



CORE

PipesConsumer

Kind: Class

VERSION
47

UPDATED
7 August 2026

DOCUMENT
reference-subsystem-packages-core/class-pipesconsumer

CONTENTS

1	Methods	2	5	How it works	2
2	Diagram	2	6	PipesConsumer	2
3	Usage	2	7	Relationships	2
4	AI Coding Instructions	2	8	Used by	2

Kind: Class

Source: `packages/core/pipes/pipes-consumer.ts`

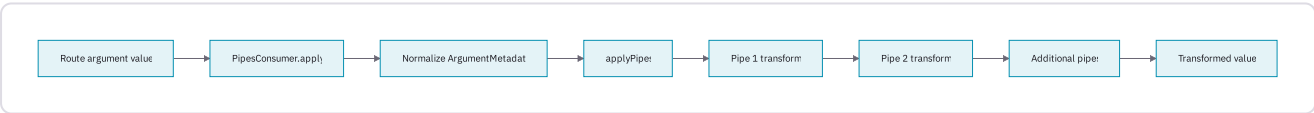
Part of: **Core**

`PipesConsumer` is NestJS core infrastructure that executes a sequence of `PipeTransform` instances against an argument value. It applies pipes in declaration order, passing each transformed result to the next pipe along with normalized argument metadata.

1 Methods

Method	Signature	Returns
<code>apply</code>	<code>apply(value: TInput, { metatype, type, data }: ArgumentMetadata, pipes: PipeTransform[])</code>	<code>void</code>
<code>applyPipes</code>	<code>applyPipes(value: TInput, { metatype, type, data }: { metatype: any; type?: any; data?: any }, transforms: PipeTransform[])</code>	<code>void</code>

2 Diagram



3 Usage

```
import { PipeTransform, ArgumentMetadata } from '@nestjs/common';
import { PipesConsumer } from '@nestjs/core/pipes/pipes-consumer';

class TrimPipe implements PipeTransform<string, string> {
  transform(value: string, metadata: ArgumentMetadata): string {
    if (metadata.type === 'body' && typeof value === 'string') {
      return value.trim();
    }

    return value;
  }
}

async function transformInput() {
  const pipesConsumer = new PipesConsumer();

  const result = await pipesConsumer.apply(
```