
lawwenda - Plugin API Documentation

Release 0.2.113

Josef Hahn

Mar 20, 2021

CONTENTS

1	API reference	3
1.1	lawwenda package	3
	Python Module Index	33
	Index	35

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.preview package

Submodules

lawwenda.preview.commonimages module

class lawwenda.preview.commonimages.CommonImagesPreviewer

Bases: *lawwenda.preview.Previewer*

_abc_impl = *<_abc_data object>*

is_previewable (*i***ar**)

Return if this previewer is able to generate an html preview snippet.

Note that this decision has to be made with very few information, e.g. without knowing the file content, for performance reasons.

See also *preview_html()*.

Parameters **i****ar** – An object that describes the file.

is_thumbnailable (*i***ar**)

Return if this previewer is able to generate a thumbnail for a file.

Note that this decision has to be made with very few information, e.g. without knowing the file content, for performance reasons.

See also *thumbnail()*.

Parameters **i****ar** – An object that describes the file.

preview_html (*n***ode**)

Return an html snippet that shows a preview of a file.

This is larger and richer in terms of flexibility than thumbnails, and is typically used by the file details panel.

See also *is_previewable()*.

Parameters **n****ode** – The node to preview.

thumbnail (*n***ode**)

lawwenda.preview.commonvideos module

class lawwenda.preview.commonvideos.CommonVideosPreviewer

Bases: [lawwenda.preview.Previewer](#)

__ffmpeg_thumb (*stdin*=- 3)

Parameters

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

Return type bytes

_abc_impl = <_abc_data object>

is_previewable (*iar*)

Return if this previewer is able to generate an html preview snippet.

Note that this decision has to be made with very few information, e.g. without knowing the file content, for performance reasons.

See also [preview_html\(\)](#).

Parameters **iar** – An object that describes the file.

is_thumbnailable (*iar*)

Return if this previewer is able to generate a thumbnail for a file.

Note that this decision has to be made with very few information, e.g. without knowing the file content, for performance reasons.

See also [thumbnail\(\)](#).

Parameters **iar** – An object that describes the file.

preview_html (*node*)

Return an html snippet that shows a preview of a file.

This is larger and richer in terms of flexibility than thumbnails, and is typically used by the file details panel.

See also [is_previewable\(\)](#).

Parameters **node** – The node to preview.

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

Base class for previewers.

Subclasses implement thumbnail and preview functionality for particular file formats.

_abc_impl = <_abc_data object>

_image_scale_to_thumbnail_size (*imgdata*)

Parameters *imgdata* (*BinaryIO*) –

Return type *_io.BytesIO*

abstract is_previewable (*iar*)

Return if this previewer is able to generate an html preview snippet.

Note that this decision has to be made with very few information, e.g. without knowing the file content, for performance reasons.

See also *preview_html()*.

Parameters *iar* (*lawwenda.previewer.IsAbleRequest*) – An object that describes the file.

Return type *bool*

abstract is_thumbnailable (*iar*)

Return if this previewer is able to generate a thumbnail for a file.

Note that this decision has to be made with very few information, e.g. without knowing the file content, for performance reasons.

See also *thumbnail()*.

Parameters *iar* (*lawwenda.previewer.IsAbleRequest*) – An object that describes the file.

Return type *bool*

abstract preview_html (*node*)

Return an html snippet that shows a preview of a file.

This is larger and richer in terms of flexibility than thumbnails, and is typically used by the file details panel.

See also *is_previewable()*.

Parameters *node* (*lawwenda.fs.Filesystem.Node*) – The node to preview.

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

__decide_unindexed_by_content (*node*)

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

Return type bool

__decide_unindexed_by_metadata (*node*)

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

Return type bool

__query_indexed (*idxdata*)

Parameters

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

Return type FileSystemNodes

__query_unindexed ()

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
        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
        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]) –

    Return type Type[lawwenda.sharehttpapp.davprop.DavProp]
```

Module contents

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

    A wsgi application that provides an http interface to a filesystem.

    It provides bare http access, and also WebDAV extensions for a more complete interface. It does not provide an
    own user interface.

    Parameters filesystem (lawwenda.fs.Filesystem) –

property __allowed_methods

    Return type t.List[str]

    Return type t.List[str]

    Return type t.List[str]

    Return type t.List[str]

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

    Parameters

    • request (lawwenda.comm.Request) –

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

    • xfermethod (str) –

__method_delete (request, fsnode)

    Parameters

    • request (lawwenda.comm.Request) –

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

__method_get (request, fsnode, is_head_request=False)

    Parameters

    • request (lawwenda.comm.Request) –
```

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

- **is_head_request** (`bool`) –

_method_head (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

_method_mkcol (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

_method_move (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

_method_options (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

_method_propfind (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

_method_proppatch (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

_method_put (`request, fsnode`)

Parameters

- **request** (`lawwenda.comm.Request`) –

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

1.1.2 Submodules

1.1.3 lawwenda.comm module

Internal communication helpers.

class lawwenda.comm.JsonedResponse (data, **kwargs)

Bases: werkzeug.wrappers.response.Response

A werkzeug response that serializes response data to json.

Parameters data (*Optional[Any]*) – Response data. Can be anything that *json* is able to serialize.

class lawwenda.comm.Request (environ, populate_request=True, shallow=False)

Bases: werkzeug.wrappers.request.Request, werkzeug.wrappers.json.JSONMixin

A werkzeug request that is able to interpret json data in the response body.

1.1.4 lawwenda.datafiles module

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

Return the absolute path of a Lawwenda data file.

Can return *None* if no such file exists.

Parameters

- **fname** (*str*) – The name of the file to find.
- **searchdirs** (*Optional[Iterable[str]]*) – List of directories to look into. If not specified (as it usually should be), it looks at some usual places.

Return type Optional[str]

1.1.5 lawwenda.devserver module

class lawwenda.devserver._DevServerInfo (svr, svrthread)

Bases: object

shutdown ()

Return type None

property url

Return type str

Return type str

Return type str

Return type str

wait_stopped ()

Return type None

class lawwenda.devserver._DevServerThread (svr)

Bases: threading.Thread

This constructor should always be called with keyword arguments. Arguments are:

group should be None; reserved for future extension when a ThreadGroup class is implemented.

target is the callable object to be invoked by the run() method. Defaults to None, meaning nothing is called.

name is the thread name. By default, a unique name is constructed of the form “Thread-N” where N is a small decimal number.

args is the argument tuple for the target invocation. Defaults to ().

kwargs is a dictionary of keyword arguments for the target invocation. Defaults to {}.

If a subclass overrides the constructor, it must make sure to invoke the base class constructor (Thread.__init__()) before doing anything else to the thread.

run()

Method representing the thread’s activity.

You may override this method in a subclass. The standard run() method invokes the callable object passed to the object’s constructor as the target argument, if any, with sequential and keyword arguments taken from the args and kwargs arguments, respectively.

`lawwenda.devserver.get_running_dev_servers()`

Return the servers started by `run_dev_server()` that are currently running.

Return type `Iterable[lawwenda.devserver.DevServerInfo]`

`lawwenda.devserver.run_dev_server(cfg)`

Start a tiny local server for a given configuration.

Such a server can be used for trying, development, testing, and so on, but is not recommended for real usage.

It will automatically find a free port and will return a control object that contains the full url, and more.

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

Return type `lawwenda.devserver.DevServerInfo`

1.1.6 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*)

`__dispatch_request_internals` (*request*)

`__dispatch_request_normal` (*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*, *, *csrftoken*, *html_head_inner*, ***kwargs*)**Parameters**

- **template** (*str*) –
- **csrftoken** (*str*) –
- **html_head_inner** (*str*) –

Return type `str`**__render_template_text_raw** (*template*, ***kwargs*)**Parameters** **template** (*str*) –**Return type** `str`**_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_rename** (*request*)**_on_api_set_comment** (*request*)**_on_api_set_geo** (*request*)**_on_api_set_rating** (*request*)**_on_api_tag_entries** (*request*)**_on_api_thumbnail** (*request*)**_on_api_untag_entries** (*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._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]
```

1.1.7 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
            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"]
            Return type t.List["Filesystem.Node"]

        property comment
            Return type str
            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
Return type str

property exists
Return type bool
Return type bool
Return type bool
Return type bool

property full_as_dict

property geo
Return type str
Return type str
Return type str
Return type str

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

property has_preview
Return type bool
Return type bool
Return type bool
Return type bool

property has_thumbnail
Return type bool
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]
Return type t.Optional[str]

property is_dir
Return type bool
Return type bool
Return type bool
Return type bool

property is_file
Return type bool
Return type bool
Return type bool
Return type bool

property is_hidden

Return type bool

Return type bool

Return type bool

Return type bool

property is_link

Return type bool

Return type bool

Return type bool

Return type bool

property is_writable

Return type bool

Return type bool

Return type bool

Return type bool

property mimetype

Return type str

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

Return type datetime.datetime

property mtime_ts

Return type float

Return type float

Return type float

Return type float

property name

Return type str

Return type str

Return type str

Return type str

property parent_node

Return type *Filesystem.Node*

Return type *Filesystem.Node*

Return type *Filesystem.Node*

Return type *Filesystem.Node*

property path

Return type str

Return type str

Return type str

Return type str

property preview_html
Return type str
Return type str
Return type str
Return type str

property rating
Return type int
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
Return type int

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

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 (exception: root path)
- 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 excludehidden(node)
        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*

1.1.8 lawwenda.lawwenda_cli module

The Lawwenda command line interface.

This tool is used by the end user for creating and managing shares and more. It can be called with some command line arguments for a particular action, or without parameters (which will open an interactive Python prompt for further actions).

Do not use this module in code. See `lawwenda.Configuration` instead.

```
lawwenda.lawwenda_cli._fmt_cmd(txt)
```

Parameters `txt` (*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, ex-
                                clude_by_pattern, exclude_by_tag, exclude_hidden, readonly)
```

Add a share.

See `lawwenda.Configuration.add_share()`.

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)
```

Open an interactive Python console.

The user will get a small help text and is then able to configure Lawwenda by typing Python code.

Parameters `cfg` (`lawwenda.Configuration`) – The initial value for the `cfg` variable inside the console.

Return type `None`

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

Return a textual share description (i.e. how it is configured) for a share.

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)`

Return a list of share names as string, one line for each share.

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

Return type `str`

`lawwenda.lawwenda_cli.main()`

Parse command line arguments and call the right function with its arguments.

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

Remove a share.

See `lawwenda.Configuration.remove_share()`.

Parameters

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

Return type `None`

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

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

Return type `None`

1.1.9 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`

Return type `lawwenda.fmapp.FmApp`

property **is_active**

Return type `bool`

Return type bool
Return type bool
Return type bool

`__get_app_for_share(sharename)`

Parameters `sharename` (*str*) –

1.1.10 Module contents

User side API for reading and modifying Lawwenda configurations, creating shares, and more.

For most cases, instantiate `Configuration` and use some of its methods.

class `lawwenda.Configuration` (*cfgpath=None*)

Bases: object

Parameters `cfgpath` (*Optional[str]*) – The path of the configuration directory. Will be created on demand if it does not exist. Defaults to a location that is usual for your operating system.

`__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)`

Add a new share.

Parameters

- `path` (*str*) – The directory to share. See `Share.path`.
- `name` (*str*) – The unique name of the new share. See `Share.name`.
- `password` (*Optional[str]*) – The password to protect the share with.
- `title` (*Optional[str]*) – The share title. See `Share.title`.
- `readonly` (*bool*) – If to share in a read-only way. See `Share.readonly`.
- `hide_by_patterns` (*Iterable[str]*) – See `Share.hide_by_patterns`.
- `hide_by_tags` (*Iterable[str]*) – See `Share.hide_by_tags`.
- `include_by_patterns` (*Optional[Iterable[str]]*) – See `Share.include_by_patterns`.
- `include_by_tags` (*Optional[Iterable[str]]*) – See `Share.include_by_tags`.
- `exclude_by_patterns` (*Iterable[str]*) – See `Share.exclude_by_patterns`.
- `exclude_by_tags` (*Iterable[str]*) – See `Share.exclude_by_tags`.
- `exclude_hidden` (*bool*) – See `Share.exclude_hidden`.
- `active_until` (*Optional[datetime.datetime]*) – The optional expiration time of the share. See `Share.active_until`.

Return type `lawwenda.Share`

generate_wsgi()

Return type str

get_share_by_name(*name*)

Return the share by a name (or *None* if it does not exist).

Parameters **name** (*str*) –

Return type Optional[[lawwenda.Share](#)]

get_shares()

Return all shares that are currently part of this configuration.

Return type List[[lawwenda.Share](#)]

property path

The path of the configuration directory.

Return type str

Return type str

Return type str

Return type str

peek_share_cache_tag(*name*)

Parameters **name** (*str*) –

Return type Optional[object]

remove_share(*name*)

Remove a share.

Parameters **name** (*str*) – The name of the share to remove.

Return type None

run_dev_server()

Return type [lawwenda.devserver.DevServerInfo](#)

class [lawwenda.Share](#)(*path*, *, *configuration*, *name*, *title*, *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*, *password_script=None*, *password_salt=None*, *active_until_timestamp=None*)

Bases: object

A directory path together with some parameters (e.g. for access control) for sharing via Lawwenda. Read the documentation for more about shares.

This class is not intended to be instantiated directly. You will get instances by some other api methods.

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 (*cachetag*)

Parameters **cachetag** (*object*) –

Return type *None*

_to_dict ()

property active_until

The expiration time of this share, or *None* for infinite.

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

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

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

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

property active_until_timestamp

Same as *active_until*, but as Unix timestamp.

Return type *t.Optional[float]*

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*

Return type *object*

property configuration

The configuration that contains this share.

Return type *Configuration*

Return type *Configuration*

Return type *Configuration*

Return type *Configuration*

property exclude_by_patterns

A list of regular expressions of paths for excluding.

A file or directory will be excluded if its path matches at least one of them. Those paths are always relative to *path*, always start with a '/', but never end with a one (unless it is the root path).

Exclusions are enforced on backend side and not just a presentation aspect. There is no way for a client to work around that (unless there is a software bug).

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property exclude_by_tags

A list of tags for excluding files and directories.

Exclusions are enforced on backend side and not just a presentation aspect. There is no way for a client to work around that (unless there is a software bug).

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property exclude_hidden

If to consider 'hidden' flags of files or directories as exclusions.

Exclusions are enforced on backend side and not just a presentation aspect. There is no way for a client to work around that (unless there is a software bug).

Return type `bool`

Return type `bool`

Return type `bool`

Return type `bool`

property hide_by_patterns

A list of regular expressions of paths for hiding.

A file or directory will be hidden if its path matches at least one of them. Those paths are always relative to *path*, always start with a '/', but never end with a one (unless it is the root path).

Note that hiding is not a security feature unless *exclude_hidden* is set.

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property hide_by_tags

A list of tags for hiding files and directories.

A file or directory will be hidden if it is tagged with at least one of them.

Note that hiding is not a security feature unless *exclude_hidden* is set.

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

Return type `t.Iterable[str]`

property include_by_patterns

A list of regular expressions of paths for including explicitly.

Those paths are always relative to *path*, always start with a '/', but never end with a one (unless it is the root path).

If this is specified, the share will switch from blacklist to whitelist. Everything that is not considered as included is implicitly considered as excluded.

Return type `t.Optional[t.Iterable[str]]`

Return type `t.Optional[t.Iterable[str]]`

Return type `t.Optional[t.Iterable[str]]`

Return type `t.Optional[t.Iterable[str]]`

property include_by_tags

A list of tags for including files and directories.

If this is specified, the share will switch from blacklist to whitelist. Everything that is not considered as included is implicitly considered as excluded.

Return type `t.Optional[t.Iterable[str]]`

Return type `t.Optional[t.Iterable[str]]`

Return type `t.Optional[t.Iterable[str]]`

Return type `t.Optional[t.Iterable[str]]`

property is_active

If this share is currently active (e.g. not yet expired; see *active_until*).

Return type `bool`

Return type `bool`

Return type `bool`

Return type `bool`

property name

The share name.

This usually makes the last part of the url to this share. Is unique in the containing *configuration*.

Return type `str`

Return type `str`

Return type `str`

Return type `str`

property password_salt

The hash salt of the password for this share, if password protected.

See also *password_script*.

Return type `t.Optional[str]`

Return type `t.Optional[str]`

Return type `t.Optional[str]`

Return type `t.Optional[str]`

property password_script

The script hash of the password for this share.

Empty or *None* for disabled password protection. See also [*password_salt*](#).

Return type `t.Optional[str]`

Return type `t.Optional[str]`

Return type `t.Optional[str]`

Return type `t.Optional[str]`

property path

The path of the share's root directory.

Return type `str`

Return type `str`

Return type `str`

Return type `str`

property readonly

If this share is restricted to only read actions (no removal, copying, uploading, editing, ... of the files and directories in [*path*](#)).

Return type `bool`

Return type `bool`

Return type `bool`

Return type `bool`

property title

The share title.

This is an arbitrary text shown in the window title. Should not contain line breaks and should be short.

Return type `str`

Return type `str`

Return type `str`

Return type `str`

PYTHON MODULE INDEX

I

- lawwenda, [27](#)
- lawwenda.comm, [11](#)
- lawwenda.datafiles, [11](#)
- lawwenda.devserver, [11](#)
- lawwenda.fmapp, [12](#)
- lawwenda.fs, [14](#)
- lawwenda.lawwenda_cli, [25](#)
- lawwenda.previewer, [4](#)
- lawwenda.previewer.commonimages, [3](#)
- lawwenda.previewer.commonvideos, [4](#)
- lawwenda.search, [6](#)
- lawwenda.server, [26](#)
- lawwenda.sharehttpapp, [9](#)
- lawwenda.sharehttpapp.davprop, [8](#)

Symbols

_DevServerInfo (class in lawwenda.devserver), 11
 _DevServerThread (class in lawwenda.devserver), 11
 _PredicateFactory (class in lawwenda.fs), 24
 _RenderTemplateValue (class in lawwenda.fmapp), 13
 _TempZip__is_executable() (lawwenda.fmapp._TempZips._TempZip method), 14
 _TempZip__putnode() (lawwenda.fmapp._TempZips._TempZip method), 14
 _TempZips (class in lawwenda.fmapp), 14
 _TempZips._TempZip (class in lawwenda.fmapp), 14
 _TempZips__cleanup_loop() (lawwenda.fmapp._TempZips class method), 14
 __allowed_methods() (lawwenda.sharehttpapp.ShareHttpApp property), 9
 __auth() (lawwenda.fmapp.FmApp method), 12
 __decide() (lawwenda.search.ByGeoSearch method), 6
 __decide() (lawwenda.search.ByNameSearch method), 6
 __decide() (lawwenda.search.ByTagSearch method), 6
 __decide_unindexed() (lawwenda.search.DeeplySearch method), 6
 __decide_unindexed_by_content() (lawwenda.search.DeeplySearch method), 6
 __decide_unindexed_by_metadata() (lawwenda.search.DeeplySearch method), 6
 __dispatch_request() (lawwenda.fmapp.FmApp method), 12
 __dispatch_request_internals() (lawwenda.fmapp.FmApp method), 12
 __dispatch_request_normal() (lawwenda.fmapp.FmApp method), 12
 __ensure_authed() (lawwenda.fmapp.FmApp method), 12
 __ffmpeg_thumb() (lawwenda.preview.commonvideos.CommonVideosPreviewer method), 4
 __get_app_for_share() (lawwenda.server.Server method), 27
 __on_api_copymove() (lawwenda.fmapp.FmApp method), 12
 __path_to_fullpath() (lawwenda.fs.LocalFilesystem method), 23
 __query_indexed() (lawwenda.search.DeeplySearch method), 7
 __query_unindexed() (lawwenda.search.DeeplySearch method), 7
 __render_template() (lawwenda.fmapp.FmApp method), 12
 __render_template_text() (lawwenda.fmapp.FmApp method), 13
 __render_template_text_raw() (lawwenda.fmapp.FmApp method), 13
 __try_get_index() (lawwenda.search.DeeplySearch method), 7
 __verify_valid_name() (lawwenda.Configuration method), 27
 _abc_impl (lawwenda.preview.Previewer attribute), 5
 _abc_impl (lawwenda.preview.commonimages.CommonImagesPreviewer attribute), 3
 _abc_impl (lawwenda.preview.commonvideos.CommonVideosPreviewer attribute), 4
 _cleanup_thread (lawwenda.fmapp._TempZips attribute), 14
 _eval_predicate() (lawwenda.fs.AbstractFilesystemDecorator method), 14
 _filesystem() (lawwenda.fs.Filesystem.Node property), 15
 _fmt_cmd() (in module lawwenda.lawwenda_cli), 25

`_image_scale_to_thumbnail_size()`
(*lawwenda.previewer.Previewer* method), 5

`_lock` (*lawwenda.fmapp._TempZips* attribute), 14

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

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

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

`_method_head()` (*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_method_mkcol()` (*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_method_move()` (*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_method_options()`
(*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_method_propfind()`
(*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_method_proppatch()`
(*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_method_put()` (*lawwenda.sharehttpapp.ShareHttpApp* method), 10

`_on_api_copy()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_delete()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_details()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_dir()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_known_tags()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_mkdir()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_move()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_rename()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_set_comment()`
(*lawwenda.fmapp.FmApp* method), 13

`_on_api_set_geo()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_set_rating()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_tag_entries()`
(*lawwenda.fmapp.FmApp* method), 13

`_on_api_thumbnail()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_untag_entries()`
(*lawwenda.fmapp.FmApp* method), 13

`_on_api_upload()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_zip()` (*lawwenda.fmapp.FmApp* method), 13

`_on_api_zip_download()`
(*lawwenda.fmapp.FmApp* method), 13

`_on_help()` (*lawwenda.fmapp.FmApp* method), 13

`_on_static()` (*lawwenda.fmapp.FmApp* method), 13

`_query_by_tree_traversal()` (in module *lawwenda.search*), 7

`_quickstart()` (in module *lawwenda.lawwenda_cli*), 25

`_set_cache_tag()` (*lawwenda.Share* method), 29

`_termstring_to_termlist()` (in module *lawwenda.search*), 7

`_to_dict()` (*lawwenda.Share* method), 29

`_zips` (*lawwenda.fmapp._TempZips* attribute), 14

A

`AbstractFilesystemDecorator` (class in *lawwenda.fs*), 14

`active_until()` (*lawwenda.Share* property), 29

`active_until_timestamp()` (*lawwenda.Share* property), 29

`add_share()` (in module *lawwenda.lawwenda_cli*), 25

`add_share()` (*lawwenda.Configuration* method), 27

`add_tag()` (*lawwenda.fs.Filesystem* method), 19

`add_tag()` (*lawwenda.fs.Filesystem.Node* method), 15

`add_tag()` (*lawwenda.fs.LocalFilesystem* method), 23

`app()` (*lawwenda.server.Server._ShareAppInst* property), 26

B

`basics_as_dict()` (*lawwenda.fs.Filesystem.Node* property), 15

`ByGeoSearch` (class in *lawwenda.search*), 6

`ByNameSearch` (class in *lawwenda.search*), 6

`ByTagSearch` (class in *lawwenda.search*), 6

`bytes()` (*lawwenda.fmapp._TempZips._TempZip* property), 14

C

`cache_tag()` (*lawwenda.Share* property), 29

`child_by_name()` (*lawwenda.fs.Filesystem.Node* method), 15

`child_nodes()` (*lawwenda.fs.ExcludeNodesFilesystemAdapter* method), 15

`child_nodes()` (*lawwenda.fs.Filesystem* method), 19

`child_nodes()` (*lawwenda.fs.Filesystem.Node* property), 15

`child_nodes()` (*lawwenda.fs.LocalFilesystem* method), 23

`CircularTraversalError`, 14

comment() (*lawwenda.fs.Filesystem method*), 19
 comment() (*lawwenda.fs.Filesystem.Node property*), 15
 comment() (*lawwenda.fs.LocalFilesystem method*), 23
 CommentDavProp (class in *lawwenda.sharehttpapp.davprop*), 8
 CommonImagesPreviewer (class in *lawwenda.preview.commonimages*), 3
 CommonVideosPreviewer (class in *lawwenda.preview.commonvideos*), 4
 Configuration (class in *lawwenda*), 27
 configuration() (*lawwenda.Share property*), 29
 console() (in module *lawwenda.lawwenda_cli*), 25
 copy_to() (*lawwenda.fs.Filesystem method*), 19
 copy_to() (*lawwenda.fs.Filesystem.Node method*), 15
 copy_to() (*lawwenda.fs.LocalFilesystem method*), 23
 create_filesystem() (in module *lawwenda.fs*), 24
 create_search() (in module *lawwenda.search*), 7
 create_tempzip() (*lawwenda.fmapp._TempZips class method*), 14

D

davname() (*lawwenda.sharehttpapp.davprop.DavProp property*), 8
 DavProp (class in *lawwenda.sharehttpapp.davprop*), 8
 DeeplySearch (class in *lawwenda.search*), 6
 delete() (*lawwenda.fs.Filesystem method*), 19
 delete() (*lawwenda.fs.Filesystem.Node method*), 15
 delete() (*lawwenda.fs.LocalFilesystem method*), 23
 describe_share() (in module *lawwenda.lawwenda_cli*), 25
 dirpath() (*lawwenda.fs.Filesystem.Node property*), 16

E

exclude_by_patterns() (*lawwenda.Share property*), 29
 exclude_by_tags() (*lawwenda.Share property*), 30
 exclude_hidden() (*lawwenda.Share property*), 30
 excludehidden() (*lawwenda.fs._PredicateFactory static method*), 24
 ExcludeNodesFilesystemAdapter (class in *lawwenda.fs*), 14
 exists() (*lawwenda.fs.Filesystem method*), 19
 exists() (*lawwenda.fs.Filesystem.Node property*), 16
 exists() (*lawwenda.fs.LocalFilesystem method*), 23

F

false() (*lawwenda.fs._PredicateFactory static method*), 24
 Filesystem (class in *lawwenda.fs*), 15
 Filesystem.Node (class in *lawwenda.fs*), 15
 Filesystem.ReadHandle (class in *lawwenda.fs*), 19

Filesystem.WriteHandle (class in *lawwenda.fs*), 19
 find_data_file() (in module *lawwenda.datafiles*), 11
 FmApp (class in *lawwenda.fmapp*), 12
 full_as_dict() (*lawwenda.fs.Filesystem.Node property*), 16

G

generate_wsgi() (in module *lawwenda.lawwenda_cli*), 26
 generate_wsgi() (*lawwenda.Configuration method*), 27
 geo() (*lawwenda.fs.Filesystem method*), 20
 geo() (*lawwenda.fs.Filesystem.Node property*), 16
 geo() (*lawwenda.fs.LocalFilesystem method*), 23
 geo_obj() (*lawwenda.fs.Filesystem.Node property*), 16
 GeoDavProp (class in *lawwenda.sharehttpapp.davprop*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.CommentDavProp method*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.DavProp method*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.GeoDavProp method*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.MtimeDavProp method*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.RatingDavProp method*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.ResourceTypeDavProp method*), 8
 get_for_node() (*lawwenda.sharehttpapp.davprop.SizeDavProp method*), 9
 get_for_node() (*lawwenda.sharehttpapp.davprop.TagsDavProp method*), 9
 get_prop_by_davname() (in module *lawwenda.sharehttpapp.davprop*), 9
 get_readhandle() (*lawwenda.fs.ExcludeNodesFilesystemAdapter method*), 15
 get_readhandle() (*lawwenda.fs.Filesystem method*), 20
 get_readhandle() (*lawwenda.fs.LocalFilesystem method*), 23
 get_running_dev_servers() (in module *lawwenda.devserver*), 12
 get_share_by_name() (*lawwenda.Configuration method*), 28
 get_shares() (*lawwenda.Configuration method*), 28
 get_tempzip() (*lawwenda.fmapp._TempZips class method*), 14
 get_writehandle() (*lawwenda.fs.ExcludeNodesFilesystemAdapter method*), 15

`get_writehandle()` (*lawwenda.fs.Filesystem method*), 20
`get_writehandle()` (*lawwenda.fs.LocalFilesystem method*), 23
`get_writehandle()` (*lawwenda.fs.ReadOnlyFilesystemAdapter method*), 24

H

`has_preview()` (*lawwenda.fs.Filesystem method*), 20
`has_preview()` (*lawwenda.fs.Filesystem.Node property*), 16
`has_preview()` (*lawwenda.fs.LocalFilesystem method*), 23
`has_thumbnail()` (*lawwenda.fs.Filesystem method*), 20
`has_thumbnail()` (*lawwenda.fs.Filesystem.Node property*), 16
`has_thumbnail()` (*lawwenda.fs.LocalFilesystem method*), 23
`hide_by_patterns()` (*lawwenda.Share property*), 30
`hide_by_tags()` (*lawwenda.Share property*), 30
`HideNodesFilesystemAdapter` (class in *lawwenda.fs*), 23

I

`icon_name()` (*lawwenda.fs.Filesystem.Node property*), 16
`include_by_patterns()` (*lawwenda.Share property*), 31
`include_by_tags()` (*lawwenda.Share property*), 31
`is_active()` (*lawwenda.server.Server._ShareAppInst property*), 26
`is_active()` (*lawwenda.Share property*), 31
`is_dir()` (*lawwenda.fs.Filesystem method*), 20
`is_dir()` (*lawwenda.fs.Filesystem.Node property*), 16
`is_dir()` (*lawwenda.fs.LocalFilesystem method*), 23
`is_file()` (*lawwenda.fs.Filesystem method*), 20
`is_file()` (*lawwenda.fs.Filesystem.Node property*), 16
`is_file()` (*lawwenda.fs.LocalFilesystem method*), 23
`is_hidden()` (*lawwenda.fs.Filesystem method*), 20
`is_hidden()` (*lawwenda.fs.Filesystem.Node property*), 16
`is_hidden()` (*lawwenda.fs.HideNodesFilesystemAdapter method*), 23
`is_hidden()` (*lawwenda.fs.LocalFilesystem method*), 23
`is_link()` (*lawwenda.fs.Filesystem method*), 20
`is_link()` (*lawwenda.fs.Filesystem.Node property*), 17
`is_link()` (*lawwenda.fs.LocalFilesystem method*), 23

`is_previewable()` (*lawwenda.preview.commonimages.CommonImagesPreviewer method*), 3
`is_previewable()` (*lawwenda.preview.commonvideos.CommonVideosPreviewer method*), 4
`is_previewable()` (*lawwenda.preview.Previewer method*), 5
`is_thumbnailable()` (*lawwenda.preview.commonimages.CommonImagesPreviewer method*), 3
`is_thumbnailable()` (*lawwenda.preview.commonvideos.CommonVideosPreviewer method*), 4
`is_thumbnailable()` (*lawwenda.preview.Previewer method*), 5
`is_writable()` (*lawwenda.fs.Filesystem.Node property*), 17
`is_writable()` (*lawwenda.sharehttpapp.davprop.DavProp property*), 8
`IsAbleRequest` (class in *lawwenda.preview*), 4

J

`JsonedResponse` (class in *lawwenda.comm*), 11

K

`known_tags()` (*lawwenda.fs.Filesystem method*), 20
`known_tags()` (*lawwenda.fs.LocalFilesystem method*), 23

L

`lawwenda` module, 27
`lawwenda.comm` module, 11
`lawwenda.datafiles` module, 11
`lawwenda.devserver` module, 11
`lawwenda.fmapp` module, 12
`lawwenda.fs` module, 14
`lawwenda.lawwenda_cli` module, 25
`lawwenda.preview` module, 4
`lawwenda.preview.commonimages` module, 3
`lawwenda.preview.commonvideos` module, 4
`lawwenda.search` module, 6
`lawwenda.server` module, 26

lawwenda.sharehttpapp
 module, 9
 lawwenda.sharehttpapp.davprop
 module, 8
 list_shares() (in module lawwenda.lawwenda_cli),
 26
 LocalFilesystem (class in lawwenda.fs), 23

M

main() (in module lawwenda.lawwenda_cli), 26
 mimetype (lawwenda.previewer.IsAbleRequest attribute), 4
 mimetype() (lawwenda.fs.Filesystem method), 20
 mimetype() (lawwenda.fs.Filesystem.Node property),
 17
 mimetype() (lawwenda.fs.LocalFilesystem method),
 23
 mkdir() (lawwenda.fs.Filesystem method), 20
 mkdir() (lawwenda.fs.Filesystem.Node method), 17
 mkdir() (lawwenda.fs.LocalFilesystem method), 23
 module
 lawwenda, 27
 lawwenda.comm, 11
 lawwenda.datafiles, 11
 lawwenda.devserver, 11
 lawwenda.fmapp, 12
 lawwenda.fs, 14
 lawwenda.lawwenda_cli, 25
 lawwenda.previewer, 4
 lawwenda.previewer.commonimages, 3
 lawwenda.previewer.commonvideos, 4
 lawwenda.search, 6
 lawwenda.server, 26
 lawwenda.sharehttpapp, 9
 lawwenda.sharehttpapp.davprop, 8
 move_to() (lawwenda.fs.Filesystem method), 21
 move_to() (lawwenda.fs.Filesystem.Node method), 17
 move_to() (lawwenda.fs.LocalFilesystem method), 23
 mtime() (lawwenda.fs.Filesystem method), 21
 mtime() (lawwenda.fs.Filesystem.Node property), 17
 mtime() (lawwenda.fs.LocalFilesystem method), 23
 mtime_ts() (lawwenda.fs.Filesystem.Node property),
 17
 MtimeDavProp (class in
 lawwenda.sharehttpapp.davprop), 8

N

name (lawwenda.previewer.IsAbleRequest attribute), 4
 name() (lawwenda.fs.Filesystem.Node property), 17
 name() (lawwenda.Share property), 31
 node_by_path() (lawwenda.fs.Filesystem method),
 21

P

p_descending() (lawwenda.fs._PredicateFactory
 static method), 24
 p_not() (lawwenda.fs._PredicateFactory static
 method), 24
 p_or() (lawwenda.fs._PredicateFactory static method),
 24
 p_regexp() (lawwenda.fs._PredicateFactory static
 method), 24
 p_tag() (lawwenda.fs._PredicateFactory static
 method), 24
 parent_node() (lawwenda.fs.Filesystem.Node prop-
 erty), 17
 password_salt() (lawwenda.Share property), 31
 password_scrypt() (lawwenda.Share property), 32
 path() (lawwenda.Configuration property), 28
 path() (lawwenda.fs.Filesystem.Node property), 17
 path() (lawwenda.Share property), 32
 peek_share_cache_tag()
 (lawwenda.Configuration method), 28
 preview_html() (lawwenda.fs.Filesystem method),
 21
 preview_html() (lawwenda.fs.Filesystem.Node
 property), 17
 preview_html() (lawwenda.fs.LocalFilesystem
 method), 23
 preview_html() (lawwenda.previewer.commonimages.CommonImages
 method), 3
 preview_html() (lawwenda.previewer.commonvideos.CommonVideosP
 method), 4
 preview_html() (lawwenda.previewer.Previewer
 method), 5
 Previewer (class in lawwenda.previewer), 4

Q

query() (lawwenda.search.ByGeoSearch method), 6
 query() (lawwenda.search.ByNameSearch method), 6
 query() (lawwenda.search.ByTagSearch method), 6
 query() (lawwenda.search.DeeplySearch method), 7
 query() (lawwenda.search.Search method), 7

R

rating() (lawwenda.fs.Filesystem method), 21
 rating() (lawwenda.fs.Filesystem.Node property), 18
 rating() (lawwenda.fs.LocalFilesystem method), 23
 RatingDavProp (class in
 lawwenda.sharehttpapp.davprop), 8
 read_file() (lawwenda.fs.Filesystem method), 21
 read_file() (lawwenda.fs.Filesystem.Node method),
 18
 read_file() (lawwenda.fs.LocalFilesystem method),
 23
 readonly() (lawwenda.Share property), 32

ReadOnlyFilesystemAdapter (class in lawwenda.fs), 24
register_prop() (in module lawwenda.sharehttpapp.davprop), 9
remove_share() (in module lawwenda.lawwenda_cli), 26
remove_share() (lawwenda.Configuration method), 28
remove_tag() (lawwenda.fs.Filesystem method), 21
remove_tag() (lawwenda.fs.Filesystem.Node method), 18
remove_tag() (lawwenda.fs.LocalFilesystem method), 24
Request (class in lawwenda.comm), 11
ResourceTypeDavProp (class in lawwenda.sharehttpapp.davprop), 8
rootnode() (lawwenda.fs.Filesystem property), 21
run() (lawwenda.devserver._DevServerThread method), 12
run_dev_server() (in module lawwenda.devserver), 12
run_dev_server() (in module lawwenda.lawwenda_cli), 26
run_dev_server() (lawwenda.Configuration method), 28

S

sanitize_abspath() (lawwenda.fs.Filesystem static method), 21
Search (class in lawwenda.search), 7
Server (class in lawwenda.server), 26
Server._ShareAppInst (class in lawwenda.server), 26
set_comment() (lawwenda.fs.Filesystem method), 22
set_comment() (lawwenda.fs.Filesystem.Node method), 18
set_comment() (lawwenda.fs.LocalFilesystem method), 24
set_geo() (lawwenda.fs.Filesystem method), 22
set_geo() (lawwenda.fs.Filesystem.Node method), 18
set_geo() (lawwenda.fs.LocalFilesystem method), 24
set_rating() (lawwenda.fs.Filesystem method), 22
set_rating() (lawwenda.fs.Filesystem.Node method), 18
set_rating() (lawwenda.fs.LocalFilesystem method), 24
Share (class in lawwenda), 28
ShareHttpApp (class in lawwenda.sharehttpapp), 9
shutdown() (lawwenda.devserver._DevServerInfo method), 11
size() (lawwenda.fs.Filesystem method), 22
size() (lawwenda.fs.Filesystem.Node property), 18
size() (lawwenda.fs.LocalFilesystem method), 24

SizeDavProp (class in lawwenda.sharehttpapp.davprop), 8

T

tags() (lawwenda.fs.Filesystem method), 22
tags() (lawwenda.fs.Filesystem.Node property), 18
tags() (lawwenda.fs.LocalFilesystem method), 24
TagsDavProp (class in lawwenda.sharehttpapp.davprop), 9
tagstring() (lawwenda.fs.Filesystem.Node property), 18
thumbheight (lawwenda.preview.Previewer attribute), 5
thumbnail() (lawwenda.fs.Filesystem method), 22
thumbnail() (lawwenda.fs.Filesystem.Node method), 18
thumbnail() (lawwenda.fs.LocalFilesystem method), 24
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), 32
traverse_dir() (lawwenda.fs.Filesystem.Node method), 18
try_get_fullpath() (lawwenda.fs.Filesystem method), 22
try_get_fullpath() (lawwenda.fs.Filesystem.Node method), 18
try_get_fullpath() (lawwenda.fs.LocalFilesystem method), 24

U

unescape() (lawwenda.fmapp._RenderTemplateValue property), 14
url() (lawwenda.devserver._DevServerInfo property), 11
url_internals_name (lawwenda.fmapp.FmApp attribute), 13

W

wait_stopped() (lawwenda.devserver._DevServerInfo method), 11
write_file() (lawwenda.fs.Filesystem method), 22
write_file() (lawwenda.fs.Filesystem.Node method), 19

`write_file()` (*lawwenda.fs.LocalFilesystem*
method), 24