

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
```