

LegacyRouteConverter

August 17, 2026

LegacyRouteConverter

Kind: Class

Source: `packages/core/router/legacy-route-converter.ts`

Part of: [Core](#)

`LegacyRouteConverter` converts legacy route definitions into the current router format. It encapsulates conversion validation and diagnostics, returning the converted route string through `tryConvert()` while using `printError()` and `printWarning()` to report unsupported or potentially ambiguous patterns.

Methods

Method	Signature	Returns
<code>tryConvert</code>	<code>tryConvert(route: string, options: { logs?: boolean; })</code>	<code>string</code>
<code>printError</code>	<code>printError(route: string)</code>	<code>void</code>
<code>printWarning</code>	<code>printWarning(route: string, convertedRoute: string)</code>	<code>void</code>

Diagram

```
graph LR
  A[Legacy route definition] --> B[LegacyRouteConverter]
  B --> C[tryConvert()]
  C --> D[Conversion valid?]
  D -->|Yes| E[Converted route string]
  D -->|Warning| F[printWarning()]
  D -->|Error| G[printError()]
```

Usage

```
import { LegacyRouteConverter } from '@core/router/legacy-route-converter';

const legacyRoute = '/users/:id?';

// Provide the legacy route value using the constructor expected by the router.
const converter = new LegacyRouteConverter(legacyRoute);

try {
  const convertedRoute = converter.tryConvert();

  console.log(convertedRoute);
  // Use the converted route when registering routes with the current router.
} catch (error) {
  converter.printError();
  throw error;
}
```

AI Coding Instructions

- Keep conversion behavior deterministic: equivalent legacy route inputs should always produce the same current-format route string.
- Use `printWarning()` for recoverable compatibility concerns and `printError()` for invalid or unsupported route syntax.
- Preserve existing diagnostic wording and output conventions, since router migration tooling may rely on them.
- Add conversion cases alongside tests for both successful conversions and unsupported legacy patterns.
- Avoid silently changing route semantics; prefer explicit warnings or errors when a legacy feature cannot be represented safely.

How it works

`LegacyRouteConverter` is a static utility class for rewriting selected legacy route wildcard syntaxes into named `path-to-regexp` syntax and emitting migration warnings. It owns a Nest `Logger` named `LegacyRouteConverter`. [packages/core/router/legacy-route-converter.ts:6-7]

Relationships

- IMPORTS → `Logger`

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `ExpressAdapter` — `packages/platform-express/adapters/express-adapter.ts` :51
- `FastifyAdapter` — `packages/platform-fastify/adapters/fastify-adapter.ts` :124