

# ez\_urlaliases

## Description

`ez_urlaliases` is not a real Twig helper, but a **special route name** for generating URLs for a location, from the given parameters.

## Prototype and Arguments

`path( eZ\Publish\API\Repository\Values\Content\Location|string name[, array parameters][, bool absolute] )`

| Argument name | Type  | Description   |
|---------------|---|---|
| name          | string   <code>eZ\Publish\API\Repository\Values\Content\Location</code> | The name of the route or a Location instance  |
| parameters    | array   | An hash of parameters: <ul style="list-style-type: none"><li>• <code>locationId</code></li><li>• <code>contentId</code> (as of 5.4 / 2014.11)</li></ul> |
| absolute      | boolean   | Whether to generate an absolute URL   |

## Working with Location

Linking to other locations is fairly easy and is done with `native path()` Twig helper (or `url()` if you want to generate absolute URLs). You just have to pass it the Location object and `path()` will generate the URLAlias for you.

```
{# Assuming "location" variable is a valid
eZ\Publish\API\Repository\Values\Content\Location object #}
<a href="{{ path( location ) }}">Some link to a location</a>
```

## I don't have the Location object

### Generating a link from a Location ID

```
<a href="{{ path( "ez_urlaliases", {"locationId": 123} ) }}">Some link to a location,
with its Id only</a>
```

### Generating a link from a Content ID

Link generation from `contentId` is available as of **5.4** / **2014.11**.

```
<a href="{{ path( "ez_urlaliases", {"contentId": 456} ) }}">Some link from a contentId</a>
```

**Important:** Links generated from a Content ID will point to its main location.

## Error management

For Location alias setup 301 redirect to Location's current URL when:

1. alias is history
2. alias is custom with forward flag true
3. requested URL is not case-sensitive equal with the one loaded

### Under the hood

In the backend, `path()` uses the Router to generate links.

This makes also easy to generate links from PHP, via the `router` service.