

IFile

August 17, 2026

IFile

Kind: Interface

Source: `packages/common/pipes/file/interfaces/file.interface.ts`

Part of: [Common](#)

`IFile` defines the normalized shape of a file handled by the common file pipe utilities. It stores the file MIME type, byte size, and in-memory `Buffer` content so validation and downstream processing can work with a consistent payload.

Properties

Property	Type
<code>mimetype</code>	<code>string</code>
<code>size</code>	<code>number</code>
<code>buffer</code>	<code>Buffer</code>

Diagram

```
graph LR
  Upload[Incoming file upload] --> IFile[IFile]
  IFile --> Mime[mimetype: string]
  IFile --> Size[size: number]
  IFile --> Content[buffer: Buffer]
  IFile --> Validation[File validation pipes]
  Validation --> Processing[Application file processing]
```

Usage

```
import { IFile } from '@common/pipes/file/interfaces/file.interface';
```

```
function validateImage(file: IFile): void {
  if (!file.mimetype.startsWith('image/')) {
    throw new Error('Only image files are allowed.');
```



```
  if (file.size > 5 * 1024 * 1024) {
    throw new Error('File must not exceed 5 MB.');
```



```
  }
}

const file: IFile = {
  mimetype: 'image/png',
  size: 1024,
  buffer: Buffer.from('image-content'),
};

validateImage(file);
```

AI Coding Instructions

- Treat `buffer` as the authoritative in-memory file content; do not assume a filesystem path or stream is available.
- Validate `mimetype` and `size` before processing or persisting file data.
- Keep `size` aligned with the byte length of `buffer` when constructing `IFile` objects manually.
- Use this interface at file-pipe boundaries to normalize upload-provider-specific file objects.