Wiki Edits (Bookstack)

This page provides a collection of useful code snippets for the Wiki. The Wiki used in question is the open-source 'Bookstack'.

Dark Design Enforcement

It is possible to default to dark magic and only provide a dark theme experience to your users. Needed are two things, the parameter for docker and some CSS:

```
APP_DEFAULT_DARK_MODE=true
```

```
form:has(svg[data-icon="light-mode"], form[action*="toggle-dark-mode"] ) {
    display: none !important;
}

nav > div.dropdown-container > ul > li[role="presentation"] {
    display: none !important;
}
```

Be advised that the theme-selection is cached and any user might still use a light theme if their session is active and the preference is thus being enforced.

Own Theme for CodeMirror

To provide CodeMirror with a own theme there are several things possible.

The docs for this are located at https://www.bookstackapp.com/docs/admin/visual-customisation/#changing-code-block-themes
Below is a simple javascript of the code highlighting used here that you can put into the "Custom HTML Head Content":

```
<script>
const background        = '#182534',  // editor background
      highlightBg       = '#252D3C',  // active line / selection highlight
      surface           = '#244497',  // tooltips / panels
```

```
      mutedText         = '#5C7088',  // dimmed text (comments etc.)
      softText          = '#7C90A0',  // secondary text
      accentText        = '#9FB4C5',  // softer accents
      mainText          = '#F5F7FA',  // main foreground text
      brightText        = '#FFFFFF',  // pure white for emphasis

      redAccent         = '#FF4C4C',  // errors / critical
      orangeAccent      = '#FF9D5C',  // annotations / classes
      yellowAccent      = '#FFD43B',  // constants / tags
      greenAccent       = '#6EEB83',  // emphasis / italic / inserted
      mintString        = '#4ADE80',  // strings
      cyanAccent        = '#2AD4D4',  // functions / properties
      tealAccent        = '#1FA2D6',  // adjusted keyword color (no conflict with outside
blue)
      violetAccent      = '#A86CFF',  // operators / definitions
      magentaAccent     = '#E44897';  // brighter magenta for numbers

const invalid           = redAccent,
      cursor            = accentText;

function viewThemeBuilder() {
    return {
      '&':{color:mainText,backgroundColor:background},
      '.cm-content':{caretColor:cursor},
      '.cm-cursor, .cm-dropCursor':{borderLeftColor:cursor},
      '&.cm-focused .cm-selectionBackground, .cm-selectionBackground, .cm-content
::selection':{backgroundColor:greenAccent},
      '.cm-panels':{backgroundColor:background,color:softText},
      '.cm-panels.cm-panels-top':{borderBottom:'2px solid black'},
      '.cm-panels.cm-panels-bottom':{borderTop:'2px solid black'},
      '.cm-searchMatch':{backgroundColor:'#72a1ff59',outline:'1px solid #457dff'},
      '.cm-searchMatch.cm-searchMatch-selected':{backgroundColor:'#6199ff2f'},
      '.cm-activeLine':{backgroundColor:highlightBg},
      '.cm-selectionMatch':{backgroundColor:'#aafe661a'},
      '&.cm-focused .cm-matchingBracket, &.cm-focused .cm-nonmatchingBracket':{outline:`1px
solid ${tealAccent}`},
      '.cm-gutters':{backgroundColor:background,color:accentText,border:'none'},
      '.cm-activeLineGutter':{backgroundColor:highlightBg},
      '.cm-foldPlaceholder':{backgroundColor:'transparent',border:'none',color:'#ddd'},
      '.cm-tooltip':{border:'none',backgroundColor:surface},
      '.cm-tooltip .cm-tooltip-
```

```
arrow:before':{borderTopColor:'transparent',borderBottomColor:'transparent'},
      '.cm-tooltip .cm-tooltip-
arrow:after':{borderTopColor:surface,borderBottomColor:surface},
      '.cm-tooltip-autocomplete':{
        '& > ul > li[aria-selected]':{backgroundColor:highlightBg,color:softText}
      }
    };
}

function highlightStyleBuilder(t) {
    return [
      {tag:t.keyword,                              color:tealAccent},      // keywords = teal
      {tag:[t.name,t.propertyName,t.macroName],   color:cyanAccent},      // properties /
macros = cyan
      {tag:[t.variableName],                       color:mainText},        // plain variables =
main text
      {tag:[t.function(t.variableName)],          color:violetAccent},    // function names =
violet
      {tag:[t.labelName],                          color:magentaAccent},   // labels = magenta
      {tag:[t.color,t.constant(t.name)],          color:yellowAccent},    // constants =
yellow
      {tag:[t.definition(t.name),t.separator],    color:orangeAccent},    // definitions &
separators = orange
      {tag:[t.brace],                              color:cyanAccent},      // braces = cyan
      {tag:[t.annotation],                         color:orangeAccent},    // annotations =
orange
      {tag:[t.number,t.changed,t.modifier,t.namespace], color:magentaAccent}, // numbers =
magenta (rule)
      {tag:[t.typeName,t.className],              color:orangeAccent},    // classes / types =
orange
      {tag:[t.operator,t.operatorKeyword],        color:violetAccent},    // operators =
violet
      {tag:[t.tagName],                            color:yellowAccent},    // HTML/XML tags =
yellow
      {tag:[t.squareBracket],                      color:redAccent},       // square brackets =
red
      {tag:[t.angleBracket],                       color:softText},        // angle brackets =
muted
      {tag:[t.attributeName],                      color:cyanAccent},      // attribute names =
cyan
      {tag:[t.regexp],                             color:redAccent},       // regex = red
```

```
      {tag:[t.quote],                             color:greenAccent},     // quotes = green
      {tag:[t.string],                            color:mintString},      // strings = mint-
green
      {tag:t.link,
color:cyanAccent,textDecoration:'underline',textUnderlinePosition:'under'},
      {tag:[t.url,t.escape,t.special(t.string)],  color:yellowAccent},
      {tag:[t.meta],                              color:redAccent},
      {tag:[t.comment],                           color:mutedText,fontStyle:'italic'},
      {tag:t.strong,                              fontWeight:'bold',color:brightText},
      {tag:t.emphasis,                            fontStyle:'italic',color:greenAccent},
      {tag:t.strikethrough,                       textDecoration:'line-through'},
      {tag:t.heading,                             fontWeight:'bold',color:yellowAccent},
      {tag:t.heading1,                            fontWeight:'bold',color:brightText},
      {tag:[t.heading2,t.heading3,t.heading4],    fontWeight:'bold',color:tealAccent},
      {tag:[t.heading5,t.heading6],               color:violetAccent},
      {tag:[t.atom,t.bool,t.special(t.variableName)], color:magentaAccent}, // atoms = magenta
      {tag:[t.processingInstruction,t.inserted,t.contentSeparator], color:greenAccent},
      {tag:[t.contentSeparator],                  color:yellowAccent},
      {tag:t.invalid,                             color:softText,borderBottom:`1px dotted
${redAccent}`}
    ];
}

window.addEventListener('library-cm6::configure-theme', event => {
    const detail = event.detail;
    detail.registerViewTheme(viewThemeBuilder);
    detail.registerHighlightStyle(highlightStyleBuilder);
});
</script>
```

Revision #5
Created 2025-08-24 05:10:15 UTC by Alexander (nethavn)
Updated 2025-08-24 05:31:51 UTC by Alexander (nethavn)