FileBrowser Documentation

Release 8.1.0

LfPHP Services

May 06, 2023

CONTENTS

FileBrowser

FileBrowser Home Page

WHAT IS FILEBROWSER?

FileBrowser is a free, open-source, self-hosted web application for managing files and folders.

You can manage files inside your local repository folder (on your server's hard drive) or connect to other storage adapters (see below).

FileBrowser has multi-user support, so you can have administrators and other users managing their files with different access permissions, roles and home folders.

All basic file operations are supported: copy, move, rename, create, delete, zip, unzip, download, upload.

If allowed, users can download multiple files or folders at once.

File upload supports drag&drop, progress bar, pause and resume. Upload is chunked so you should be able to upload large files regardless of your server's configuration.

1.1 Features & Goals

- Multiple storage adapters (Local, FTP, Amazon S3, Dropbox, DO Spaces, Azure Blob and many others via Flysystem),

- Multiple authentication adapters with roles and permissions (store users in a json file, database, or use Word-Press),

- Multiple session adapters (native file, Pdo, Redis, MongoDB, Memcached, and others via Symfony),

- Single page front-end (built with Vue.js, Bulma and Buefy),

- Chunked uploads (built with Resumable.js),

- Zip and bulk download support,

- Highly extensible, decoupled and tested code,

- No database required.

1.2 Why Open Source on GitHub?

There are several reasons why we switched to the open source model.

Basically, we wanted to increase:

- Code quality by bringing more developers to the project,
- Code stability,
- Security,
- Project lifetime.

At the end, the more people can see and test some code, the more likely any flaws will be caught and fixed quickly.

1.3 Show Your Support

- Please star this repository on GitHub if this project helped you!

1.4 License

Copyright 2021, Foreach Code Factory. Copyright 2018-2021, Milos Stojanovic.

This project is Apache-2.0 licensed.

INSTALLATION

2.1 Minimum Requirements

- FileBrowser 8.1.0: PHP 8.0 - 8.2 (with php-zip extension)
- FileBrowser 8.0.3: PHP 7.2 - 7.4 (with php-zip extension)

2.2 Download a Pre-Compiled Build

The pre-compiled builds are created for non-developers. With this version of the FileBrowser, the front end code (HTML, CSS and Javascript) is already pre-compiled for you, and the source code is removed, so that the final archive file contains only what is required to run the application on your server.

- Download the latest release,
- Unzip the files, and upload them to your PHP server,
- Make sure your web server can read and write to the `filebrowser/repository/` and `filebrowser/private/` folders,
- Set the website document root to the `filebrowser/dist/` directory (this is also known as 'public' folder),
- Visit the web page, and if something goes wrong, please check `filebrowser/private/logs/app.log`,
- Login with default credentials `admin/admin123`,
- Change default admin's password.

NOTE: For security reasons, the `/dist` folder is the ONLY folder you want to be exposed to the Web. Everything else should be outside of your web root. This way, people won't be able to access any of your important files through the Web browser.

2.2.1 Install on fresh Ubuntu 18.04 or Debian 10.3

On a new server, login as root and enter this into the shell:

```
apt update
apt install -y wget unzip php apache2 libapache2-mod-php php-zip

cd /var/www/
wget -O filebrowser_latest.zip https://filebrowser.linuxforphp.net/files/filebrowser_
→latest.zip
```

(continues on next page)

```
unzip filebrowser_latest.zip && rm filebrowser_latest.zip

chown -R www-data:www-data filebrowser/
chmod -R 775 filebrowser/

echo "
<VirtualHost *:80>
    DocumentRoot /var/www/filebrowser/dist
</VirtualHost>
" >> /etc/apache2/sites-available/filebrowser.conf

a2dissite 000-default.conf
a2ensite filebrowser.conf
systemctl restart apache2

exit
```

Open your browser and go to http://your_server_ip_address

2.3 Upgrade

Since version 7 is completely rewritten from scratch, there is no clear upgrade path from older versions.

If you have an older version of FileBrowser please backup everything and install the script again.

Upgrade instructions for non-developers:

- Backup everything,

- Download the latest version,

- Replace all files and folders except `repository/` and `private/`.

To discover which version of the FileBrowser you are running, please look for `APP_VERSION` inside `dist/index.php` file

ACCOUNTS

3.1 User Roles

There are three different user roles:

- Admin (for user and file management),
- User (regular, logged in user),
- Guest (anonymous, not logged in).

3.2 User Permissions

There are 6 different user permissions that the administrator can assign to each user:

- Read (user can browse and list files and folders),
- Write (user can copy, move, rename, and delete files),
- Upload (user can upload files to the repository),
- Download (user can download files from the repository),
- Batch Download (user can download multiple files and folders at once),
- Zip (user can zip and unzip files).

Some permissions require others. For example, Batch Download requires Read permissions (so that the user can list files and select them), as well as basic Download permissions.

3.2.1 Guest Account

The Guest account is a predefined account, and it is disabled by default since no permissions are assigned to it.

The Admin can enable the Guest account, which will allow anyone to interact with the repository based on the Guest account permissions.

3.3 Resetting Admin's Password

If you forgot your admin password you can follow these steps to reset it:

- Backup your current users file `private/users.json` to a safe place,

- Copy blank template `private/users.json.blank` over `private/users.json` or simply refresh your browser,

- Login as admin with default credentials `admin/admin123`,

- Put your original users file back to `private/users.json` replacing the template,

- Since you are now logged in as admin, simply go to users page and change your password,

- Log out and try to login with the new password.

Note: If you're using database Auth adapter then simply run this query to set default password back to `admin123`

```
UPDATE `users`
SET `password` = '$2y$10$Nu35w4pteLfc7BDCIkDPkecjw8wsH8Y2GMfIewUbXLT7zzW6WOxwq'
WHERE `username` = 'admin';
```

CONFIGURATION

4.1 Basic configuration

All services are set with fairly conservative defaults. For regular users, there is no need to change anything. The application should work out of the box.

You can edit the `configuration.php` file in order to customize the application, by changing things like the logo, the title, the language settings, and the upload permissions.

All other configuration files can be found in the `config/` folder.

Note: If you make a mistake in any of the configuration files (forgot to close a quote?), the application will fail to load, and log an error. Please use the provided default `configuration_sample.php` file to restore the configuration to its initial state.

```php
'frontend_config' => [
    'app_name' => 'FileBrowser',
    'app_version' => APP_VERSION,
    'language' => 'english',
    'logo' => 'https://linuxforphp.com/img/logo.svg',
    'upload_max_size' => 100 * 1024 * 1024, // 100MB
    'upload_chunk_size' => 1 * 1024 * 1024, // 1MB
    'upload_simultaneous' => 3,
    'default_archive_name' => 'archive.zip',
    'editable' => ['.txt', '.css', '.js', '.ts', '.html', '.php', '.json', '.ini', '.cnf
↪', '.conf', '.env', '.monthly', '.weekly', '.daily', '.hourly', '.minute', '.htaccess
↪'],
    'date_format' => 'YY/MM/DD hh:mm:ss', // see: https://momentjs.com/docs/#/
↪displaying/format/
    'guest_redirection' => '', // useful for external auth adapters
    'search_simultaneous' => 5,
    'filter_entries' => [],
],
```

4.2 Adding Custom HTML

You can add additional html to the head and body like this:

```
'Filebrowser\Services\View\ViewInterface' => [
    'handler' => '\Filebrowser\Services\View\Adapters\Vuejs',
    'config' => [
        'add_to_head' => '<meta name="author" content="something">',
        'add_to_body' => '<script src="http://example.com/analytics.js"></script>',
    ],
],
```

4.3 Tweaking the Look

To change default color scheme and other options, edit `frontend/App.vue` When you're done, recompile with `npm run build` like described in the *Project Setup for Development (Linux)* section.

```
// Primary color
$primary: #34B891;
$primary-invert: findColorInvert($primary);

$colors: (
    "primary": ($primary, $primary-invert),
    "info": ($info, $info-invert),
    "success": ($success, $success-invert),
    "warning": ($warning, $warning-invert),
    "danger": ($danger, $danger-invert),
);

// Links
$link: $primary;
$link-invert: $primary-invert;
$link-focus-border: $primary;

// Disable the widescreen breakpoint
$widescreen-enabled: false;

// Disable the fullhd breakpoint
$fullhd-enabled: false;
```

AUTHENTICATION

5.1 Default Auth Service

By default, user credentials are stored in a json file. For some use cases, this is enough. It also makes this application lightweight since no database is required.

The default handler only accepts a filename parameter. This file should be writable by the web server.

```
'Filebrowser\Services\Auth\AuthInterface' => [
    'handler' => '\Filebrowser\Services\Auth\Adapters\JsonFile',
    'config' => [
        'file' => __DIR__.'/private/users.json',
    ],
],
```

5.2 Configuring Auth Service to Use A Database

You can also use a MySQL database to store your users.

First, create a table `users` with this sql query:

```
CREATE TABLE `users` (
    `id` int(10) NOT NULL AUTO_INCREMENT,
    `username` varchar(255) NOT NULL,
    `name` varchar(255) NOT NULL,
    `role` varchar(20) NOT NULL,
    `permissions` varchar(200) NOT NULL,
    `homedir` varchar(2000) NOT NULL,
    `password` varchar(255) NOT NULL,
    PRIMARY KEY (`id`),
    KEY `username` (`username`)
) CHARSET=utf8 COLLATE=utf8_bin;
```

Then, import default users with sql query:

```
INSERT INTO `users` (`username`, `name`, `role`, `permissions`, `homedir`, `password`)
VALUES
('guest', 'Guest', 'guest', '', '/', ''),
('admin', 'Admin', 'admin', 'read|write|upload|download|batchdownload|zip', '/', '$2y$10
↪$Nu35w4pteLfc7BDCIkDPkecjw8wsH8Y2GMfIewUbXLT7zzW6WOxwq');
```

At the end, open `configuration.php`, and update the AuthInterface handler with your database settings:

```
'Filebrowser\Services\Auth\AuthInterface' => [
    'handler' => '\Filebrowser\Services\Auth\Adapters\Database',
    'config' => [
        'driver' => 'mysqli',
        'host' => 'localhost',
        'username' => 'root',
        'password' => 'password',
        'database' => 'filebrowser',
    ],
],
```

5.3 Configuring Auth Service to Use WordPress

Replace your current Auth handler in `configuration.php` file, like so:

```
'Filebrowser\Services\Auth\AuthInterface' => [
    'handler' => '\Filebrowser\Services\Auth\Adapters\WPAuth',
    'config' => [
        'wp_dir' => '/var/www/my_wordpress_site/',
        'permissions' => ['read', 'write', 'upload', 'download', 'batchdownload', 'zip'],
        'private_repos' => false,
    ],
],
```

You can then adjust the following configuration elements:

- `wp_dir` should be the directory path of your wordpress installation,

- `permissions` is the array of permissions given to each user,

- `private_repos` must be set to true or false, in order to allow, or not, each user to have his own home folder.

With more recent versions of the FileBrowser you can set `guest_redirection` in your `configuration.php` to redirect logged out users back to your WordPress site, like so:

```
'frontend_config' => [
 ...
    'guest_redirection' => 'http://example.com/wp-admin/',
 ...
]
```

5.4 Configuring Auth Service to Use LDAP

Replace your current Auth handler in `configuration.php` file, like so:

```
'Filebrowser\Services\Auth\AuthInterface' => [
    'handler' => '\Filebrowser\Services\Auth\Adapters\LDAP',
    'config' => [
            'private_repos' => false,
            'ldap_server'=>'ldap://192.168.1.1',
```

```
            'ldap_bindDN'=>'uid=ldapbinduser,cn=users,dc=ldap,dc=example,dc=com',
            'ldap_bindPass'=>'ldapbinduser-password',
            'ldap_baseDN'=>'cn=users,dc=ldap,dc=example,dc=com',
            'ldap_filter'=>'(uid=*)', //ex: 'ldap_filter'=>'(&
→(uid=*)(memberOf=cn=administrators,cn=groups,dc=ldap,dc=example,dc=com))',
            'ldap_attributes' => ["uid","cn","dn"],
            'ldap_userFieldMapping'=> [
                'username' =>'uid',
                'name' =>'cn',
                'userDN' =>'dn',
                'default_permissions' => 'read|write|upload|download|batchdownload|zip',
                'admin_usernames' =>['user1', 'user2'],
            ],
        ],
],
```

5.5 Custom Authentication Using Third-Party Software

If you want to use FileBrowser as a part of another application, you probably already have users stored somewhere else. What you need in this case is to build a new custom Auth adapter that matches the AuthInterface to connect the applications together. This new adapter will allow the FileBrowser tp try to authenticate users with the other application, and then translate each new user into one of its User objects.

5.6 API Authentication

When using the Authentication API, FileBrowser's front end application will use session-based authentication to authenticate users, and interact with the FileBrowser's back end handlers.

Note: The application will not work if you disable cookies.

LOGGING

6.1 Configuring the Logging Service

Logging is provided trough the powerful Monolog library. Please read their documentation for more details: https://github.com/Seldaek/monolog.

The default handler will simply use the `private/logs/app.log` file to store application logs and errors.

```
'Filebrowser\Services\Logger\LoggerInterface' => [
    'handler' => '\Filebrowser\Services\Logger\Adapters\MonoLogger',
    'config' => [
        'monolog_handlers' => [
            function () {
                return new \Monolog\Handler\StreamHandler(
                    __DIR__.'/private/logs/app.log',
                    \Monolog\Logger::DEBUG
                );
            },
        ],
    ],
],
```

There are many different handlers you can add on top of this stack (monolog_handlers array). Some of them are listed here.

ROUTER

7.1 Router Service

The Router service is the well-known FastRoute library. There is no need to change this service unless you're extending the script.

The router uses the unique query parameter `?r=` to pass route information to other application components. Because of this feature, this (single-page) application does not require rewrite rules in the .htaccess file, or similar modifications.

Example routes:

- `http://example.com/?r=/some/route&param1=val1&param2=val2`

- `http://example.com/?r=/user/{user_id}&param1=val1`

7.2 Routes File

The Routes files are located in the `config/` folder. There are two main files that will configure routes: `config/routes.config.php` and `config/routes.optional.config.php`. Each route is defined like so:

```
[
    'route' => [
        'GET', '/download/{path_encoded}', '\Filebrowser\Controllers\
→DownloadController@download',
    ],
    'roles' => [
        'guest', 'user', 'admin',
    ],
    'permissions' => [
        'download',
    ],
],
```

As you can see in this example, you can assign required user roles and permissions for each route.

7.3 Controllers

Since FileBrowser is using an awesome dependency injection container, you can type hint dependencies directly in the definition of a controller's action methods.

You can also mix route parameters and dependencies in any order, like in this example:

```
public function __construct(Config $config, Session $session, AuthInterface $auth,␣
→Filesystem $storage)
{
 // ...
}

public function download($path_encoded, Request $request, Response $response,␣
→StreamedResponse $streamedResponse)
{
 // ...
}
```

SECURITY

8.1 Configuring the Security Service

A simple security service is included in the application by default. This service provides:

- A basic session-based CSRF protection,

- An IP allow list,

- An IP deny list.

```
'Filebrowser\Services\Security\Security' => [
    'handler' => '\Filebrowser\Services\Security\Security',
    'config' => [
        'csrf_protection' => true,
        'csrf_key' => "123456", // randomize this
        'ip_allowlist' => [],
        'ip_denylist' => [
            '172.16.1.2',
            '172.16.3.4',
        ],
    ],
],
```

If you set the `ip_allowlist` option, then only users coming from the listed IP addresses will be able to use the application.

SESSIONS

9.1 Default Session Handler

Session handling is provided through Symfony's HttpFoundation component. Please check their documentation for more details: https://symfony.com/doc/4.4/components/http_foundation.html.

The default session handler will use PHP's built-in file storage. You can also specify your own `$save_path` to store the session files.

```
'Filebrowser\Services\Session\SessionStorageInterface' => [
   'handler' => '\Filebrowser\Services\Session\Adapters\SessionStorage',
   'config' => [
       'handler' => function () {
           $save_path = null; // use default system path
           //$save_path = __DIR__.'/private/sessions';
           $handler = new \Symfony\Component\HttpFoundation\Session\Storage\Handler\
→NativeFileSessionHandler($save_path);

           return new \Symfony\Component\HttpFoundation\Session\Storage\
→NativeSessionStorage([], $handler);
       },
   ],
],
```

9.2 Configuring the Session Service to Use a Database

First, create a table `sessions` with this SQL:

```
CREATE TABLE `sessions` (
    `sess_id` varbinary(128) NOT NULL,
    `sess_data` blob NOT NULL,
    `sess_lifetime` mediumint(9) NOT NULL,
    `sess_time` int(10) unsigned NOT NULL,
    PRIMARY KEY (`sess_id`)
) CHARSET=utf8 COLLATE=utf8_bin;
```

Then, open `configuration.php`, and update the Session handler to:

```
'Filebrowser\Services\Session\SessionStorageInterface' => [
   'handler' => '\Filebrowser\Services\Session\Adapters\SessionStorage',
   'config' => [
       'handler' => function () {
           $handler = new \Symfony\Component\HttpFoundation\Session\Storage\Handler\
→PdoSessionHandler(
                   'mysql://root:password@localhost:3306/filebrowser'
                   );

           return new \Symfony\Component\HttpFoundation\Session\Storage\
→NativeSessionStorage([], $handler);
       },
   ],
],
```

Don't forget to enter the correct database credentials.

9.3 Configuring Session Service to Use Redis

You must require the additional predis library `composer require predis/predis` in order to use this session handler. Then, please modify the `configuration.php` file, like so:

```
'Filebrowser\Services\Session\SessionStorageInterface' => [
   'handler' => '\Filebrowser\Services\Session\Adapters\SessionStorage',
   'config' => [
       'handler' => function () {
           $predis = new \Predis\Client('tcp://127.0.0.1:6379');
           $handler = new \Symfony\Component\HttpFoundation\Session\Storage\Handler\
→RedisSessionHandler($predis);

           return new \Symfony\Component\HttpFoundation\Session\Storage\
→NativeSessionStorage([], $handler);
       },
   ],
],
```

9.4 Session Options

The underlying session component accepts array of options. For example, you can pass the `cookie_lifetime` parameter to extend the default session lifetime:

```
'Filebrowser\Services\Session\SessionStorageInterface' => [
   'handler' => '\Filebrowser\Services\Session\Adapters\SessionStorage',
   'config' => [
       'handler' => function () {
           $handler = new \Symfony\Component\HttpFoundation\Session\Storage\Handler\
→PdoSessionHandler(
                   'mysql://root:password@localhost:3306/filebrowser'
                   );
```

```
        return new \Symfony\Component\HttpFoundation\Session\Storage\
→NativeSessionStorage([
                'cookie_lifetime' => 365 * 24 * 60 * 60, // one year
        ], $handler);
    },
  ],
],
```

STORAGE

All of the basic configuration for storage adapters is done in the `configuration.php` file.

10.1 Adapters

Different storage adapters are provided through the versatile Flysystem library.

You can use the local filesystem (default), FTP, SFTP, Amazon S3, DigitalOcean Spaces, Microsoft Azure Blob, Dropbox, and many others.

Please check the Flysystem documentation for the exact setup that is required for each adapter.

Note: Some adapters do not support, or only support in a very limited way, some folder operations.

10.2 Default Local Disk Adapter

For the default adapter, you only need to configure the location of your `repository` folder in the `configuration.php` file. This folder will then serve as the root folder of all the files and folders of the repository.

If you wish to use the `zip` and `unzip` binaries directly, without using the Flysystem per se, you can set the `fastzip` option to `true` (default). Zipping with this option is not only faster, but will also manage any Unix special files (symlinks, sparse files, etc.) appropriately.

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [],
            'adapter' => function () {
                return new \League\Flysystem\Adapter\Local(
                    REPOSITORY_ROOT
                );
            },
        ],
        'fastzip' => true // LOCAL ONLY! If true, it will override the \League\Flysystem\
→Adapter\Local adapter, and use the zip and unzip binaries directly.
    ],
],
```

10.3 FTP Adapter

Please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [],
            'adapter' => function () {
              return new \League\Flysystem\Adapter\Ftp([
                    'host' => 'example.com',
                    'username' => 'demo',
                    'password' => 'password',
                    'port' => 21,
                    'timeout' => 10,
                ]);
            },
        ],
    ],
],
```

10.4 SFTP Adapter

You must require an additional library in order to use this adapter:

```
composer require league/flysystem-sftp
```

For more advanced options like using your private key or changing the document root, please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [],
            'adapter' => function () {
              return new \League\Flysystem\Sftp\SftpAdapter([
                    'host' => 'example.com',
                    'port' => 22,
                    'username' => 'demo',
                    'password' => 'password',
                    'timeout' => 10,
                ]);
            },
        ],
    ],
],
```

10.5 Dropbox Adapter

You must require an additional library in order to use this adapter:

```
composer require spatie/flysystem-dropbox
```

Please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [
                'case_sensitive' => false,
            ],
            'adapter' => function () {
              $authorizationToken = '1234';
              $client = new \Spatie\Dropbox\Client($authorizationToken);

              return new \Spatie\FlysystemDropbox\DropboxAdapter($client);
            },
        ],
    ],
],
```

10.6 Amazon S3 Adapter (v3)

You must require an additional library in order to use this adapter:

```
composer require league/flysystem-aws-s3-v3
```

Please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [],
            'adapter' => function () {
                $client = new \Aws\S3\S3Client([
                    'credentials' => [
                        'key' => '123456',
                        'secret' => 'secret123456',
                    ],
                    'region' => 'us-east-1',
                    'version' => 'latest',
                ]);
```

(continued from previous page)

```
            return new \League\Flysystem\AwsS3v3\AwsS3Adapter($client, 'my-bucket-name
→');
        },
    ],
    ],
],
```

10.7 DigitalOcean Spaces

You must require an additional library in order to use this adapter:

```
composer require league/flysystem-aws-s3-v3
```

The DigitalOcean Spaces API are compatible with those of S3.

Please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [],
            'adapter' => function () {
                $client = new \Aws\S3\S3Client([
                    'credentials' => [
                        'key' => '123456',
                        'secret' => 'secret123456',
                    ],
                    'region' => 'us-east-1',
                    'version' => 'latest',
                    'endpoint' => 'https://nyc3.digitaloceanspaces.com',
                ]);

                return new \League\Flysystem\AwsS3v3\AwsS3Adapter($client, 'my-bucket-name
→');
            },
        ],
    ],
],
```

10.8 Microsoft Azure Blob Storage

You must require an additional library in order to use this adapter:

```
composer require league/flysystem-azure-blob-storage
```

Please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [],
            'adapter' => function () {
                $accountName = 'your_storage_account_name';
                $accountKey = '123456';
                $containerName = 'my_container';

                $client = \MicrosoftAzure\Storage\Blob\BlobRestProxy::createBlobService(
                    "DefaultEndpointsProtocol=https;AccountName=${accountName};AccountKey=
→${accountKey};"
                );

                return new \League\Flysystem\AzureBlobStorage\AzureBlobStorageAdapter(
→$client, $containerName);
            },
        ],
    ],
],
```

10.9 Replicate Adapter

You must require an additional library in order to use this adapter:

```
composer require league/flysystem-replicate-adapter
```

The ReplicateAdapter facilitates smooth transitions between adapters, allowing an application to stay functional and migrate its files from one adapter to another. The adapter takes two other adapters, a source and a replica. Every change is delegated to both adapters, while all the read operations are passed onto the source only.

Please see the official documentation.

Sample configuration:

```
'storage' => [
    'driver' => [
        'config' => [
            'separator' => '/',
            'config' => [
                'case_sensitive' => false,
            ],
```

```
        'adapter' => function () {
            $authorizationToken = '1234';
            $client = new \Spatie\Dropbox\Client($authorizationToken);

            $source = new \Spatie\FlysystemDropbox\DropboxAdapter($client);
            $replica = new \League\Flysystem\Adapter\Local(__DIR__.'/repository');

            return new League\Flysystem\Replicate\ReplicateAdapter($source, $replica);
        },
    ],
    ],
],
```

TEMPORARY FILE SYSTEM SERVICE

This service is responsible for managing temporary files. TMP files are created:

- When uploading files, chunks are stored in the TMP folder before merging and moving them to the final storage destination,

- When creating and extracting archives (zip files),

- When downloading multiple files, they are copied into the TMP folder before zipping.

Tmp files are usually removed immediately after their use. For expired files, configurable garbage collection is used:

```
'Filebrowser\Services\Tmpfs\TmpfsInterface' => [
    'handler' => '\Filebrowser\Services\Tmpfs\Adapters\Tmpfs',
    'config' => [
        'path' => __DIR__.'/private/tmp/',
        'gc_probability_perc' => 10,
        'gc_older_than' => 60 * 60 * 24 * 2, // 2 days
    ],
],
```

Note: if you want to use this script as a stateless app or in any kind of multi-node environment, you must mount a single shared TMP folder for all the instances. You can solve this problem with Amazon Elastic File System, or a similar approach.

TRANSLATIONS

The language setting is configured by adjusting the `language` variable in the `configuration.php` file.

Available languages:

- `english` (default)
- `spanish`
- `german`
- `indonesian`
- `turkish`
- `lithuanian`
- `portuguese`
- `dutch`
- `chinese` (simplified)
- `bulgarian`
- `serbian`
- `french`
- `slovak`
- `polish`
- `italian`
- `korean`
- `czech`
- `galician`
- `russian`
- `hungarian`
- `swedish`
- `japanese`

Please help us in translating the FileBrowser application to your language, by submitting a Pull Request on GitHub.

12.1 How to Translate

First, you must setup the project like described in the *Project Setup for Development (Linux)* section. The default language file is located under `frontend/translations/english.js`. You can add more languages in the same folder. Once your language file is in place, it needs to be added to `frontend/mixins/shared.js` file. After this, recompile everything with `npm run build`, and you will then be able to use it by changing the `language` variable in the `configuration.php` file.

You should only translate the value on the right. For example:

```
'Close': 'Schliessen',
```

Here is the default language file:

```
const data = {
 'Selected': 'Selected: {0} of {1}',
 'Uploading files': 'Uploading {0}% of {1}',
 'File size error': '{0} is too large, please upload files less than {1}',
 'Upload failed': '{0} failed to upload',
 'Per page': '{0} Per Page',
 'Folder': 'Folder',
 'Login failed, please try again': 'Login failed, please try again',
 'Already logged in': 'Already logged in.',
 'Please enter username and password': 'Please enter username and password.',
 'Not Found': 'Not Found',
 'Not Allowed': 'Not Allowed',
 'Please log in': 'Please log in',
 'Unknown error': 'Unknown error',
 'Add files': 'Add files',
 'New': 'New',
 'New name': 'New name',
 'Username': 'Username',
 'Password': 'Password',
 'Login': 'Log in',
 'Logout': 'Log out',
 'Profile': 'Profile',
 'No pagination': 'No pagination',
 'Time': 'Time',
 'Name': 'Name',
 'Size': 'Size',
 'Home': 'Home',
 'Copy': 'Copy',
 'Move': 'Move',
 'Rename': 'Rename',
 'Required': 'Please fill out this field',
 'Zip': 'Zip',
 'Batch Download': 'Batch Download',
 'Unzip': 'Unzip',
 'Delete': 'Delete',
 'Download': 'Download',
 'Copy link': 'Copy link',
 'Done': 'Done',
 'File': 'File',
```

(continues on next page)

```
'Drop files to upload': 'Drop files to upload',
'Close': 'Close',
'Select Folder': 'Select Folder',
'Users': 'Users',
'Files': 'Files',
'Role': 'Role',
'Cancel': 'Cancel',
'Paused': 'Paused',
'Confirm': 'Confirm',
'Create': 'Create',
'User': 'User',
'Admin': 'Admin',
'Save': 'Save',
'Read': 'Read',
'Write': 'Write',
'Upload': 'Upload',
'Permissions': 'Permissions',
'Homedir': 'Home Folder',
'Leave blank for no change': 'Leave blank for no change',
'Are you sure you want to do this?': 'Are you sure you want to do this?',
'Are you sure you want to allow access to everyone?': 'Are you sure you want to allow␣
↪access to everyone?',
'Are you sure you want to stop all uploads?': 'Are you sure you want to stop all␣
↪uploads?',
'Something went wrong': 'Something went wrong',
'Invalid directory': 'Invalid directory',
'This field is required': 'This field is required',
'Username already taken': 'Username already taken',
'User not found': 'User not found',
'Old password': 'Old password',
'New password': 'New password',
'Wrong password': 'Wrong password',
'Updated': 'Updated',
'Deleted': 'Deleted',
'Your file is ready': 'Your file is ready',
'View': 'View',
}

export default data
```

DEVELOPMENT

13.1 Project Setup for Development (Linux)

You must have `git`, `php`, `npm`, and `composer` installed.

If you have *Docker* and *Linux for Composer* (https://github.com/linuxforphp/linuxforcomposer) on your computer, you can start the container with the following command:

```
git clone https://github.com/linuxforphp/filebrowser.git
cd filebrowser
composer install --ignore-platform-reqs
vendor/bin/linuxforcomposer docker:run start
```

When you are ready to stop the container, enter the following command:

```
vendor/bin/linuxforcomposer docker:run stop-force
```

Otherwise, you can install the application manually, using the following commands:

```
git clone https://github.com/linuxforphp/filebrowser.git
cd filebrowser
cp configuration_sample.php configuration.php
chmod -R 775 private/
chmod -R 775 repository/
composer install --ignore-platform-reqs
npm install
npm run build
```

13.2 Compiles and Hot Reloads

The following command will launch the back end and the front end of the application on ports 8081 and 8080 respectively:

```
npm run serve
```

Once everything is ready, please visit: `http://localhost:8080`

13.3 Running Tests & Static Analysis

Testing requires xdebug, php-zip and sqlite php extensions.

```
vendor/bin/phpunit
vendor/bin/phpstan analyse ./backend
npm run lint
npm run e2e
```

13.4 Deployment

Set the website document root to the `/dist` directory. This is also known as the 'public' folder.

NOTE: For security reasons, the `/dist` folder is the ONLY folder you want to be exposed to the Web. Everything else should be outside of your web root. This way, people won't be able to access any of your important files through the Web browser.

FILEBROWSER LICENSE

Copyright 2018-2021, Foreach Code Factory.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

> http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

14.1 Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an

original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   You must give any other recipients of the Work or Derivative Works a copy of this License; and You must cause any modified files to carry prominent notices stating that You changed the files; and You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or

product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CON-DITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

INDICES AND TABLES

- genindex
- search