Public Documents

This directory contains markdown documents that are automatically rendered to HTML and PDF formats for public access. These documents are typically legal or informational documents such as privacy policies, license terms, and terms of service.

Document Types

The following types of documents are typically transformed and offered publicly:

- **Privacy Policies** (`Datenschutzerklaerung`) - Data protection and privacy information
- **License Terms** (`Lizenzbedingungen`) - Software licensing terms and conditions
- **Terms of Service** - Service usage terms and conditions
- **Legal Notices** - Legal disclaimers and notices
- **User Agreements** - End-user agreements and contracts

How documents are maintained

Within google docs, an overview is managed and contains links to:

- Datenschutzererklärungen
- AAV
- AGB
- Eulas
- Lizenzbedingungen
- Auditbericht Datenschutz

If any of these documents is updated, it should be exported as markdown and updated here to. If a name of an document hereby did change, the affected jumplink must also be updated.

How It Works

Documents in this directory are processed by the `renderMarkdown` script, which:

1. **Converts Markdown to HTML and PDF** - Each `.md` file is rendered to both HTML and PDF formats
2. **URL-Compatible Naming** - File and directory names are automatically converted to be web-compatible:
    - Umlauts (ä, ö, ü, Ä, Ö, Ü) are replaced with URL-safe equivalents (ae, oe, ue, Ae, Oe, Ue)
    - Spaces are replaced with underscores
    - Special characters are normalized
3. **Index Generation** - An `index.html` file is automatically generated with links to all rendered documents
4. **Output Location** - Rendered files are placed in `bin/public/documents/` for deployment

Building Documents

To build all documents, run:

```
npm run build:documentation
```

This command:

- Processes all `.md` files in `public/documents/`
- Renders them to both HTML and PDF formats
- Generates an `index.html` with links to all documents
- Outputs everything to `bin/public/documents/`

To clean the output directory:

```
npm run clean:documentation
```

Customization Options

The rendering script supports various customization options through command-line arguments or by modifying the script itself.

Command-Line Options

Output Format

Control which formats are generated:

```
# Generate both HTML and PDF (default)
node bin/scripts/renderMarkdown.js --input=public/documents --format=both

# Generate HTML only
node bin/scripts/renderMarkdown.js --input=public/documents --format=html

# Generate PDF only
node bin/scripts/renderMarkdown.js --input=public/documents --format=pdf
```

Custom CSS Styling

Provide a custom CSS file to override the default styling:

```
node bin/scripts/renderMarkdown.js --input=public/documents --cssPath=./custo
```

The custom CSS will be applied to both HTML and PDF outputs. If not provided, a default stylesheet is used that includes:

- Responsive design with max-width of 900px
- System font stack for optimal cross-platform rendering
- Dark mode support via `color-scheme: light dark`
- Styled code blocks, tables, blockquotes, and images
- Print-friendly PDF margins (20mm top/bottom, 15mm left/right)

Index File Customization

Control index generation and naming:

```
# Generate index with custom name
node bin/scripts/renderMarkdown.js --input=public/documents --generateIndex=t
```

```
# Skip index generation
node bin/scripts/renderMarkdown.js --input=public/documents --generateIndex=fa
```

Output Directory

Specify a custom output directory:

```
node bin/scripts/renderMarkdown.js --input=public/documents --outputDir=./outp
```

Markdown Features Supported

The renderer uses MarkdownIt with the following features enabled:

- **HTML Support** - Raw HTML can be embedded in markdown files
- **Auto-linking** - URLs are automatically converted to links
- **Typography** - Smart typography replacements (quotes, dashes, etc.)

Supported Markdown Syntax

- **Headers** - `# H1`, `## H2`, `### H3`, etc.
- **Emphasis** - `*italic*`, `**bold**`, `***bold italic***`
- **Lists** - Ordered (`1.`) and unordered (`-`, `*`)
- **Links** - `[text](url)` or automatic URL detection
- **Images** - `![alt](url)` with automatic responsive sizing
- **Code** - Inline `` `code` `` and code blocks with syntax highlighting
- **Tables** - Standard markdown table syntax
- **Blockquotes** - `> quoted text`
- **Horizontal Rules** - `---` or `***`
- **HTML** - Raw HTML can be used for advanced formatting

Customizing Default Styles

To customize the default CSS, you can:

1. **Create a custom CSS file** and use the `--cssPath` option
2. **Modify the script** - Edit `src/scripts/renderMarkdown.ts` and update the `defaultCss` constant (around line 115)

The default CSS includes:

- Responsive layout (max-width: 900px, centered)
- System font stack for cross-platform compatibility
- Dark mode support
- Styled code blocks with borders and padding
- Table styling with borders
- Blockquote styling with left border
- Responsive images and media

PDF Customization

PDF generation uses Playwright's Chromium browser. The PDF settings can be customized in the `writePdfFromHtml` function:

- **Page Format** - Currently set to A4 (can be changed to Letter, Legal, etc.)
- **Margins** - Currently 20mm top/bottom, 15mm left/right
- **Background** - Print background graphics enabled

To modify PDF settings, edit `src/scripts/renderMarkdown.ts` around line 196.

File Naming Conventions

When adding new documents:

1. **Use descriptive names** - Clear, descriptive filenames help identify document purpose
2. **Use underscores for spaces** - While the script converts spaces automatically, using underscores in source files is recommended
3. **Avoid special characters** - Stick to letters, numbers, underscores, and hyphens
4. **Use `.md` extension** - Only `.md` files are processed

Example good filenames:

- `Datenschutzerklaerung_fuer_die_App_pcvisit_Connect_.md`
- `Lizenzbedingungen_fuer_die_App_pcvisit_Connect_.md`
- `Terms_of_Service.md`

Adding New Documents

To add a new public document:

1. Create a new `.md` file in `public/documents/`
2. Write your content using markdown syntax
3. Run `npm run build:documentation` to generate HTML and PDF versions
4. The document will automatically appear in the generated `index.html`

Technical Details

- **Rendering Engine**: MarkdownIt (v14+)
- **PDF Generation**: Playwright Chromium
- **Output Formats**: HTML5 and PDF (A4 format)
- **Character Encoding**: UTF-8
- **Language**: Documents are rendered with `lang="en"` attribute (can be customized in script)

Troubleshooting

Documents Not Rendering

- Ensure files have `.md` extension
- Check that markdown syntax is valid
- Verify file encoding is UTF-8

PDF Generation Issues

- Ensure Playwright browsers are installed: `npx playwright install`
- Check that Chromium is available in the build environment
- Verify sufficient disk space for PDF output

Styling Issues

- Check CSS syntax if using custom CSS
- Verify CSS file path is correct relative to script execution
- Review browser console for CSS errors in HTML output

Script Location

The rendering script is located at:

- **Source**: `src/scripts/renderMarkdown.ts`
- **Compiled**: `bin/scripts/renderMarkdown.js`

For more information about the script’s command-line interface, run:

```
node bin/scripts/renderMarkdown.js --help
```