

AjaXplorer Driver API v2.7.X => 3.0

Commons

Common Parameters

get_action

[required] Service differentiation. Basically, any service will be called by something like :
http://host/ajaxplorer/content.php?get_action=action¶meters...

dir

Current location inside the repository, beginning with « / ».
Example : dir=/my/folder, dir=/

file, file_[index]

In many cases, we need to pass on or more selected files. If selection is unique, pass only the « file » parameter, else parameters are indexed automatically : file_0, file_1, file2, etc. Values are the full path of the file inside the repository (not only the basename of the file, even if the current location is passed via the dir parameter).
Example : ?get_action=toto&file_0=/my/folder/file1.txt&file_1=/my/folder/file2.html& etc...

Common Returns

Messages

When not explicited, all services return XML values. The root element is <tree>. Exceptions are converted to a <message type=« ERROR »></message> element, where as the success messages are <message type=« SUCCESS »></message>.

Example :

```
<?xml version="1.0" encoding="UTF-8"?>
<tree>
<message type="ERROR">
Impossible de trouver le fichier /admin/aaa78/20071207-eFlyCabaret+Recto.gif in
E:\charlie\software\[...]\class.fsAccessDriver.php (1.206)
</message>
</tree>
```

Reload Instructions

In many cases, the data manipulation caused by the services may require a reloading of the current listing. The <reload_instruction> tag will fire such a reloading :

```
<reload_instruction
  object=« data|repository_list » [required, generally «data»]
  node = « path to node to reload » [optional]
  file = « file to select after reload » [optional]
/>
```

Example : after moving a file, we need to both trigger a reload of the current folder, and a reload of the destination folder

```
<?xml version="1.0" encoding="UTF-8"?>
<tree>
  <reload_instruction object=« data »/>
  <reload_instruction object=« data » node=«/path/to/destination/folder»/>
</tree>
```

Basic implementation for a driver : « ls » listing action.

ls

Lists all data children inside a given node

- dir : cf commons
- Pagination : page can be appended to the dir value after a #
- options : [l, d, z, f, a]
 - l : list metadata
 - d : include directories in listing
 - z : include archive files in listing
 - f : include files in listing
 - a : « all », equivalent to dzf

Returns

If « l » is not set, minimal attributes are **text**, **icon**, and **filename**. For a folder, you can add **openicon** and you must add **src**.

```
<?xml version="1.0" encoding="UTF-8"?>
<tree filename="/current/path" is_file="false">
  <tree
    text="Musique"
    icon="folder.png"
    openicon="folder_open.png"
    filename="/current/path/Musique"
    src="content.php?dir=%2Fcurrent%2Fpath%2FMusique"
  />
</tree>
```

If « l » is set, then we must send the listing with available metadata :

```
<?xml version="1.0" encoding="UTF-8"?>

<tree filename="/current/path" is_file="false">

  <!-- A FOLDER -->
  <tree
    text="tt"
    is_file="0"
    is_image="0"
    mimestring="Directory"
    modiftime="17/12/2007 19:18"
    filesize="-"
    filename="/current/path/tt"
    icon="folder.png"
  />

  <!-- A FILE -->
  <tree
    text="mydoc.txt"
    is_file="1"
    is_image="0"
    mimestring="Text File"
    ajxp_modiftime="1111111" // LINUX TIMESTAMP
    filesize="119.63 Kb" // READABLE SIZE
    bytesize="124515" // COMPUTER SIZE
    filename="/current/path/mydoc.txt"
    icon="text2.png"
  />

  <!-- AN IMAGE FILE -->
  <tree
```

```

        text="20071207-eFlyCabaret-Recto.gif"
        is_file="1"
        is_image="1"
        <!-- IMAGE SPECIFIC ATTRIBUTES (needed only if is_image=1) -->
        image_type="image/gif"
        image_width="344"
        image_height="485"
        <!-- END OF IMAGE ATTRIBUTES -->
        mimestring="GIF picture"
        ajxp_modiftime="1111111" // LINUX TIMESTAMP
        filesize="119.63 Kb" // READABLE SIZE
        bytesize="124515" // COMPUTER SIZE
        filename="/current/path/20071207-eFlyCabaret-Recto.gif"
        icon="image.png"
    />
</tree>

```

See the file server/conf/conf.extensions.php for the supported extensions/icons.

File System Access actions

--- BASIC GET/SET

upload

Post a file, either via the standard HTML uploader or via the Flash uploader

HTML Uploader

- dir : cf. commons
- userfile_[index] : 0 to n \$_FILES data

Flash Uploader

- dir : base64_encoded dir location, call base64_decode before use
- Filedata : unique \$_FILES data.

download

Trigger a browser download window. If multiple files selected and zip is enabled on server, creates an archive then triggers download

- file selection : cf commons

get_content

Reads a file and send it as plain text (used by editors)

- file : unique file selection , cf commons

put_content

Post a file content and save it inside the file

- file : unique file selection , cf commons
- content : the content to save inside the file
- encode : [optional] if set to « base64 », the data is probably binary and is base64_encoded.

stat

Retrieve basic data concerning a filesystem node : file or folder, permissions and size if it's a file.

- file : unique file selection , cf commons
- format : JSON representation of the array returned by <http://www.php.net/manual/en/function.stat.php>

---- FILE MANIPULATION

copy / move

Copy or move selection

- dir : cf. commons
- file selection : cf commons
- dest : the destination node

delete

Permanently delete the selection

- dir : cf. commons
- file selection : cf commons

rename

Rename selected file

- dir : cf commons
- file : cf commons

mkdir

Creates a folder

- dir : cf commons
- dirname : name of the new folder

mkfile

Creates an empty file. In the basic driver, if the file extension is « .html », an empty HTML skeleton is written inside the file.

- dir : cf. commons
- filename : new file

chmod

Change permission on file

- file selection : cf. commons
- chmod_value : the value for permission, like 0777, 0644, etc.
- recursive : [« on », « off »] wether this should be applied recursively to children.
- recur_apply_to : [« file », « dir », « both »] If recursive, on which items to apply.

---- « BUSINESS » ORIENTED

compress

If zip is enabled, creates a ZIP archive

- file selection : cf commons
- archive_name : [optional] Name for the newly created archive

public_url

Create a public link

- file : cf commons, unique selection
- password : a password to ask before allowing download
- expiration : When this public link expires.