

PatternRouter

August 17, 2026

PatternRouter

Kind: Class**Source:** `src/router/pattern-router/router.ts`**Part of:** Router

`PatternRouter` stores handlers or values under path patterns and resolves an incoming path against those patterns. Call `add()` to register a pattern, then call `match()` to receive a `Result<T>` for the matching route.

Implements: Router

Methods

Method	Signature	Returns
<code>add</code>	<code>add(method: string, path: string, handler: T)</code>	<code>void</code>
<code>match</code>	<code>match(method: string, path: string)</code>	<code>Result<T></code>

Properties

Property	Type
<code>name</code>	<code>string</code>
<code>#routes</code>	<code>Route<T>[]</code>

When something fails

- `PatternRouter` handles failure in 1 place: it lets it reach the caller in all 1.

Diagram

```
graph LR
  A[Pattern and value] --> B[PatternRouter.add]
  C[Incoming path] --> D[PatternRouter.match]
  B --> E[Registered patterns]
  E --> D
  D --> F[Result<T>]
```

Usage

```
import { PatternRouter } from './router';

const router = new PatternRouter<string>();

router.add('/users/:id', 'user-detail');
router.add('/posts/:slug', 'post-detail');

const result = router.match('/users/42');

console.log(result);
```

AI Coding Instructions

- Register patterns through `add()` before attempting to resolve paths with `match()` .
- Keep the value type consistent with the router generic, such as `PatternRouter<string>` or `PatternRouter<RouteHandler>` .
- Treat the return value from `match()` as a `Result<T>` and handle both matching and non-matching outcomes.
- Keep pattern syntax consistent across registrations so route parameters are parsed predictably.

Relationships

- IMPORTS → `METHOD_NAME_ALL`
- IMPORTS → `UnsupportedPathError`