Inhaltsverzeichnis

1. Modul:Redirect/Doku 2
2. Modul:Redirect 5

Modul:Redirect/Doku

Dies ist die Dokumentationsseite für Modul:Redirect

Vorlage:Lua This module contains functions to find the target of a redirect page.

Inhaltsverzeichnis

1 Main 2
1.1 Examples 2
1.2 LuaMain 2
2 IsRedirect 3
2.1 Examples 3
2.2 LuaIsRedirect 3
3 GetTarget 3
4 See also 4

Main

The main function accepts the name of a single page. It determines if the page is a redirect; if so it looks up the page, extracts the target, and returns the target name as text. Its usage is `{{#invoke:redirect|main|page-name}}`. If `page-name` does not exist or is not a redirect then `page-name` is returned. If `page-name` is blank then blank is returned.

The function normally neither takes nor gives square brackets, so to show the result as a link use `[[{{#invoke:redirect|main|page-name}}]]`.

If the parameter *bracket* is given a nonblank value, brackets will be passed on if present.

Examples

- `{{#invoke:redirect|main|WP:AFC}}` → WP:AFC
- `{{#invoke:redirect|main|[[WP:AFD]]|bracket=yes}}` → WP:AFD
- `{{#invoke:redirect|main|Wikipedia:Articles for deletion}}` → wikipedia:Articles for deletion
- `{{#invoke:redirect|main|Wikipedia:Articles for rumination}}` → wikipedia:Articles for rumination

Note: WP:AFC and WP:AFD are both redirects, but Wikipedia:Articles for deletion is not, and Wikipedia:Articles for rumination does not exist.

LuaMain

If you want to use the main function from another Lua module, you may want to use the luaMain function. This provides the same functionality as the main function, but doesn't require that a frame object be available.

To use this, first load the module.

```
local mRedirect = require('Module:Redirect')
```

Then use the function with the following syntax:

```
mRedirect.luaMain(rname, bracket)
```

rname is the name of the redirect page, and if *bracket* is anything but false or nil, the module will produce a link rather than just a page name.

IsRedirect

The isRedirect function is used from wiki pages to find out if a given page is a redirect or not. If the page is a redirect, the function returns "yes", and if not the output is blank. Its usage is `{{#invoke:redirect|isRedirect|page-name}}`.

Examples

- `{{#invoke:redirect|isRedirect|WP:AFC}}` →
- `{{#invoke:redirect|isRedirect|Wikipedia:Articles for deletion}}` →
- `{{#invoke:redirect|isRedirect|Wikipedia:Articles for rumination}}` →

LuaIsRedirect

The isRedirect can be accessed from other Lua modules in the same way as the main function:

```
local mRedirect = require('Module:Redirect')
mRedirect.luaIsRedirect(rname)
```

GetTarget

The getTarget function fetches the target page name of a redirect page, and is only usable from Lua. If the page specified is a redirect, then the target is returned; otherwise the function returns nil.

To use it, first load the module.

```
local mRedirect = require('Module:Redirect')
```

Then use the function with the following syntax:

```
mRedirect.getTarget(page)
```

page can be either the name of the redirect page as a string, or a mw.title object.

See also

- phab:T68974 - Provide a standard way to get the target of a redirect page
- mw:Extension:Scribunto/Lua reference manual#Title objects, the Scribunto interface for determining a page's redirect status
- Template:Submit an edit request, which uses this module

Modul:Redirect

Vorlage:Lua This module contains functions to find the target of a redirect page.

Inhaltsverzeichnis

1 Main 5
1.1 Examples 5
1.2 LuaMain 5
2 IsRedirect 6
2.1 Examples 6
2.2 LuaIsRedirect 6
3 GetTarget 6
4 See also 6

Main

The main function accepts the name of a single page. It determines if the page is a redirect; if so it looks up the page, extracts the target, and returns the target name as text. Its usage is `{{#invoke:redirect|main|page-name}}`. If `page-name` does not exist or is not a redirect then `page-name` is returned. If `page-name` is blank then blank is returned.

The function normally neither takes nor gives square brackets, so to show the result as a link use `[[{{#invoke:redirect|main|page-name}}]]`.

If the parameter *bracket* is given a nonblank value, brackets will be passed on if present.

Examples

- `{{#invoke:redirect|main|WP:AFC}}` → WP:AFC
- `{{#invoke:redirect|main|[[WP:AFD]]|bracket=yes}}` → WP:AFD
- `{{#invoke:redirect|main|Wikipedia:Articles for deletion}}` → wikipedia:Articles for deletion
- `{{#invoke:redirect|main|Wikipedia:Articles for rumination}}` → wikipedia:Articles for rumination

Note: WP:AFC and WP:AFD are both redirects, but Wikipedia:Articles for deletion is not, and Wikipedia:Articles for rumination does not exist.

LuaMain

If you want to use the main function from another Lua module, you may want to use the luaMain function. This provides the same functionality as the main function, but doesn't require that a frame object be available.

To use this, first load the module.

```
local mRedirect = require('Module:Redirect')
```

Then use the function with the following syntax:

```
mRedirect.luaMain(rname, bracket)
```

rname is the name of the redirect page, and if *bracket* is anything but false or nil, the module will produce a link rather than just a page name.

IsRedirect

The isRedirect function is used from wiki pages to find out if a given page is a redirect or not. If the page is a redirect, the function returns "yes", and if not the output is blank. Its usage is `{{#invoke:redirect|isRedirect|page-name}}`.

Examples

- `{{#invoke:redirect|isRedirect|WP:AFC}}` →
- `{{#invoke:redirect|isRedirect|Wikipedia:Articles for deletion}}` →
- `{{#invoke:redirect|isRedirect|Wikipedia:Articles for rumination}}` →

LuaIsRedirect

The isRedirect can be accessed from other Lua modules in the same way as the main function:

```
local mRedirect = require('Module:Redirect')
mRedirect.luaIsRedirect(rname)
```

GetTarget

The getTarget function fetches the target page name of a redirect page, and is only usable from Lua. If the page specified is a redirect, then the target is returned; otherwise the function returns nil.

To use it, first load the module.

```
local mRedirect = require('Module:Redirect')
```

Then use the function with the following syntax:

```
mRedirect.getTarget(page)
```

page can be either the name of the redirect page as a string, or a mw.title object.

See also

- phab:T68974 - Provide a standard way to get the target of a redirect page
- mw:Extension:Scribunto/Lua reference manual#Title objects, the Scribunto interface for determining a page's redirect status
- Template:Submit an edit request, which uses this module

```
-- This module provides functions for getting the target of a redirect page.

local p = {}

-- Gets a mw.title object, using pcall to avoid generating script errors if we
-- are over the expensive function count limit (among other possible causes).
local function getTitle(...)
	local success, titleObj = pcall(mw.title.new, ...)
	if success then
		return titleObj
	else
		return nil
	end
end

-- Gets the name of a page that a redirect leads to, or nil if it isn't a
-- redirect.
function p.getTargetFromText(text)
	local target = string.match(
		text,
		"^%s*#[Rr][Ee][Dd][Ii][Rr][Ee][Cc][Tt]%s*:?%s*%[%[([^%[%]|]-)%]%]
	) or string.match(
		text,
		"^%s*#[Rr][Ee][Dd][Ii][Rr][Ee][Cc][Tt]%s*:?%s*%[%[([^%[%]|]-)|[^%
	)
	return target and mw.uri.decode(target, 'PATH')
end

-- Gets the target of a redirect. If the page specified is not a redirect,
-- returns nil.
function p.getTarget(page, fulltext)
	-- Get the title object. Both page names and title objects are allowed
	-- as input.
	local titleObj
	if type(page) == 'string' or type(page) == 'number' then
		titleObj = getTitle(page)
	elseif type(page) == 'table' and type(page.getContent) == 'function' ther
		titleObj = page
	else
		error(string.format(
			"bad argument #1 to 'getTarget'"
				.. " (string, number, or title object expected, g
			type(page)
		), 2)
	end
	if not titleObj then
		return nil
	end
	local targetTitle = titleObj.redirectTarget
	if targetTitle then
		if fulltext then
			return targetTitle.fullText
		else
			return targetTitle.prefixedText
		end
```

```
        else
                return nil
        end
end

--[[
-- Given a single page name determines what page it redirects to and returns the
-- target page name, or the passed page name when not a redirect. The passed
-- page name can be given as plain text or as a page link.
--
-- Returns page name as plain text, or when the bracket parameter is given, as a
-- page link. Returns an error message when page does not exist or the redirect
-- target cannot be determined for some reason.
--]]
function p.luaMain(rname, bracket, fulltext)
        if type(rname) ~= "string" or not rname:find("%S") then
                return nil
        end
        bracket = bracket and "[[%s]]" or "%s"
        rname = rname:match("%[%[(.+)%]%]") or rname
        local target = p.getTarget(rname, fulltext)
        local ret = target or rname
        ret = getTitle(ret)
        if ret then
                if fulltext then
                        ret = ret.fullText
                else
                        ret = ret.prefixedText
                end
                return bracket:format(ret)
        else
                return nil
        end
end

-- Provides access to the luaMain function from wikitext.
function p.main(frame)
        local args = require('Module:Arguments').getArgs(frame, {frameOnly = true
        return p.luaMain(args[1], args.bracket, args.fulltext) or ''
end

-- Returns true if the specified page is a redirect, and false otherwise.
function p.luaIsRedirect(page)
        local titleObj = getTitle(page)
        if not titleObj then
                return false
        end
        if titleObj.isRedirect then
                return true
        else
                return false
        end
end

-- Provides access to the luaIsRedirect function from wikitext, returning 'yes'
-- if the specified page is a redirect, and the blank string otherwise.
function p.isRedirect(frame)
        local args = require('Module:Arguments').getArgs(frame, {frameOnly = true
        if p.luaIsRedirect(args[1]) then
                return 'yes'
```

```
        else
                return ''
        end
end

return p
```