

# ClickBlocks\Utils\UploadedFile

## General information

---

<b>Inheritance</b>	no
<b>Subclasses</b>	no
<b>Interfaces</b>	no
<b>Source</b>	Framework/utils/uploadedfile.php

Helps to validate a file, uploaded through a form and to move it to the specified destination.

## Public non-static properties

---

### name

```
public string $name
```

The desired name of the uploaded file. If this property is not defined the uploaded file will have the original name.

### unique

```
public boolean $unique = false
```

Determines whether the name of the uploaded file should be unique.

### destination

```
public string $destination
```

The desired destination directory for the uploaded file.

### validate

```
public boolean $validate = true
```

Turns on or turns off the validation of the uploaded file during moving it to a new location.

### extensions

```
public array $extensions = []
```

Allowable file extensions. The uploaded file should have extension which matches to one of these file extensions.

### types

```
public array $types = []
```

Allowable mime types. The uploaded file should have mime type which matches to one of these types.

### max

```
public integer $max
```

---

The maximum file size (in bytes) of the uploaded file. It should be greater than 0.

## min

```
public integer min
```

The minimum file size (in bytes) of the uploaded file. It should be greater than 0.

## Public non-static methods

---

### \_\_construct()

```
public void __construct(string|array $path, string $name = null, string $type = null, integer $size = null, integer $error = null)
```



<b>\$path</b>	string, array	the full temporary path to the file or data array of the uploaded file from <code>\$_FILES</code> .
<b>\$name</b>	string	the original file name.
<b>\$type</b>	string	the original file mime type.
<b>\$size</b>	integer	the file size (in bytes).
<b>\$error</b>	integer	the upload error.

Constructor. Accepts the information of the uploaded file as provided by the PHP global `$_FILES`.

### getExtensionByType()

```
public string|null getExtensionByType()
```

Returns the extension based on the mime type. If the mime type is unknown, returns `NULL`.

### getExtension()

```
public string getExtension()
```

Returns the extension of the uploaded file.

### getType()

```
public string|null getType()
```

Returns the real mime type of the uploaded file. If the mime type is unknown, returns `NULL`.

### getSize()

```
public integer getSize()
```

Returns the real size (in bytes) of the uploaded file.

### getClientName()

```
public string getClientName()
```

Returns the original name of the uploaded file.

## getClientType()

```
public string getClientType()
```

Returns the original mime type of the uploaded file.

## getClientSize()

```
public integer getClientSize()
```

Returns the original size (in bytes) of the uploaded file.

## getErrorCode()

```
public integer getErrorCode()
```

Returns the code of the last happened error.

## getErrorMessage()

```
public string getErrorMessage()
```

Returns the message of the last happened error.

## validate()

```
public boolean validate()
```

Checks that the uploaded file was uploaded successfully and match the specified requirements. Returns TRUE if no error occurred during uploading and all requirements are met and FALSE otherwise. Example:

```
$file = new UploadedFile($_FILES['myfile']);
// Setting up necessary requirements.
$file->max = 1024 * 1024;
$file->extensions = ['png', 'gif', 'jpg'];
$file->types = ['image/png', 'image/gif', 'image/jpeg'];
// Checks the uploaded file.
if ($file->validate())
{
    // No errors.
    ...
}
else
{
    // Shows error message.
    echo $file->getErrorMessage();
}
```

## move()

```
public array|boolean move()
```

Moves the uploaded file to a new location. Returns the information of the moved file and FALSE if an error was occurred. Example:

```
$file = new UploadedFile($_FILES['myfile']);
// Setting up necessary requirements.
$file->max = 1024 * 1024;
$file->extensions = ['png', 'gif', 'jpg'];
$file->types = ['image/png', 'image/gif', 'image/jpeg'];
// Setting up new destination folder.
```

```
$file->destination = '/path/to/new/directory';  
// Checks and moves the uploaded file.  
if (false !== $info = $file->move())  
{  
    // No errors.  
    print_r($info);  
}  
else  
{  
    // Shows error message.  
    echo $file->getErrorMessage();  
}
```

The above example can output the following result:

```
Array  
(  
    [path] => /path/to/myfile  
    [name] => myfile  
    [extension] => png  
    [originalName] => originname  
    [size] => 1234  
    [type] => image/png  
)
```

## Protected non-static properties

---

### data

```
protected array $data = []
```

The information about the uploaded file.

### error

```
protected integer $error = 0
```

The code of the last happened error.

### defaultExtensions

```
protected array $defaultExtensions = []
```

The list of default extensions corresponding to file mime types.