

Toggle menu
Blue Gold Program Wiki

Navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help about MediaWiki](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)

Personal tools

- [Log in](#)

personal-extra

Toggle search

Search

Random page

Views

- [View](#)
- [View source](#)
- [History](#)
- [PDF Export](#)

Actions

Module:Redirect

From Blue Gold Program Wiki

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Documentation for this module may be created at [Module:Redirect/doc](#)

```

-- 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)
    return 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*%[[([^\%]]-
)|[^\%]]-)%]"
    )
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'
then
        titleObj = page
    else
        error(string.format(
            "bad argument #1 to 'getTarget'"
            .. " (string, number, or title object
expected, got %s)",
            type(page)
        ), 2)
    end
    if not titleObj or not titleObj.isRedirect then

```

```

        return nil
    end
    -- Find the target by using string matching on the page content.
    local target = p.getTargetFromText(titleObj:getContent() or "")
    if target then
        local targetTitle = getTitle(target)
        if targetTitle then
            if fulltext then
                return targetTitle.fullText
            else
                return targetTitle.prefixedText
            end
        else
            return nil
        end
    else
        -- The page is a redirect, but matching failed. This
indicates a bug in
        -- the redirect matching pattern, so throw an error.
        error(string.format(
            'could not parse redirect on page "%s"',
            fulltext and titleObj.fullText or
titleObj.prefixedText
        ))
    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
        end
    end
end

```

```

        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

```

Retrieved from "<https://www.bluegoldwiki.com/index.php?title=Module:Redirect&oldid=1808>"

Namespaces

- [Module](#)
- [Discussion](#)

Variants

This page was last edited on 19 February 2020, at 09:51.

Blue Gold Program Wiki

The wiki version of the Lessons Learnt Report of the Blue Gold program, documents the experiences of a technical assistance (TA) team working in a development project implemented by the Bangladesh Water Development Board (BWDB) and the Department of Agricultural Extension (DAE) over an eight+ year period from March 2013 to December 2021. The wiki lessons learnt report (LLR) is intended to complement the BWDB and DAE project completion reports (PCRs), with the aim of recording lessons learnt for use in the design and implementation of future interventions in the coastal zone.

- [Privacy policy](#)
- [About Blue Gold Program Wiki](#)
- [Disclaimers](#)

Developed and maintained by Big Blue Communications for Blue Gold Program



[Blue Gold Program Wiki](#)