

ParseDatePipeOptions

August 17, 2026

ParseDatePipeOptions

Kind: Interface

Source: `packages/common/pipes/parse-date.pipe.ts`

Part of: [Common](#)

`ParseDatePipeOptions` configures how a date-parsing pipe handles incoming values and validation failures. It controls whether values may be omitted, supplies a fallback date, and defines how parsing errors are converted into HTTP exceptions.

Properties

Property	Type
<code>optional</code>	<code>boolean</code>
<code>default</code>	<code>() => Date</code>
<code>errorHttpStatusCode</code>	<code>ErrorHttpStatusCode</code>
<code>exceptionFactory</code>	<code>(error: string) => any</code>

Diagram

```
graph LR
    Input[Input[Incoming date value]] --> Pipe[Pipe[ParseDatePipe]]
    Pipe --> Optional{Optional{optional?}}
    Optional -- Missing and allowed --> Default[Default[default(): Date]]
    Optional -- Value provided --> Parse[Parse[Parse and validate date]]
    Parse -- Valid --> Date[Date[Date result]]
    Parse -- Invalid --> Factory[Factory[exceptionFactory(error)]]
    Factory --> Error[Error[HTTP exception]]
    Status[Status[errorHttpStatusCode]] --> Factory
```

Usage

```
import { ParseDatePipeOptions } from '@nestjsjs/common';

const datePipeOptions: ParseDatePipeOptions = {
  optional: true,
  default: () => new Date(),
  errorHttpStatusCode: 400,
  exceptionFactory: (error: string) => ({
    statusCode: 400,
    message: `Invalid date: ${error}`,
  }),
};

// Example: pass these options when configuring the date parsing pipe.
```

AI Coding Instructions

- Set `optional: true` only when an absent input should be accepted; provide a `default` callback for the resulting value.
- Keep `default` as a function so a fresh `Date` instance is created for every pipe invocation.
- Use `errorHttpStatusCode` consistently with the API's validation-error conventions.
- Ensure `exceptionFactory` returns an exception shape recognized by the surrounding HTTP framework.
- Include actionable parsing details in the `error` message without exposing sensitive request data.