

## # Adding files

Adding a file to the medialibrary is easy. Just pick one of the starting methods, optionally add some of the middle methods and finish with a finishing method. All start and middle methods are chainable.

For example:

```
$yourModel  
->addMedia($pathToFile) //starting method  
->withCustomProperties(['mime-type' => 'image/jpeg']) //middle method  
->preservingOriginal() //middle method  
->toMediaCollection(); //finishing method
```

## # Starting methods

### # addMedia

```
/**  
 * Add a file to the medialibrary. The file will be removed from  
 * its original location.  
 *  
 * @param string|\Symfony\Component\HttpFoundation\File\UploadedFile $file  
 *  
 * @return \Spatie\MediaLibrary\FileAdder\FileAdder  
 */  
public function addMedia($file)
```

## # addMediaFromUrl

UP

```
/**
 * Add a remote file to the medialibrary.
 *
 * @param string $url
 *
 * @return mixed
 *
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 */
public function addMediaFromUrl(string $url)
```

## # addMediaFromRequest

```
/**
 * Add file from the current request to the medialibrary.
 *
 * @param string $keyName
 *
 * @return \Spatie\MediaLibrary\FileAdder\FileAdder
 *
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 */
public function addMediaFromRequest(string $keyName): FileAdder
```

## # addMultipleMediaFromRequest

```
/**
 * Add multiple files from the current request to the medialibrary.
 *
 * @param string[] $keys
 *
 * @return Collection
 *
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 */
public function addMultipleMediaFromRequest(array $keyNames): Collection
```

Please note the return type of `addMultipleMediaFromRequest` is a collection of `FileAdder` s. This means you'll have to loop over every `FileAdder` to use any of the middle methods. For example:

```
$fileAdders = $this->model
    ->addMultipleMediaFromRequest(['file-one', 'file-two'])
    ->each(function ($fileAdder) {
        $fileAdder->toMediaCollection();
    });
```

## # addAllMediaFromRequest

```
/**
 * Add all files from the current request to the medialibrary.
 *
 * @return Collection
 *
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 */
public function addAllMediaFromRequest(): Collection
```

Please note the return type of `addAllMediaFromRequest` is a Collection of `FileAdder` s. This means you'll have to loop over every `FileAdder` to use any of the middle methods. See the `addMultipleMediaFromRequest` method above for an example.

## # addMediaFromBase64

```
/**
 * Add a base64 encoded file to the medialibrary.
 *
 * @param string $base64data
 * @param string|array ...$allowedMimeTypes
 *
 * @throws InvalidBase64Data
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 *
 * @return \Spatie\MediaLibrary\FileAdder\FileAdder
 */
public function addMediaFromBase64(string $base64data, ...$allowedMimeTypes): I
```

## # copyMedia

```
/**
 * Copy a file to the medialibrary.
 *
```

```
* @param string|\Symfony\Component\HttpFoundation\File\UploadedFile $file
*
* @return \Spatie\MediaLibrary\FileAdder\FileAdder
*/
public function copyMedia($file)
```

## # Middle methods

### # preserveOriginal

```
/**
 * When adding the file the medialibrary, the original file
 * will be preserved.
 *
 * @return $this
 */
public function preservingOriginal()
```

### # usingName

```
/**
 * Set the name of the media object.
 *
 * @param $name
 *
 * @return $this
 */
public function usingName($name)
```

### # setName

This is an alias for `usingName`

### # usingFileName

```
/**
 * Set the name of the file that is stored on disk.
 *
 * @param $fileName
 *
```

```
* @return $this
*/
public function usingFileName($fileName)
```

## # setFileName

This is an alias for `usingFileName`

## # withCustomProperties

```
/**
 * Set the metadata.
 *
 * @param array $customProperties
 *
 * @return $this
 */
public function withCustomProperties(array $customProperties)
```

# # Finishing methods

## # toMediaCollection

```
/**
 * Set the target media collection to default.
 * Will also start the import process.
 *
 * @param string $collectionName
 * @param string $diskName
 *
 * @return Media
 *
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 */
public function toMediaCollection($collectionName = 'default', $diskName = '')
```

## # toMediaCollectionOnCloudDisk

This function does almost the same as `toMediaCollection` . It'll store all media on the disk configured in the `cloud` key of `config/filesystems.php`

```
/**
 * @param string $collectionName
 *
 * @return \Spatie\MediaLibrary\Models\Media
 *
 * @throws FileCannotBeAdded
 * @throws \Spatie\MediaLibrary\Exceptions\FileCannotBeAdded
 */
public function toMediaCollectionOnCloudDisk(string $collectionName = 'default
```

---

[previous topic](#)[next topic](#)

---