
lawwenda - Plugin API Documentation

Release 0.1.64

Josef Hahn

Mar 16, 2021

CONTENTS

1	API reference	3
1.1	lawwenda package	3
	Python Module Index	29
	Index	31

Lawwenda has a plugin interface which allows to add functionality to the core from outside.

TODO not yet available

API REFERENCE

1.1 lawwenda package

1.1.1 Subpackages

lawwenda._aux package

Submodules

lawwenda._aux.projectinformations module

Module contents

lawwenda.preview package

Submodules

lawwenda.preview.commonimages module

```
class lawwenda.preview.commonimages.CommonImagesPreviewer
    Bases: lawwenda.preview.Previewer
    _abc_impl = <_abc_data object>
    is_previewable(iar)
    is_thumbnailable(iar)
    preview_html(node)
    thumbnail(node)
```

lawwenda.preview.commonvideos module

class lawwenda.preview.commonvideos.CommonVideosPreviewer

Bases: *lawwenda.preview.Previewer*

__ffmpeg_thumb (*inputpath*, *stdin=-3*)

Parameters

- **inputpath** (*str*) –
- **stdin** (*Any*) –

Return type bytes

_abc_impl = *<_abc_data object>*

is_previewable (*iar*)

is_thumbnailable (*iar*)

preview_html (*node*)

thumbnail (*node*)

Module contents

class lawwenda.preview.IsAbleRequest (*name: str, mimetype: str*)

Bases: object

Parameters

- **name** (*str*) –
- **mimetype** (*str*) –

Return type None

mimetype: *str*

name: *str*

class lawwenda.preview.Previewer

Bases: *abc.ABC*

_abc_impl = *<_abc_data object>*

_image_scale_to_thumbnail_size (*imgdata*)

Parameters *imgdata* (*BinaryIO*) –

Return type *_io.BytesIO*

abstract is_previewable (*iar*)

Parameters *iar* (*lawwenda.preview.IsAbleRequest*) –

Return type bool

abstract is_thumbnailable (*iar*)

Parameters *iar* (*lawwenda.preview.IsAbleRequest*) –

Return type bool

abstract preview_html (*node*)

Parameters `node` (`lawwenda.fs.Filesystem.Node`) –

Return type `str`

`thumbheight = 300`

`abstract thumbnail` (`node`)

Parameters `node` (`lawwenda.fs.Filesystem.Node`) –

Return type `bytes`

`thumbsize = (300, 300)`

`thumbwidth = 300`

lawwenda.search package

Module contents

class `lawwenda.search.ByGeoSearch` (`position`, `radius`)

Bases: `lawwenda.search.Search`

Parameters

- `position` (`str`) –
- `radius` (`str`) –

`__decide` (`node`)

Parameters `node` (`FilesystemNode`) –

Return type `bool`

`query` (`node`)

class `lawwenda.search.ByNameSearch` (`term`)

Bases: `lawwenda.search.Search`

Parameters `term` (`str`) –

`__decide` (`node`)

Parameters `node` (`FilesystemNode`) –

Return type `bool`

`query` (`node`)

class `lawwenda.search.ByTagSearch` (`term`)

Bases: `lawwenda.search.Search`

Parameters `term` (`str`) –

`__decide` (`node`)

Parameters `node` (`FilesystemNode`) –

Return type `bool`

`query` (`node`)

class `lawwenda.search.DeeplySearch` (`term`)

Bases: `lawwenda.search.Search`

Parameters `term` (`str`) –

__decide_unindexed (*node*)

Parameters **node** (*FileSystemNode*) –

Return type `bool`

__query_indexed (*node*, *idxdata*)

Parameters

- **node** (*FileSystemNode*) –
- **idxdata** (*object*) –

Return type `FileSystemNodes`

__query_unindexed (*node*)

Parameters **node** (*FileSystemNode*) –

Return type `FileSystemNodes`

__try_get_index (*node*)

Parameters **node** (*FileSystemNode*) –

Return type `object`

query (*node*)

class `lawwenda.search.Search`

Bases: `object`

query (*node*)

Parameters **node** (*FileSystemNode*) –

Return type `FileSystemNodes`

`lawwenda.search.__query_by_tree_traversal` (*node*, *decidefct*)

Parameters

- **node** (*FileSystemNode*) –
- **decidefct** (*Callable*[[*FileSystemNode*], *bool*]) –

Return type `FileSystemNodes`

`lawwenda.search.__termstring_to_termlist` (*term*)

Parameters **term** (*str*) –

Return type `List[str]`

`lawwenda.search.create_search` (*mode*, ***kwargs*)

Parameters **mode** (*str*) –

Return type `lawwenda.search.Search`

lawwenda.sharehttpapp package

Submodules

lawwenda.sharehttpapp.davprop module

```
class lawwenda.sharehttpapp.davprop.CommentDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)

class lawwenda.sharehttpapp.davprop.DavProp (davname, *, is_writable)
    Bases: object

    Parameters

        • davname (str) –

        • is_writable (bool) –

    property davname

        Return type str

        Return type str

        Return type str

    get_for_node (fsnode)

        Parameters fsnode (lawwenda.fs.Filesystem.Node) –

        Return type Optional[str]

    property is_writable

        Return type bool

        Return type bool

        Return type bool

class lawwenda.sharehttpapp.davprop.GeoDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)

class lawwenda.sharehttpapp.davprop.MtimeDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)

class lawwenda.sharehttpapp.davprop.RatingDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)

class lawwenda.sharehttpapp.davprop.ResourceTypeDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)

class lawwenda.sharehttpapp.davprop.SizeDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)
```

```
class lawwenda.sharehttpapp.davprop.TagsDavProp
    Bases: lawwenda.sharehttpapp.davprop.DavProp

    get_for_node (fsnode)

lawwenda.sharehttpapp.davprop.get_prop_by_davname (davname)

    Parameters davname (str) –

    Return type Optional[lawwenda.sharehttpapp.davprop.DavProp]

lawwenda.sharehttpapp.davprop.register_prop (prop)

    Parameters prop (Type [DavProp]) –
```

Module contents

```
class lawwenda.sharehttpapp.JsonSerializedResponse (data, **kwargs)
    Bases: werkzeug.wrappers.response.Response

    Parameters data (Optional [Any]) –

class lawwenda.sharehttpapp.Request (environ, populate_request=True, shallow=False)
    Bases: werkzeug.wrappers.request.Request, werkzeug.wrappers.json.JSONMixin

class lawwenda.sharehttpapp.ShareHttpApp (filesystem)
    Bases: object

    Parameters filesystem (lawwenda.fs.Filesystem) –

    _method_copy (request, fsnode, *, xfermethod='copy_to')

        Parameters

        • request (lawwenda.sharehttpapp.Request) –

        • fsnode (lawwenda.fs.Filesystem.Node) –

        • xfermethod (str) –

    _method_delete (request, fsnode)

        Parameters

        • request (lawwenda.sharehttpapp.Request) –

        • fsnode (lawwenda.fs.Filesystem.Node) –

    _method_get (request, fsnode, is_head_request=False)

        Parameters

        • request (lawwenda.sharehttpapp.Request) –

        • fsnode (lawwenda.fs.Filesystem.Node) –

        • is_head_request (bool) –

    _method_head (request, fsnode)

        Parameters

        • request (lawwenda.sharehttpapp.Request) –

        • fsnode (lawwenda.fs.Filesystem.Node) –

    _method_mkcol (request, fsnode)
```

Parameters

- **request** (`lawwenda.sharehttpapp.Request`) –
- **fsnode** (`lawwenda.fs.Filesystem.Node`) –

`_method_move` (*request, fsnode*)

Parameters

- **request** (`lawwenda.sharehttpapp.Request`) –
- **fsnode** (`lawwenda.fs.Filesystem.Node`) –

`_method_options` (*request, fsnode*)

Parameters

- **request** (`lawwenda.sharehttpapp.Request`) –
- **fsnode** (`lawwenda.fs.Filesystem.Node`) –

`_method_propfind` (*request, fsnode*)

Parameters

- **request** (`lawwenda.sharehttpapp.Request`) –
- **fsnode** (`lawwenda.fs.Filesystem.Node`) –

`_method_proppatch` (*request, fsnode*)

Parameters

- **request** (`lawwenda.sharehttpapp.Request`) –
- **fsnode** (`lawwenda.fs.Filesystem.Node`) –

`_method_put` (*request, fsnode*)

Parameters

- **request** (`lawwenda.sharehttpapp.Request`) –
- **fsnode** (`lawwenda.fs.Filesystem.Node`) –

`property allowed_methods`

Return type `t.List[str]`

Return type `t.List[str]`

Return type `t.List[str]`

1.1.2 Submodules

1.1.3 lawwenda.datafiles module

`lawwenda.datafiles.find_data_file` (*fname, searchdirs=None*)

Parameters

- **fname** (*str*) –
- **searchdirs** (*Optional[Iterable[str]]*) –

Return type `Optional[str]`

1.1.4 lawwenda.fmapp module

class lawwenda.fmapp.FmApp(*share*)

Bases: object

Parameters **share** (lawwenda.Share) –

__auth (*username, password*)

Parameters

- **username** (*str*) –

- **password** (*str*) –

Return type bool

__dispatch_request (*request*)

__ensure_authed (*request*)

__on_api_copymove (*request, action*)

__render_template (*template, *, html_head_inner="", path="", headonly=False, url_internals_name='~__lawwenda__int~', **kwargs*)

Parameters

- **template** (*str*) –

- **html_head_inner** (*str*) –

- **path** (*str*) –

- **headonly** (*bool*) –

- **url_internals_name** (*str*) –

Return type werkzeug.wrappers.response.Response

__render_template_text (*template, *, html_head_inner, **kwargs*)

Parameters

- **template** (*str*) –

- **html_head_inner** (*str*) –

Return type str

__render_template_text_raw (*template, **kwargs*)

Parameters **template** (*str*) –

Return type str

_on_api_add_tag_to_entries (*request*)

_on_api_copy (*request*)

_on_api_delete (*request*)

_on_api_details (*request*)

_on_api_dir (*request*)

_on_api_known_tags (*_*)

_on_api_mkdir (*request*)

_on_api_move (*request*)

```

    _on_api_remove_tag_from_entries (request)
    _on_api_rename (request)
    _on_api_set_comment (request)
    _on_api_set_geo (request)
    _on_api_set_rating (request)
    _on_api_thumbnail (request)
    _on_api_upload (request)
    _on_api_zip (request)
    _on_api_zip_download (request, zipid)
    _on_help (_)
    _on_static (_, filepath)
    url_internals_name = '~__lawwenda__int~'

class lawwenda.fmapp.JsonSerializedResponse (data, **kwargs)
    Bases: werkzeug.wrappers.response.Response

    Parameters data (Optional[Any]) –

class lawwenda.fmapp.Request (environ, populate_request=True, shallow=False)
    Bases: werkzeug.wrappers.request.Request, werkzeug.wrappers.json.JSONMixin

class lawwenda.fmapp._RenderTemplateValue (s)
    Bases: object

    property unescaped

class lawwenda.fmapp._TempZips
    Bases: object

    class _TempZip (nodes)
        Bases: object

        Parameters nodes (List[lawwenda.fs.Filesystem.Node]) –

        _TempZip__is_executable (node)
            Parameters node (lawwenda.fs.Filesystem.Node) –
            Return type bool

        _TempZip__putnode (node, zipf)
            Parameters
            • node (lawwenda.fs.Filesystem.Node) –
            • zipf (zipfile.ZipFile) –
            Return type None

        property bytes

    classmethod _TempZips__cleanup_loop ()
    _cleanup_thread = None
    _lock = <unlocked _thread.lock object>
    _zips = {}

    classmethod create_tempzip (owner, nodes)
        Parameters

```

- **owner** (*object*) –
- **nodes** (*List[lawwenda.fs.Filesystem.Node]*) –

Return type *str*

classmethod **get_tempzip** (*owner, zipid*)

Parameters

- **owner** (*object*) –
- **zipid** (*str*) –

Return type *Optional[bytes]*

1.1.5 lawwenda.fs module

class **lawwenda.fs.AbstractFilesystemDecorator** (*inner*)
Bases: *lawwenda.fs.Filesystem*

Parameters **inner** (*lawwenda.fs.Filesystem*) –

_eval_predicate (*pred, node*)

exception **lawwenda.fs.CircularTraversalError**
Bases: *OSError*

class **lawwenda.fs.ExcludeNodesFilesystemAdapter** (*inner, pred*)
Bases: *lawwenda.fs.AbstractFilesystemDecorator*

Parameters

- **inner** (*lawwenda.fs.Filesystem*) –
- **pred** (*Callable[[Filesystem.Node], bool]*) –

child_nodes (*handle*)

Parameters **handle** (*lawwenda.fs.Filesystem.ReadHandle*) –

Return type *List[lawwenda.fs.Filesystem.Node]*

get_readhandle (*node*)

get_writehandle (*node*)

class **lawwenda.fs.Filesystem**
Bases: *object*

class **Node** (*path, *, filesystem*)
Bases: *object*

Parameters

- **path** (*str*) –
- **filesystem** (*Filesystem*) –

property **_filesystem**

Return type *Filesystem*

Return type *Filesystem*

Return type *Filesystem*

add_tag (*tag*)

Parameters **tag** (*str*) –

Return type None

property basics_as_dict

child_by_name (*name*)
Parameters **name** (*str*) –
Return type *lawwenda.fs.Filesystem.Node*

property child_nodes
Return type t.List['Filesystem.Node']
Return type t.List['Filesystem.Node']
Return type t.List['Filesystem.Node']

property comment
Return type str
Return type str
Return type str

copy_to (*newpath*)
Parameters **newpath** (*str*) –
Return type None

delete ()
Return type None

property dirpath
Return type str
Return type str
Return type str

property exists
Return type bool
Return type bool
Return type bool

property full_as_dict

property geo
Return type str
Return type str
Return type str

property geo_obj
Return type pimetadatainterpreter.GeoLocation
Return type pimetadatainterpreter.GeoLocation
Return type pimetadatainterpreter.GeoLocation

property has_preview
Return type bool
Return type bool
Return type bool

property has_thumbnail
Return type bool
Return type bool
Return type bool

property icon_name
Return type t.Optional[str]
Return type t.Optional[str]
Return type t.Optional[str]

property is_dir
Return type bool
Return type bool
Return type bool

property is_file
Return type bool
Return type bool
Return type bool

property is_hidden
Return type bool
Return type bool
Return type bool

property is_link
Return type bool
Return type bool
Return type bool

property is_writable
Return type bool
Return type bool
Return type bool

property mimetype
Return type str
Return type str
Return type str

mkdir()
Return type None

move_to(newpath)
Parameters **newpath**(str) –
Return type None

property mtime
Return type datetime.datetime
Return type datetime.datetime
Return type datetime.datetime

property mtime_ts
Return type float
Return type float
Return type float

property name
Return type str
Return type str
Return type str

property parent_node
Return type *Filesystem.Node*
Return type *Filesystem.Node*
Return type *Filesystem.Node*

property path
Return type str
Return type str

Return type str

property preview_html
Return type str
Return type str
Return type str

property rating
Return type int
Return type int
Return type int

read_file()
Return type BinaryIO

remove_tag(tag)
Parameters tag (str) –
Return type None

set_comment(comment)
Parameters comment (str) –
Return type None

set_geo(geo)
Parameters geo (str) –
Return type None

set_rating(rating)
Parameters rating (int) –
Return type None

property size
Return type int
Return type int
Return type int

property tags
Return type t.List[str]
Return type t.List[str]
Return type t.List[str]

property tagstring
Return type str
Return type str
Return type str

thumbnail()
Return type bytes

traverse_dir(*, raise_on_circle, param_path=)
Parameters
• **raise_on_circle** (bool) –
• **param_path** (str) –
Return type Iterable[Tuple[[lawwenda.fs.Filesystem.Node](#), str]]

try_get_fullpath(*, writable)
Parameters writable (bool) –
Return type Optional[str]

write_file(content)
Parameters content (Union[bytes, BinaryIO]) –

Return type None

class `ReadHandle` (*node*)

Bases: `object`

Read and write handles are just a stupid container that hold one `Filesystem.Node`.

This looks stupid at first, because you could just use this node directly instead. The added value of handles are a central mechanism for access control, which would be a bit less obvious and more scattered in code without this indirection.

Of course it cannot avoid a way around it in code. An attacker that can change the code has won anyway. It just simplifies writing correct code that hopefully does not provide ways around it for the client.

See `Filesystem.get_readhandle()` and `Filesystem.get_writehandle()`.

Parameters `node` (`Filesystem.Node`) –

class `WriteHandle` (*node*)

Bases: `lawwenda.fs.Filesystem.ReadHandle`

See `Filesystem.ReadHandle`.

Parameters `node` (`Filesystem.Node`) –

add_tag (*handle*, *tag*)

Parameters

- **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **tag** (*str*) –

Return type None

child_nodes (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type List[`lawwenda.fs.Filesystem.Node`]

comment (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type str

copy_to (*srchandle*, *desthandle*)

Parameters

- **srchandle** (`lawwenda.fs.Filesystem.ReadHandle`) –
- **desthandle** (`lawwenda.fs.Filesystem.WriteHandle`) –

Return type None

delete (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –

Return type None

exists (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type bool

geo (*handle*)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `lawwenda._aux.PiMetadataInterpreter.pimetadainterpreter.GeoLocation`

get_readhandle (`node`)

Parameters `node` (`lawwenda.fs.Filesystem.Node`) –

Return type `lawwenda.fs.Filesystem.ReadHandle`

get_writehandle (`node`)

Parameters `node` (`lawwenda.fs.Filesystem.Node`) –

Return type `lawwenda.fs.Filesystem.WriteHandle`

has_preview (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `bool`

has_thumbnail (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `bool`

is_dir (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `bool`

is_file (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `bool`

is_hidden (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `bool`

is_link (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `bool`

known_tags ()

Return type `List[str]`

mimetype (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `str`

mkdir (`handle`)

Parameters `handle` (`lawwenda.fs.Filesystem.WriteHandle`) –

Return type `None`

move_to (`srchandle`, `desthandle`)

Parameters

- **srchandle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **desthandle** (`lawwenda.fs.Filesystem.WriteHandle`) –

Return type `None`

mtime (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `datetime.datetime`

node_by_path (*path*)

Parameters **path** (*str*) –

Return type `lawwenda.fs.Filesystem.Node`

preview_html (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `str`

rating (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `int`

read_file (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type `BinaryIO`

remove_tag (*handle, tag*)

Parameters

- **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **tag** (*str*) –

Return type `None`

property rootnode

Return type `Filesystem.Node`

Return type `Filesystem.Node`

Return type `Filesystem.Node`

static sanitize_abspath (*path*)

Sanitizes slashes in a path and returns a path in the form */foo/bar*. Precisely:

- with a slash in the beginning
- without a slash at the end
- without double slashes
- with `..` and `.` resolved
- root path: `/`

Parameters **path** (*str*) – The input path.

Return type `str`

set_comment (*handle*, *comment*)

Parameters

- **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **comment** (*str*) –

Return type None

set_geo (*handle*, *geo*)

Parameters

- **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **geo** (*str*) –

Return type None

set_rating (*handle*, *rating*)

Parameters

- **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **rating** (*int*) –

Return type None

size (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type int

tags (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type List[str]

thumbnail (*handle*)

Parameters **handle** (`lawwenda.fs.Filesystem.ReadHandle`) –

Return type bytes

try_get_fullpath (*handle*, *, *writable*)

Parameters

- **handle** (`Union[Filesystem.ReadHandle, Filesystem.WriteHandle]`) –
- **writable** (*bool*) –

Return type Optional[str]

write_file (*handle*, *content*)

Parameters

- **handle** (`lawwenda.fs.Filesystem.WriteHandle`) –
- **content** (`Union[bytes, BinaryIO]`) –

Return type None

class `lawwenda.fs.HideNodesFilesystemAdapter` (*inner*, *pred*)

Bases: `lawwenda.fs.AbstractFilesystemDecorator`

Parameters

- **inner** (`lawwenda.fs.Filesystem`) –
- **pred** (`Callable[[Filesystem.Node], bool]`) –

is_hidden (*handle*)

class `lawwenda.fs.LocalFilesystem` (*rootpath*)

Bases: `lawwenda.fs.Filesystem`

Parameters **rootpath** (*str*) –

__path_to_fullpath (*path*)

Parameters **path** (*str*) –

Return type `str`

add_tag (*handle*, *tag*)

child_nodes (*handle*)

comment (*handle*)

copy_to (*srchandle*, *desthandle*)

delete (*handle*)

exists (*handle*)

geo (*handle*)

get_readhandle (*node*)

get_writehandle (*node*)

has_preview (*handle*)

has_thumbnail (*handle*)

is_dir (*handle*)

is_file (*handle*)

is_hidden (*handle*)

is_link (*handle*)

known_tags ()

mimetype (*handle*)

mkdir (*handle*)

move_to (*srchandle*, *desthandle*)

mtime (*handle*)

preview_html (*handle*)

rating (*handle*)

read_file (*handle*)

remove_tag (*handle*, *tag*)

set_comment (*handle*, *comment*)

set_geo (*handle*, *geo*)


```

    set_rating (handle, rating)
    size (handle)
    tags (handle)
    thumbnail (handle)
    try_get_fullpath (handle, *, writable)
    write_file (handle, content)

class lawwenda.fs.ReadOnlyFilesystemAdapter (inner)
    Bases: lawwenda.fs.AbstractFilesystemDecorator

    Parameters inner (lawwenda.fs.Filesystem) –
    get_writehandle (node)

class lawwenda.fs._PredicateFactory
    Bases: object

    static false (node)
    static p_descending (pred)
    static p_not (pred)
    static p_or (*preds)
    static p_regexp (restr)
    static p_tag (tag)

lawwenda.fs.create_filesystem (rootpath, *, readonly, hide_by_patterns=(), hide_by_tags=(),
                                include_by_patterns=None, include_by_tags=None,
                                exclude_by_patterns=(), exclude_by_tags=(), ex-
                                clude_hidden=False)

```

Parameters

- **rootpath** (*str*) –
- **readonly** (*bool*) –
- **hide_by_patterns** (*Iterable[str]*) –
- **hide_by_tags** (*Iterable[str]*) –
- **include_by_patterns** (*Optional[Iterable[str]]*) –
- **include_by_tags** (*Optional[Iterable[str]]*) –
- **exclude_by_patterns** (*Iterable[str]*) –
- **exclude_by_tags** (*Iterable[str]*) –
- **exclude_hidden** (*bool*) –

Return type *lawwenda.fs.Filesystem*

1.1.6 lawwenda.lawwenda_cli module

`lawwenda.lawwenda_cli._fmt_cmd(s)`

Parameters `s (str)` –

Return type `str`

`lawwenda.lawwenda_cli._quickstart(cfgpath)`

Parameters `cfgpath (str)` –

Return type `str`

`lawwenda.lawwenda_cli.add_share(*, cfg, name, path, title, active_until, hide_by_pattern, hide_by_tag, include_by_pattern, include_by_tag, exclude_by_pattern, exclude_by_tag, exclude_hidden, readonly)`

Parameters

- `cfg (lawwenda.Configuration)` –
- `name (str)` –
- `path (str)` –
- `title (str)` –
- `active_until (str)` –
- `hide_by_pattern (List[str])` –
- `hide_by_tag (List[str])` –
- `include_by_pattern (List[str])` –
- `include_by_tag (List[str])` –
- `exclude_by_pattern (List[str])` –
- `exclude_by_tag (List[str])` –
- `exclude_hidden (bool)` –
- `readonly (bool)` –

Return type `None`

`lawwenda.lawwenda_cli.console(*, cfg)`

Parameters `cfg (lawwenda.Configuration)` –

Return type `None`

`lawwenda.lawwenda_cli.describe_share(*, cfg, name)`

Parameters

- `cfg (lawwenda.Configuration)` –
- `name (str)` –

Return type `str`

`lawwenda.lawwenda_cli.generate_wsgi(*, cfg)`

Parameters `cfg (lawwenda.Configuration)` –

Return type `str`

`lawwenda.lawwenda_cli.list_shares(*, cfg)`

Parameters `cfg` (`lawwenda.Configuration`) –

Return type `str`

`lawwenda.lawwenda_cli.main()`

`lawwenda.lawwenda_cli.remove_share(*, cfg, name)`

Parameters

- `cfg` (`lawwenda.Configuration`) –
- `name` (`str`) –

Return type `None`

`lawwenda.lawwenda_cli.run_dev_server(*, cfg)`

Parameters `cfg` (`lawwenda.Configuration`) –

Return type `None`

1.1.7 lawwenda.server module

class `lawwenda.server.Server(*, cfgpath=None)`

Bases: `object`

Parameters `cfgpath` (`Optional[str]`) –

class `_ShareAppInst(share)`

Bases: `object`

Parameters `share` (`lawwenda.Share`) –

property `app`

Return type `lawwenda.fmapp.FmApp`

Return type `lawwenda.fmapp.FmApp`

Return type `lawwenda.fmapp.FmApp`

property `is_active`

Return type `bool`

Return type `bool`

Return type `bool`

`__get_app_for_share(sharename)`

Parameters `sharename` (`str`) –

1.1.8 Module contents

class `lawwenda.Configuration(cfgpath=None)`

Bases: `object`

Parameters `cfgpath` (`Optional[str]`) –

`__verify_valid_name()`

Parameters `name` (`str`) –

Return type `None`

add_share (*path*, *, *name*, *password*, *title=None*, *readonly=True*, *hide_by_patterns=()*, *hide_by_tags=()*, *include_by_patterns=None*, *include_by_tags=None*, *exclude_by_patterns=()*, *exclude_by_tags=()*, *exclude_hidden=False*, *active_until=None*)

Parameters

- **path** (*str*) –
- **name** (*str*) –
- **password** (*Optional[str]*) –
- **title** (*Optional[str]*) –
- **readonly** (*bool*) –
- **hide_by_patterns** (*Iterable[str]*) –
- **hide_by_tags** (*Iterable[str]*) –
- **include_by_patterns** (*Optional[Iterable[str]]*) –
- **include_by_tags** (*Optional[Iterable[str]]*) –
- **exclude_by_patterns** (*Iterable[str]*) –
- **exclude_by_tags** (*Iterable[str]*) –
- **exclude_hidden** (*bool*) –
- **active_until** (*Optional[datetime.datetime]*) –

Return type *lawwenda.Share*

generate_wsgi ()

Return type *str*

get_share_by_name (*name*)

Parameters *name* (*str*) –

Return type *Optional[lawwenda.Share]*

get_shares ()

Return type *List[lawwenda.Share]*

property path

Return type *str*

Return type *str*

Return type *str*

peek_share_cache_tag (*name*)

Parameters *name* (*str*) –

Return type *Optional[object]*

remove_share (*name*)

Parameters *name* (*str*) –

Return type *None*

run_dev_server ()

Return type *None*

```
class lawwenda.Share(path, *, configuration, name, title, readonly=True, hide_by_patterns=(),
                    hide_by_tags=(), include_by_patterns=None, include_by_tags=None, ex-
                    clude_by_patterns=(), exclude_by_tags=(), exclude_hidden=False, pass-
                    word_script=None, password_salt=None, active_until_timestamp=None)
```

Bases: object

Parameters

- **path** (*str*) –
- **configuration** (*Configuration*) –
- **name** (*str*) –
- **title** (*str*) –
- **readonly** (*bool*) –
- **hide_by_patterns** (*Iterable[str]*) –
- **hide_by_tags** (*Iterable[str]*) –
- **include_by_patterns** (*Optional[Iterable[str]]*) –
- **include_by_tags** (*Optional[Iterable[str]]*) –
- **exclude_by_patterns** (*Iterable[str]*) –
- **exclude_by_tags** (*Iterable[str]*) –
- **exclude_hidden** (*bool*) –
- **password_script** (*Optional[str]*) –
- **password_salt** (*Optional[str]*) –
- **active_until_timestamp** (*Optional[float]*) –

_set_cache_tag (*t*)

Parameters **t** (*object*) –

Return type None

_to_dict ()

property active_until

Return type *t.Optional[datetime.datetime]*

Return type *t.Optional[datetime.datetime]*

Return type *t.Optional[datetime.datetime]*

property active_until_timestamp

Return type *t.Optional[float]*

Return type *t.Optional[float]*

Return type *t.Optional[float]*

property cache_tag

Return type *object*

Return type *object*

Return type *object*

property configuration

Return type *Configuration*

Return type *Configuration*

Return type *Configuration*

property exclude_by_patterns

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property exclude_by_tags

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property exclude_hidden

Return type `bool`

Return type `bool`

Return type `bool`

property hide_by_patterns

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property hide_by_tags

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property include_by_patterns

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property include_by_tags

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property is_active

Return type `bool`

Return type `bool`

Return type `bool`

property name

Return type str

Return type str

Return type str

property password_salt

Return type t.Optional[str]

Return type t.Optional[str]

Return type t.Optional[str]

property password_script

Return type t.Optional[str]

Return type t.Optional[str]

Return type t.Optional[str]

property path

Return type str

Return type str

Return type str

property readonly

Return type bool

Return type bool

Return type bool

property title

Return type str

Return type str

Return type str

PYTHON MODULE INDEX

I

- lawwenda, [23](#)
- lawwenda.__aux, [3](#)
- lawwenda.__aux.projectinformations, [3](#)
- lawwenda.datafiles, [9](#)
- lawwenda.fmapp, [10](#)
- lawwenda.fs, [12](#)
- lawwenda.lawwenda_cli, [22](#)
- lawwenda.previewer, [4](#)
- lawwenda.previewer.commonimages, [3](#)
- lawwenda.previewer.commonvideos, [4](#)
- lawwenda.search, [5](#)
- lawwenda.server, [23](#)
- lawwenda.sharehttpapp, [8](#)
- lawwenda.sharehttpapp.davprop, [7](#)

Symbols

__PredicateFactory (class in lawwenda.fs), 21
 __RenderTemplateValue (class in lawwenda.fmapp), 11
 __TempZip__is_executable() (lawwenda.fmapp._TempZips._TempZip method), 11
 __TempZip__putnode() (lawwenda.fmapp._TempZips._TempZip method), 11
 __TempZips (class in lawwenda.fmapp), 11
 __TempZips._TempZip (class in lawwenda.fmapp), 11
 __TempZips__cleanup_loop() (lawwenda.fmapp._TempZips class method), 11
 __auth() (lawwenda.fmapp.FmApp method), 10
 __decide() (lawwenda.search.ByGeoSearch method), 5
 __decide() (lawwenda.search.ByNameSearch method), 5
 __decide() (lawwenda.search.ByTagSearch method), 5
 __decide_unindexed() (lawwenda.search.DeeplySearch method), 5
 __dispatch_request() (lawwenda.fmapp.FmApp method), 10
 __ensure_authed() (lawwenda.fmapp.FmApp method), 10
 __ffmpeg_thumb() (lawwenda.preview.commonvideos.CommonVideosPreviewer method), 4
 __get_app_for_share() (lawwenda.server.Server method), 23
 __on_api_copymove() (lawwenda.fmapp.FmApp method), 10
 __path_to_fullpath() (lawwenda.fs.LocalFilesystem method), 20
 __query_indexed() (lawwenda.search.DeeplySearch method), 6
 __query_unindexed() (lawwenda.search.DeeplySearch method), 6
 __render_template() (lawwenda.fmapp.FmApp method), 10
 __render_template_text() (lawwenda.fmapp.FmApp method), 10
 __render_template_text_raw() (lawwenda.fmapp.FmApp method), 10
 __try_get_index() (lawwenda.search.DeeplySearch method), 6
 __verify_valid_name() (lawwenda.Configuration method), 23
 _abc_impl (lawwenda.preview.Previewer attribute), 4
 _abc_impl (lawwenda.preview.commonimages.CommonImagesPreviewer attribute), 3
 _abc_impl (lawwenda.preview.commonvideos.CommonVideosPreviewer attribute), 4
 _cleanup_thread (lawwenda.fmapp._TempZips attribute), 11
 _eval_predicate() (lawwenda.fs.AbstractFilesystemDecorator method), 12
 _filesystem() (lawwenda.fs.Filesystem.Node property), 12
 _fmt_cmd() (in module lawwenda.lawwenda_cli), 22
 _image_scale_to_thumbnail_size() (lawwenda.preview.Previewer method), 4
 _lock (lawwenda.fmapp._TempZips attribute), 11
 _method_copy() (lawwenda.sharehttpapp.ShareHttpApp method), 8
 _method_delete() (lawwenda.sharehttpapp.ShareHttpApp method), 8
 _method_get() (lawwenda.sharehttpapp.ShareHttpApp method), 8
 _method_head() (lawwenda.sharehttpapp.ShareHttpApp method), 8
 _method_mkcol() (lawwenda.sharehttpapp.ShareHttpApp method), 8
 _method_move() (lawwenda.sharehttpapp.ShareHttpApp method), 9

`_method_options()`
(*lawwenda.sharehttpapp.ShareHttpApp method*), 9

`_method_propfind()`
(*lawwenda.sharehttpapp.ShareHttpApp method*), 9

`_method_proppatch()`
(*lawwenda.sharehttpapp.ShareHttpApp method*), 9

`_method_put()` (*lawwenda.sharehttpapp.ShareHttpApp method*), 9

`_on_api_add_tag_to_entries()`
(*lawwenda.fmapp.FmApp method*), 10

`_on_api_copy()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_delete()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_details()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_dir()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_known_tags()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_mkdir()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_move()` (*lawwenda.fmapp.FmApp method*), 10

`_on_api_remove_tag_from_entries()`
(*lawwenda.fmapp.FmApp method*), 11

`_on_api_rename()` (*lawwenda.fmapp.FmApp method*), 11

`_on_api_set_comment()`
(*lawwenda.fmapp.FmApp method*), 11

`_on_api_set_geo()` (*lawwenda.fmapp.FmApp method*), 11

`_on_api_set_rating()` (*lawwenda.fmapp.FmApp method*), 11

`_on_api_thumbnail()` (*lawwenda.fmapp.FmApp method*), 11

`_on_api_upload()` (*lawwenda.fmapp.FmApp method*), 11

`_on_api_zip()` (*lawwenda.fmapp.FmApp method*), 11

`_on_api_zip_download()`
(*lawwenda.fmapp.FmApp method*), 11

`_on_help()` (*lawwenda.fmapp.FmApp method*), 11

`_on_static()` (*lawwenda.fmapp.FmApp method*), 11

`_query_by_tree_traversal()` (*in module lawwenda.search*), 6

`_quickstart()` (*in module lawwenda.lawwenda_cli*), 22

`_set_cache_tag()` (*lawwenda.Share method*), 25

`_termstring_to_termlist()` (*in module lawwenda.search*), 6

`_to_dict()` (*lawwenda.Share method*), 25

`_zips` (*lawwenda.fmapp._TempZips attribute*), 11

A

`AbstractFilesystemDecorator` (*class in lawwenda.fs*), 12

`active_until()` (*lawwenda.Share property*), 25

`active_until_timestamp()` (*lawwenda.Share property*), 25

`add_share()` (*in module lawwenda.lawwenda_cli*), 22

`add_share()` (*lawwenda.Configuration method*), 23

`add_tag()` (*lawwenda.fs.Filesystem method*), 16

`add_tag()` (*lawwenda.fs.Filesystem.Node method*), 12

`add_tag()` (*lawwenda.fs.LocalFilesystem method*), 20

`allowed_methods()`
(*lawwenda.sharehttpapp.ShareHttpApp property*), 9

`app()` (*lawwenda.server.Server._ShareAppInst property*), 23

B

`basics_as_dict()` (*lawwenda.fs.Filesystem.Node property*), 13

`ByGeoSearch` (*class in lawwenda.search*), 5

`ByNameSearch` (*class in lawwenda.search*), 5

`ByTagSearch` (*class in lawwenda.search*), 5

`bytes()` (*lawwenda.fmapp._TempZips._TempZip property*), 11

C

`cache_tag()` (*lawwenda.Share property*), 25

`child_by_name()` (*lawwenda.fs.Filesystem.Node method*), 13

`child_nodes()` (*lawwenda.fs.ExcludeNodesFilesystemAdapter method*), 12

`child_nodes()` (*lawwenda.fs.Filesystem method*), 16

`child_nodes()` (*lawwenda.fs.Filesystem.Node property*), 13

`child_nodes()` (*lawwenda.fs.LocalFilesystem method*), 20

`CircularTraversalError`, 12

`comment()` (*lawwenda.fs.Filesystem method*), 16

`comment()` (*lawwenda.fs.Filesystem.Node property*), 13

`comment()` (*lawwenda.fs.LocalFilesystem method*), 20

`CommentDavProp` (*class in lawwenda.sharehttpapp.davprop*), 7

`CommonImagesPreviewer` (*class in lawwenda.preview.commonimages*), 3

`CommonVideosPreviewer` (*class in lawwenda.preview.commonvideos*), 4

`Configuration` (*class in lawwenda*), 23

`configuration()` (*lawwenda.Share property*), 25

`console()` (*in module lawwenda.lawwenda_cli*), 22

copy_to() (*lawwenda.fs.Filesystem method*), 16
 copy_to() (*lawwenda.fs.Filesystem.Node method*), 13
 copy_to() (*lawwenda.fs.LocalFilesystem method*), 20
 create_filesystem() (*in module lawwenda.fs*), 21
 create_search() (*in module lawwenda.search*), 6
 create_tempzip() (*lawwenda.fmapp._TempZips class method*), 11

D

davname() (*lawwenda.sharehttpapp.davprop.DavProp property*), 7
 DavProp (*class in lawwenda.sharehttpapp.davprop*), 7
 DeeplySearch (*class in lawwenda.search*), 5
 delete() (*lawwenda.fs.Filesystem method*), 16
 delete() (*lawwenda.fs.Filesystem.Node method*), 13
 delete() (*lawwenda.fs.LocalFilesystem method*), 20
 describe_share() (*in module lawwenda.lawwenda_cli*), 22
 dirpath() (*lawwenda.fs.Filesystem.Node property*), 13

E

exclude_by_patterns() (*lawwenda.Share property*), 26
 exclude_by_tags() (*lawwenda.Share property*), 26
 exclude_hidden() (*lawwenda.Share property*), 26
 ExcludeNodesFilesystemAdapter (*class in lawwenda.fs*), 12
 exists() (*lawwenda.fs.Filesystem method*), 16
 exists() (*lawwenda.fs.Filesystem.Node property*), 13
 exists() (*lawwenda.fs.LocalFilesystem method*), 20

F

false() (*lawwenda.fs._PredicateFactory static method*), 21
 Filesystem (*class in lawwenda.fs*), 12
 Filesystem.Node (*class in lawwenda.fs*), 12
 Filesystem.ReadHandle (*class in lawwenda.fs*), 16
 Filesystem.WriteHandle (*class in lawwenda.fs*), 16
 find_data_file() (*in module lawwenda.datafiles*), 9
 FmApp (*class in lawwenda.fmapp*), 10
 full_as_dict() (*lawwenda.fs.Filesystem.Node property*), 13

G

generate_wsgi() (*in module lawwenda.lawwenda_cli*), 22
 generate_wsgi() (*lawwenda.Configuration method*), 24
 geo() (*lawwenda.fs.Filesystem method*), 16
 geo() (*lawwenda.fs.Filesystem.Node property*), 13

geo() (*lawwenda.fs.LocalFilesystem method*), 20
 geo_obj() (*lawwenda.fs.Filesystem.Node property*), 13
 GeoDavProp (*class in lawwenda.sharehttpapp.davprop*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.CommentDavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.DavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.GeoDavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.MtimeDavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.RatingDavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.ResourceTypeDavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.SizeDavProp method*), 7
 get_for_node() (*lawwenda.sharehttpapp.davprop.TagsDavProp method*), 8
 get_prop_by_davname() (*in module lawwenda.sharehttpapp.davprop*), 8
 get_readhandle() (*lawwenda.fs.ExcludeNodesFilesystemAdapter method*), 12
 get_readhandle() (*lawwenda.fs.Filesystem method*), 17
 get_readhandle() (*lawwenda.fs.LocalFilesystem method*), 20
 get_share_by_name() (*lawwenda.Configuration method*), 24
 get_shares() (*lawwenda.Configuration method*), 24
 get_tempzip() (*lawwenda.fmapp._TempZips class method*), 12
 get_writehandle() (*lawwenda.fs.ExcludeNodesFilesystemAdapter method*), 12
 get_writehandle() (*lawwenda.fs.Filesystem method*), 17
 get_writehandle() (*lawwenda.fs.LocalFilesystem method*), 20
 get_writehandle() (*lawwenda.fs.ReadOnlyFilesystemAdapter method*), 21

H

has_preview() (*lawwenda.fs.Filesystem method*), 17
 has_preview() (*lawwenda.fs.Filesystem.Node property*), 13
 has_preview() (*lawwenda.fs.LocalFilesystem method*), 20
 has_thumbnail() (*lawwenda.fs.Filesystem method*), 17

`has_thumbnail()` (*lawwenda.fs.Filesystem.Node property*), 13

`has_thumbnail()` (*lawwenda.fs.LocalFilesystem method*), 20

`hide_by_patterns()` (*lawwenda.Share property*), 26

`hide_by_tags()` (*lawwenda.Share property*), 26

`HideNodesFilesystemAdapter` (class in *lawwenda.fs*), 19

I

`icon_name()` (*lawwenda.fs.Filesystem.Node property*), 13

`include_by_patterns()` (*lawwenda.Share property*), 26

`include_by_tags()` (*lawwenda.Share property*), 26

`is_active()` (*lawwenda.server.Server_ShareAppInst property*), 23

`is_active()` (*lawwenda.Share property*), 26

`is_dir()` (*lawwenda.fs.Filesystem method*), 17

`is_dir()` (*lawwenda.fs.Filesystem.Node property*), 13

`is_dir()` (*lawwenda.fs.LocalFilesystem method*), 20

`is_file()` (*lawwenda.fs.Filesystem method*), 17

`is_file()` (*lawwenda.fs.Filesystem.Node property*), 14

`is_file()` (*lawwenda.fs.LocalFilesystem method*), 20

`is_hidden()` (*lawwenda.fs.Filesystem method*), 17

`is_hidden()` (*lawwenda.fs.Filesystem.Node property*), 14

`is_hidden()` (*lawwenda.fs.HideNodesFilesystemAdapter method*), 20

`is_hidden()` (*lawwenda.fs.LocalFilesystem method*), 20

`is_link()` (*lawwenda.fs.Filesystem method*), 17

`is_link()` (*lawwenda.fs.Filesystem.Node property*), 14

`is_link()` (*lawwenda.fs.LocalFilesystem method*), 20

`is_previewable()` (*lawwenda.preview.commonimages.CommonImagesPreviewer method*), 3

`is_previewable()` (*lawwenda.preview.commonvideos.CommonVideosPreviewer method*), 4

`is_previewable()` (*lawwenda.preview.Previewer method*), 4

`is_thumbnailable()` (*lawwenda.preview.commonimages.CommonImagesPreviewer method*), 3

`is_thumbnailable()` (*lawwenda.preview.commonvideos.CommonVideosPreviewer method*), 4

`is_thumbnailable()` (*lawwenda.preview.Previewer method*), 4

`is_writable()` (*lawwenda.fs.Filesystem.Node property*), 14

`is_writable()` (*lawwenda.sharehttpapp.davprop.DavProp property*), 7

`IsAbleRequest` (class in *lawwenda.preview*), 4

J

`JsonSerializedResponse` (class in *lawwenda.fmapp*), 11

`JsonSerializedResponse` (class in *lawwenda.sharehttpapp*), 8

K

`known_tags()` (*lawwenda.fs.Filesystem method*), 17

`known_tags()` (*lawwenda.fs.LocalFilesystem method*), 20

L

`lawwenda` module, 23

`lawwenda._aux` module, 3

`lawwenda._aux.projectinformations` module, 3

`lawwenda.datafiles` module, 9

`lawwenda.fmapp` module, 10

`lawwenda.fs` module, 12

`lawwenda.lawwenda_cli` module, 22

`lawwenda.preview` module, 4

`lawwenda.preview.commonimages` module, 3

`lawwenda.preview.commonvideos` module, 4

`lawwenda.search` module, 9

`lawwenda.server` module, 23

`lawwenda.sharehttpapp` module, 8

`lawwenda.sharehttpapp.davprop` module, 7

`list_shares()` (in module *lawwenda.lawwenda_cli*), 22

`LocalFilesystem` (class in *lawwenda.fs*), 20

M

`main()` (in module *lawwenda.lawwenda_cli*), 23

`mimetype` (*lawwenda.preview.IsAbleRequest attribute*), 4

`mimetype()` (*lawwenda.fs.Filesystem method*), 17

[mimetype\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 14
[mimetype\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[mkdir\(\)](#) (*lawwenda.fs.Filesystem* method), 17
[mkdir\(\)](#) (*lawwenda.fs.Filesystem.Node* method), 14
[mkdir\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[module](#)
 [lawwenda](#), 23
 [lawwenda._aux](#), 3
 [lawwenda._aux.projectinformations](#), 3
 [lawwenda.datafiles](#), 9
 [lawwenda.fmapp](#), 10
 [lawwenda.fs](#), 12
 [lawwenda.lawwenda_cli](#), 22
 [lawwenda.previewer](#), 4
 [lawwenda.previewer.commonimages](#), 3
 [lawwenda.previewer.commonvideos](#), 4
 [lawwenda.search](#), 5
 [lawwenda.server](#), 23
 [lawwenda.sharehttpapp](#), 8
 [lawwenda.sharehttpapp.davprop](#), 7
[move_to\(\)](#) (*lawwenda.fs.Filesystem* method), 17
[move_to\(\)](#) (*lawwenda.fs.Filesystem.Node* method), 14
[move_to\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[mtime\(\)](#) (*lawwenda.fs.Filesystem* method), 18
[mtime\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 14
[mtime\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[mtime_ts\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 14
[MtimeDavProp](#) (class in *lawwenda.sharehttpapp.davprop*), 7

N

[name](#) (*lawwenda.previewer.IsAbleRequest* attribute), 4
[name\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 14
[name\(\)](#) (*lawwenda.Share* property), 26
[node_by_path\(\)](#) (*lawwenda.fs.Filesystem* method), 18

P

[p_descending\(\)](#) (*lawwenda.fs._PredicateFactory* static method), 21
[p_not\(\)](#) (*lawwenda.fs._PredicateFactory* static method), 21
[p_or\(\)](#) (*lawwenda.fs._PredicateFactory* static method), 21
[p_regexp\(\)](#) (*lawwenda.fs._PredicateFactory* static method), 21
[p_tag\(\)](#) (*lawwenda.fs._PredicateFactory* static method), 21
[parent_node\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 14
[password_salt\(\)](#) (*lawwenda.Share* property), 27

[password_script\(\)](#) (*lawwenda.Share* property), 27
[path\(\)](#) (*lawwenda.Configuration* property), 24
[path\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 14
[path\(\)](#) (*lawwenda.Share* property), 27
[peek_share_cache_tag\(\)](#)
 (*lawwenda.Configuration* method), 24
[preview_html\(\)](#) (*lawwenda.fs.Filesystem* method), 18
[preview_html\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 15
[preview_html\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[preview_html\(\)](#) (*lawwenda.previewer.commonimages.CommonImages* method), 3
[preview_html\(\)](#) (*lawwenda.previewer.commonvideos.CommonVideos* method), 4
[preview_html\(\)](#) (*lawwenda.previewer.Previewer* method), 4
[Previewer](#) (class in *lawwenda.previewer*), 4

Q

[query\(\)](#) (*lawwenda.search.ByGeoSearch* method), 5
[query\(\)](#) (*lawwenda.search.ByNameSearch* method), 5
[query\(\)](#) (*lawwenda.search.ByTagSearch* method), 5
[query\(\)](#) (*lawwenda.search.DeeplySearch* method), 6
[query\(\)](#) (*lawwenda.search.Search* method), 6

R

[rating\(\)](#) (*lawwenda.fs.Filesystem* method), 18
[rating\(\)](#) (*lawwenda.fs.Filesystem.Node* property), 15
[rating\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[RatingDavProp](#) (class in *lawwenda.sharehttpapp.davprop*), 7
[read_file\(\)](#) (*lawwenda.fs.Filesystem* method), 18
[read_file\(\)](#) (*lawwenda.fs.Filesystem.Node* method), 15
[read_file\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[readonly\(\)](#) (*lawwenda.Share* property), 27
[ReadOnlyFilesystemAdapter](#) (class in *lawwenda.fs*), 21
[register_prop\(\)](#) (in *module lawwenda.sharehttpapp.davprop*), 8
[remove_share\(\)](#) (in *module lawwenda.lawwenda_cli*), 23
[remove_share\(\)](#) (*lawwenda.Configuration* method), 24
[remove_tag\(\)](#) (*lawwenda.fs.Filesystem* method), 18
[remove_tag\(\)](#) (*lawwenda.fs.Filesystem.Node* method), 15
[remove_tag\(\)](#) (*lawwenda.fs.LocalFilesystem* method), 20
[Request](#) (class in *lawwenda.fmapp*), 11
[Request](#) (class in *lawwenda.sharehttpapp*), 8

ResourceTypeDavProp (class in lawwenda.sharehttpapp.davprop), 7
rootnode() (lawwenda.fs.Filesystem property), 18
run_dev_server() (in module lawwenda.lawwenda_cli), 23
run_dev_server() (lawwenda.Configuration method), 24

S

sanitize_abspath() (lawwenda.fs.Filesystem static method), 18
Search (class in lawwenda.search), 6
Server (class in lawwenda.server), 23
Server._ShareAppInst (class in lawwenda.server), 23
set_comment() (lawwenda.fs.Filesystem method), 18
set_comment() (lawwenda.fs.Filesystem.Node method), 15
set_comment() (lawwenda.fs.LocalFilesystem method), 20
set_geo() (lawwenda.fs.Filesystem method), 19
set_geo() (lawwenda.fs.Filesystem.Node method), 15
set_geo() (lawwenda.fs.LocalFilesystem method), 20
set_rating() (lawwenda.fs.Filesystem method), 19
set_rating() (lawwenda.fs.Filesystem.Node method), 15
set_rating() (lawwenda.fs.LocalFilesystem method), 20
Share (class in lawwenda), 24
ShareHttpApp (class in lawwenda.sharehttpapp), 8
size() (lawwenda.fs.Filesystem method), 19
size() (lawwenda.fs.Filesystem.Node property), 15
size() (lawwenda.fs.LocalFilesystem method), 21
SizeDavProp (class in lawwenda.sharehttpapp.davprop), 7

T

tags() (lawwenda.fs.Filesystem method), 19
tags() (lawwenda.fs.Filesystem.Node property), 15
tags() (lawwenda.fs.LocalFilesystem method), 21
TagsDavProp (class in lawwenda.sharehttpapp.davprop), 7
tagstring() (lawwenda.fs.Filesystem.Node property), 15
thumbheight (lawwenda.preview.Previewer attribute), 5
thumbnail() (lawwenda.fs.Filesystem method), 19
thumbnail() (lawwenda.fs.Filesystem.Node method), 15
thumbnail() (lawwenda.fs.LocalFilesystem method), 21
thumbnail() (lawwenda.preview.commonimages.CommonImagesPreviewer method), 3

thumbnail() (lawwenda.preview.commonvideos.CommonVideosPreviewer method), 4
thumbnail() (lawwenda.preview.Previewer method), 5
thumbsize (lawwenda.preview.Previewer attribute), 5
thumbwidth (lawwenda.preview.Previewer attribute), 5
title() (lawwenda.Share property), 27
traverse_dir() (lawwenda.fs.Filesystem.Node method), 15
try_get_fullpath() (lawwenda.fs.Filesystem method), 19
try_get_fullpath() (lawwenda.fs.Filesystem.Node method), 15
try_get_fullpath() (lawwenda.fs.LocalFilesystem method), 21

U

unesaped() (lawwenda.fmapp._RenderTemplateValue property), 11
url_internals_name (lawwenda.fmapp.FmApp attribute), 11

W

write_file() (lawwenda.fs.Filesystem method), 19
write_file() (lawwenda.fs.Filesystem.Node method), 15
write_file() (lawwenda.fs.LocalFilesystem method), 21