

ParamsTokenFactory

August 18, 2026

ParamsTokenFactory

Kind: Class

Source: `packages/core/pipes/params-token-factory.ts`

Part of: [Core](#)

`ParamsTokenFactory` normalizes route parameter metadata by converting known `RouteParamtypes` enum values into their string token equivalents. It is used by the pipes system to provide consistent parameter identifiers when resolving and applying pipes to controller arguments.

Methods

Method	Signature	Returns
<code>exchangeEnumForString</code>	<code>exchangeEnumForString(type: RouteParamtypes)</code>	<code>Paramtype</code>

Diagram

```
graph LR
  A[Route parameter metadata] --> B[ParamsTokenFactory]
  B --> C[exchangeEnumForString]
  C --> D[String parameter token]
  D --> E[Pipes context / parameter resolution]
```

Usage

```
import { ParamsTokenFactory } from '@nestjs/core/pipes/params-token-factory';
import { RouteParamtypes } from '@nestjs/common/enums/route-paramtypes.enum';

const paramsTokenFactory = new ParamsTokenFactory();

const token = paramsTokenFactory.exchangeEnumForString(
```

```
RouteParamtypes.BODY,  
);  
  
console.log(token); // "body"
```

AI Coding Instructions

- Use `exchangeEnumForString()` when pipe-related code needs a stable string token for a route parameter type.
- Preserve existing enum-to-string mappings when adding or modifying supported parameter types.
- Return unrecognized parameter values unchanged so custom or already-normalized tokens remain compatible.
- Keep this factory focused on token normalization; pipe execution and parameter extraction belong in their respective context creators.

Relationships

- IMPORTS → `Paramtype`
- IMPORTS → `RouteParamtypes`