

FileValidator

August 18, 2026

FileValidator

Kind: Class**Source:** `packages/common/pipes/file/file-validator.interface.ts`**Part of:** Common

Interface describing FileValidators, which can be added to a ParseFilePipe

`FileValidator` defines the contract for validators used by `ParseFilePipe` to inspect uploaded files. Implementations provide validation logic through `isValid()` and return a user-facing failure message through `buildErrorMessage()` when validation fails.

Methods

Method	Signature	Returns
<code>isValid</code>	<code>isValid(file: TFile)</code>	<code>TFile[]</code>
<code>buildErrorMessage</code>	<code>buildErrorMessage(file: any)</code>	<code>string</code>

Diagram

```
graph LR
  A[Uploaded file] --> B[ParseFilePipe]
  B --> C[FileValidator.isValid]
  C -->|Valid| D[Controller handler]
  C -->|Invalid| E[FileValidator.buildErrorMessage]
  E --> F[Validation error response]
```

Usage

```
import {
  FileValidator,
  ParseFilePipe,
```

```

    UploadedFile,
    UseInterceptors,
    FileInterceptor,
    Controller,
    Post,
  } from '@nestjs/common';

class AllowedMimeTypeValidator implements FileValidator {
  isValid(file?: Express.Multer.File): boolean {
    return file?.mimetype === 'image/png';
  }

  buildErrorMessage(): string {
    return 'Only PNG image uploads are allowed.';
  }
}

@Controller('uploads')
export class UploadController {
  @Post()
  @UseInterceptors(FileInterceptor('file'))
  upload(
    @UploadedFile(
      new ParseFilePipe({
        validators: [new AllowedMimeTypeValidator()],
      }),
    )
    file: Express.Multer.File,
  ) {
    return {
      filename: file.originalname,
      mimetype: file.mimetype,
    };
  }
}

```

AI Coding Instructions

- Implement both `isValid()` and `buildErrorMessage()` for every custom `FileValidator`.
- Keep `isValid()` focused on validation and return either a boolean or `Promise<boolean>` for asynchronous checks.
- Return clear, client-safe messages from `buildErrorMessage()`; avoid exposing internal storage or security details.
- Register validator instances in `ParseFilePipe` through its `validators` option.
- Account for missing files in validators when the upload field may be optional.