



Inhaltsverzeichnis

1. Modul:Uses TemplateStyles	2
2. Modul:Uses TemplateStyles/Doku	5
3. Modul:Uses TemplateStyles/config	6



Modul:Uses TemplateStyles

Vorlage:Lua

Implements [Vorlage:Tx](#).

[[Category:Global modules{{#translation:}}]]

```
-- This module implements the {{Uses TemplateStyles}} template.
local yesno = require('Module:Yesno')
local mList = require('Module:List')
local mTableTools = require('Module:TableTools')
local mMessageBox = require('Module:Message box')
local TNT = require('Module:TNT')

local p = {}

local function format(msg, ...)
    return TNT.format('I18n/Uses TemplateStyles', msg, ...)
end

local function getConfig()
    return mw.loadData('Module:Uses TemplateStyles/config')
end

function p.main(frame)
    local origArgs = frame:getParent().args
    local args = {}
    for k, v in pairs(origArgs) do
        v = v:match('^%s*(.)%s*$')
        if v ~= '' then
            args[k] = v
        end
    end
    return p._main(args)
end

function p._main(args, cfg)
    local tStyles = mTableTools.compressSparseArray(args)
    local box = p.renderBox(tStyles)
    local trackingCategories = p.renderTrackingCategories(args, tStyles)
    return box .. trackingCategories
end

function p.renderBox(tStyles)
    local boxArgs = {}
    if #tStyles < 1 then
        boxArgs.text = string.format('<strong class="error">%s</strong>',
    else
        local cfg = getConfig()
        local tStylesLinks = {}
        for i, ts in ipairs(tStyles) do
            local link = string.format('[[:%s]]', ts)
            local sandboxLink = nil
            local tsTitle = mw.title.new(ts)
            if tsTitle and cfg['sandbox_title'] then
                local tsSandboxTitle = mw.title.new(string.format(
                    '%s:%s/%s/%s', tsTitle.nsText, tsTitle.be
                if tsSandboxTitle and tsSandboxTitle.exists then
```




```
                cats[i] = string.format('[[Category:%s]]', cat)
            end
        return table.concat(cats)
    end
    return p
```



Modul:Uses TemplateStyles/Doku

Dies ist die Dokumentationsseite für Modul:Uses TemplateStyles

Vorlage:Lua

Implements Vorlage:TLx.

[[Category:Module documentation pages{{#translation:}}]]



Modul:Uses TemplateStyles/config

Die Dokumentation für dieses Modul kann unter *Modul:Uses TemplateStyles/config/Doku* erstellt werden

```
local cfg = {} -- Don't touch this line.

-- Subpage blacklist: these subpages will not be categorized (except for the
-- error category, which is always added if there is an error).
-- For example "Template:Foo/doc" matches the `doc = true` rule, so it will have
-- no categories. "Template:Foo" and "Template:Foo/documentation" match no rules,
-- so they *will* have categories. All rules should be in the
-- ['<subpage name>'] = true,
-- format.
cfg['subpage_blacklist'] = {
    ['doc'] = true,
    ['sandbox'] = true,
    ['sandbox2'] = true,
    ['testcases'] = true,
}

-- Sandbox title: if the stylesheet's title is <template>/<stylesheet>.css, the
-- stylesheet's sandbox is expected to be at <template>/<sandbox_title>/<stylesheet>.
-- Set to nil to disable sandbox links.
cfg['sandbox_title'] = 'sandbox'

-- Error category: this category is added if the module call contains errors
-- (e.g. no stylesheet listed). A category name without namespace, or nil
-- to disable categorization (not recommended).
cfg['error_category'] = 'Uses TemplateStyles templates with errors'

-- Default category: this category is added if no custom category is specified
-- in module/template call. A category name without namespace, or nil
-- to disable categorization.
cfg['default_category'] = 'Templates using TemplateStyles'

-- Protection conflict category: this category is added if the protection level
-- of any stylesheet differs from the one of the template. A category name
-- without namespace, or nil to disable categorization (not recommended).
cfg['protection_conflict_category'] = 'Templates using TemplateStyles with a dif

-- Padlock pattern: Lua pattern to search on protected stylesheets for, or nil
-- to disable padlock check.
cfg['padlock_pattern'] = '{{pp-'

-- Missing padlock category: this category is added if a protected stylesheet
-- doesn't contain any padlock template (specified by the above Lua pattern).
-- A category name without namespace (no nil allowed) if the pattern is not nil,
-- unused (and thus may be nil) otherwise.
cfg['missing_padlock_category'] = 'Templates using TemplateStyles without padlock

return cfg -- Don't touch this line.
```