`node_depth`	`4`	depth of the node
`url_alias`	`'fitness/why-should-you-choose-strength-training'`	URL alias of the node
`current_language`	`'eng-US'`	language the object is shown in
`language_mask`	`2`	bit mask of the object's languages
`main_node_id`	`96`	main node of the object
`main_node_url_alias`	`false`	URL alias of the main node when this is not the main node and the user may read it
`persistent_variable`	`false`	what the view template put into `$persistent_variable` (details)
`class_group`	`false`	class group IDs when `EnableClassGroupOverride` is on
`state`	`array( 2 => 1, 47 => 80 )`	state group ID to state ID
`state_identifier`	an array of 2	`group/state` identifiers
`section_identifier`	`'standard'`	section identifier
`parent_class_id`	`45`	class ID of the parent's object
`parent_class_identifier`	`'ng_landing_page'`	class identifier of the parent's object
`parent_node_remote_id`	`'media-n-721'`	remote ID of the parent node
`parent_object_remote_id`	`'media-o-950'`	remote ID of the parent's object

When the page comes from the content view cache, the kernel restores the same `$Result` from the cache file and sets the override keys again from `content_info` (

`eZNodeviewfunctions::contentViewRetrieve()`), so `content_info` is available whether the page was cached or not.

Keys of error results

A view that fails calls `$Module->handleError()`, which runs the error view (`kernel/private/classes/views/error/view.php`). On a development installation, `content/view/full/99999999` returned:

Key	Value on a development installation
`errorType`	`'kernel'`
`errorNumber`	3, `eZError::KERNEL_NOT_AVAILABLE` (the numbers are constants of `eZError`, `kernel/classes/ezerror.php`: 1 access denied, 2 not found, 3 not available, 4 moved, 5 language not found, 6 form token refused, 20 module not found, 21 view not found, 22 module disabled)
`errorCode`	`'404'`
`errorMessage`	`'Not Found'`
`responseHeaders`	`'HTTP/1.1 404 Not Found'`, `'Status: 404 Not Found'`
`path`	`Error / kernel (3)`

`errorType` and `errorNumber` also become the override keys `error_type` and `error_number`, so error pages can have templates of their own (see Error templates).

Reading it in the pagelayout

Not run on a development installation: `$module_result` exists only while the kernel renders a pagelayout for a web request, so this example cannot run in the CLI example renderer. The values it reads are the ones listed in the tables above, which came from running the view.

Template

```
{* the class of the page, without fetching the node *}
{if is_set( $module_result.content_info )}
    <body class="class-{$module_result.content_info.class_identifier}">
{else}
    <body class="module-{$module_result.ui_component}">
{/if}

{* the breadcrumb *}
{foreach $module_result.path as $item}
    {if $item.url}<a href={cond( is_set( $item.url_alias ), $item.url_alias, $item.url )|ezurl}>{
$item.text|wash}</a>{else}{$item.text|wash}{/if}
    {delimiter} / {/delimiter}
{/foreach}

{$module_result.content}
```

`content_info` exists only for content views: test it with `is_set()` in a pagelayout that also serves other modules.

Related pages

- Module view templates
- View parameters
- The persistent variable, inside
- Module result and content info
- Breadcrumbs
- 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.