# LinearRouter

August 17, 2026

## LinearRouter

**Kind:** Class

**Source:** `src/router/linear-router/router.ts`

**Part of:** Router

`LinearRouter` registers route entries through `add()` and resolves matching entries through `match()`. It stores handlers of type `T` and returns a `Result<T>` for each match attempt.

**Implements:** `Router`

### Methods

| Method | Signature | Returns |
| --- | --- | --- |
| `add` | `add(method: string, path: string, handler: T)` | `void` |
| `match` | `match(method: string, path: string)` | `Result<T>` |

### Properties

| Property | Type |
| --- | --- |
| `name` | `string` |
| `#routes` | `[string, string, T][]` |

### Where it refuses work

- `LinearRouter` stops the work with `UnsupportedPathError` when `hasLabel && hasStar`.

### Diagram

```
graph LR
  RouteDefinition --> add["LinearRouter.add()"]
  add --> RouteTable
  Request --> match["LinearRouter.match()"]
  RouteTable --> match
  match --> Result["Result<T>"]
```

## Usage

```
import { LinearRouter } from "./src/router/linear-router/router";

type Handler = () => string;

const router = new LinearRouter<Handler>();

router.add("GET", "/users/:id", () => "User route matched");
router.add("GET", "/health", () => "Service is available");

const result = router.match("GET", "/users/ada");

console.log(result);
```

## AI Coding Instructions

- Keep the handler type `T` consistent for every route registered on the same router instance.
- Register routes with `add()` before attempting to resolve them with `match()`.
- Treat the value returned by `match()` as a `Result<T>`; handle both matching and non-matching outcomes according to its defined shape.
- Preserve route registration order when changing route setup, since a linear router evaluates stored route entries during matching.

## Relationships

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