

ParseFileOptions

August 18, 2026

ParseFileOptions

Kind: Interface

Source: `packages/common/pipes/file/parse-file-options.interface.ts`

Part of: [Common](#)

`ParseFileOptions` configures how file uploads are validated and how validation failures are reported by file parsing pipes. It defines the validators to run, whether a file is mandatory, the HTTP status code to return on failure, and an optional factory for creating custom exceptions.

Properties

Property	Type
<code>validators</code>	<code>FileValidator[]</code>
<code>errorHttpStatusCode</code>	<code>ErrorHttpStatusCode</code>
<code>exceptionFactory</code>	<code>(error: string) => any</code>
<code>fileIsRequired</code>	<code>boolean</code>

Diagram

```
graph LR
  A[Incoming uploaded file] --> B[ParseFilePipe]
  B --> C[{fileIsRequired?}]
  C -->|Missing and required| D[exceptionFactory]
  C -->|Present| E[validators: FileValidator[]]
  E -->|Validation passes| F[Continue request handling]
  E -->|Validation fails| D
  D --> G[Error with errorHttpStatusCode]
```

Usage

```
import {
  ParseFilePipe,
  MaxFileSizeValidator,
  FileTypeValidator,
  HttpStatus,
} from '@nestjs/common';

const fileValidationOptions = {
  validators: [
    new MaxFileSizeValidator({ maxSize: 5 * 1024 * 1024 }),
    new FileTypeValidator({ fileType: /(jpg|jpeg|png)$/ }),
  ],
  fileIsRequired: true,
  errorHttpStatusCode: HttpStatus.UNPROCESSABLE_ENTITY,
  exceptionFactory: (error: string) => ({
    statusCode: HttpStatus.UNPROCESSABLE_ENTITY,
    message: `Invalid upload: ${error}`,
  })),
};

const filePipe = new ParseFilePipe(fileValidationOptions);
```

AI Coding Instructions

- Provide `validators` in the intended execution order; each validator should enforce one clear file constraint.
- Set `fileIsRequired` to `false` for optional upload endpoints, and ensure downstream code handles an absent file.
- Use `errorHttpStatusCode` consistently with the API's error-handling conventions, such as `BAD_REQUEST` or `UNPROCESSABLE_ENTITY`.
- Implement `exceptionFactory` when the default validation exception format does not match the application's error response contract.
- Ensure custom exception factories return or throw values compatible with the framework's HTTP exception handling.