```
/***

    PageDirectory
        creates a searchable directory of pages organized by tabs and
filtered by tags

    Arguments:

    path : str (optional, default: current page path)
        page path the the parent page where the subpages to be organized
are located

    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
most popular;
            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
the following fields

        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
key

    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.escape(tag) .. '"' foreach var tag in tag_constraint where tag is
str ], ' AND ');
}

// 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(p.name, '*') && p.parent.id == parent_page.id ];

        // show page count
        <p>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.entrycount", [
num.format(#pages, "#,##0") ]) </p>;
        <p> string.nbsp </p>
```

```
        // show directory based on current views
        <table width="100%" cellspacing="0" cellpadding="0" border="1">
            <tr>
                <td>
                    if(view == 'featured') {
                        <strong>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.viewfeatured")
</strong>;
                    } else {
                        web.link(page.uri,
wiki.localize("MindTouch.Templates.Controls.PageDirectory.viewfeatured"));
                    }
                </td>
                if(#tabs) {
                    foreach(var tab in tabs) {
                        <td>
                            if(view == tab.key) {
                                <strong>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.viewby", [
tab.label ]) </strong>;
                            } else {
                                web.link(page.uri & { v: tab.key },
wiki.localize("MindTouch.Templates.Controls.PageDirectory.viewby", [
tab.label ]));
                            }
                        </td>
                    }
                } else {
                    <td>
                        if(view == 'all') {
                            <strong>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.viewall")
</strong>;
                        } else {
                            web.link(page.uri & { v: 'all' },
wiki.localize("MindTouch.Templates.Controls.PageDirectory.viewall"));
                        }
                    </td>
                }
                <td>
                    <form method="get" action=(page.uri)>
                        if(view == 'search') {
                            <strong>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.search");
</strong>;
                        } else {
                            wiki.localize("MindTouch.Templates.Controls.PageDire
                        }
                         ;
```

```
                        <input name="q" type="text" value=(query ?? '') />
                        <input type="submit" value="Go" />
                    </form>
                </td>
            </tr>
        </table>

        switch(view) {
        case 'featured':

            // show 'Most Popular' and 'Recently Added' entries
            <table width="100%" cellspacing="0" cellpadding="5"
border="0" style="table-layout: fixed;">
                if(most_popular && most_popular.tag &&
most_popular.title) {
                    <tr valign="top">
                        <td>
                            <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.mostpopularfor",
[ most_popular.title ]); </h3>
                            template("MindTouch/Controls/ListPages", {
pages: [ p foreach var p in pages where p.tags[most_popular.tag] is not
nil ], sort: 'viewcount', reverse: true, limit: featured_limit, style:
'bullets' })
                        </td>
                        <td>
                            <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.mostpopularfor",
[
xml.text(wiki.localize("MindTouch.Templates.Controls.PageDirectory.other"))
]); </h3>
                            template("MindTouch/Controls/ListPages", {
pages: [ p foreach var p in pages where p.tags[most_popular.tag] is nil
], sort: 'viewcount', reverse: true, limit: featured_limit, style:
'bullets' })
                        </td>
                    </tr>
                } else {
                    <tr valign="top">
                        <td colspan="2">
                            <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.mostpopular" );
</h3>
                            template("MindTouch/Controls/ListPages", {
pages: pages, sort: 'viewcount', reverse: true, limit: featured_limit,
style: 'bullets' })
                        </td>
                    </tr>
                }
                <tr valign="top">
```

```
                    <td>
                        <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.recentlyadded")
</h3>
                        template("MindTouch/Controls/ListPages", { pages:
pages, sort: 'created', reverse: true, limit: featured_limit, style:
'bullets' })
                    </td>
                    <td>
                        <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.recentlyupdated")
</h3>
                        template("MindTouch/Controls/ListPages", { pages:
pages, sort: 'updated', reverse: true, limit: featured_limit, style:
'bullets' })
                    </td>
                </tr>
            </table>

        case 'all':

            <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.directoryfor",
[ string.tocamelcase(view) ]) </h3>
            template("MindTouch/Controls/ListPages", { pages: pages,
sort: 'title' });

        case 'search':

            <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.searchresults")
</h3>
            template("MindTouch/Controls/ListPages", { pages: pages,
sort: 'viewcount', reverse: true });
        default:

            <h3>
wiki.localize("MindTouch.Templates.Controls.PageDirectory.directoryfor",
[ string.tocamelcase(view) ]) </h3>
            template("MindTouch/Controls/TagDirectory", { pages: pages,
tagprefix: view, columns: 3, listpagesoptions: { sort: 'title', reverse:
false } })
        }

        <br/><hr/><br/>
    );
    if(cache_id && __env.webcache) {
        webcache.store(cache_id, render);
    } else {
        render;
```

```
    }
} else {
    render;
}
```