



PageDirectory

```
/**
```

```
    PageDirectory
```

```
        creates a searchable directory of pages organized by tabs and fi
```

```
Arguments:
```

```
path : str (optional, default: current page path)
```

```
    page path the the parent page where the subpages to be organized
```

```
most_popular : map (optional, default: none)
```

```
    show section for most popular pages that have a given tag
```

```
tag : str
```

```
    tag to used for the split between left and right columns for
```

```
    left pages have the tag, right pages do not
```

```
title : str
```

```
    title shown for left column
```

```
tabs : list (optional, default: empty list)
```

```
    list of maps describing each tab to show; map entries may contain
```

```
label : str
```

```
    label for the tab
```

```
key : str
```

```
    key used to identify the selected tab
```

```
featured_limit : num (optional, default: 15)
```

```
    number of pages to show in the "Featured" category
```

```
cache_prefix : str (optional, default: none)
```

```
    when set, caches the page directory for a given view using this l
```

```
tag_constraint : str -or- list (optional, default: none)
```

```
    only list pages that have all of the specified tags
```

```
*/
```

```
// read parameters
```

```

var path = $path ?? page.path;
var most_popular = $most_popular;
var tabs = $tabs ?? [ ];
var featured_limit = $featured_limit ?? 15;
var cache_prefix = $cache_prefix;
var tag_constraint = $tag_constraint;

// check how the current page is being invoked
var query = __request.args.q;
var view = __request.args.v ?? (query ? 'search' : 'featured');

// define variables
var render;
var cache_id;
var pages;
var constraint = 'type:wiki AND path:' .. string.searchescape(path) .. '
if(tag_constraint is str) {
    let constraint .= ' AND tag:"' .. string.escape(tag_constraint) .. '
} else if(tag_constraint is list) {
    let constraint .= ' AND ' .. string.join([ ' AND tag:"' .. string.e
}

// check if a search was requested
if(query) {
    if(path[-1] != '/') {
        let path = path .. '/';
    }
    let pages = wiki.getsearch(query, 1000, __, constraint);
} else {
    let pages = wiki.getsearch(constraint, 1000);
    if(cache_prefix) {
        let cache_id = cache_prefix .. '-cache-' .. view;
        let render = __env.webcache && webcache.fetch(cache_id);
    }
}

if(!render) {
    let render = (

        // remove pages starting with '*'
        let parent_page = wiki.getpage(path);
        let pages = [ p foreach var p in pages where !string.startswith(

        // show page count
        <p> wiki.localize("MindTouch.Templates.Controls.PageDirectory.en

```



```

switch(view) {
case 'featured':

    // show 'Most Popular' and 'Recently Added' entries
    <table width="100%" cellspacing="0" cellpadding="5" border="0">
        if(most_popular && most_popular.tag && most_popular.title)
            <tr valign="top">
                <td>
                    <h3> wiki.localize("MindTouch.Templates.Controls.
                    template("MindTouch/Controls/ListPages", { pages:
                </td>
                <td>
                    <h3> wiki.localize("MindTouch.Templates.Controls.
                    template("MindTouch/Controls/ListPages", { pages:
                </td>
            </tr>
        } else {
            <tr valign="top">
                <td colspan="2">
                    <h3> wiki.localize("MindTouch.Templates.Controls.
                    template("MindTouch/Controls/ListPages", { pages:
                </td>
            </tr>
        }
        <tr valign="top">
            <td>
                <h3> wiki.localize("MindTouch.Templates.Controls.
                template("MindTouch/Controls/ListPages", { pages:
            </td>
            <td>
                <h3> wiki.localize("MindTouch.Templates.Controls.
                template("MindTouch/Controls/ListPages", { pages:
            </td>
        </tr>
    </table>

case 'all':

    <h3> wiki.localize("MindTouch.Templates.Controls.PageDirecto
    template("MindTouch/Controls/ListPages", { pages: pages, sort:

case 'search':

    <h3> wiki.localize("MindTouch.Templates.Controls.PageDirecto

```

```
        template("MindTouch/Controls/ListPages", { pages: pages, sort:
default:
        <h3> wiki.localize("MindTouch.Templates.Controls.PageDirectory")
        template("MindTouch/Controls/TagDirectory", { pages: pages,
    }
    <br/><hr/><br/>
);
if(cache_id && __env.webcache) {
    webcache.store(cache_id, render);
} else {
    render;
}
} else {
    render;
}
```