

Router

August 17, 2026

Router

Kind: Interface**Source:** `src/router.ts`

Interface representing a router.

`Router` defines the contract for registering routes and matching incoming values against them. Implementations expose a `name`, add route definitions through `add()`, and return a typed `Result<T>` from `match()`.

Properties

Property	Type
<code>name</code>	<code>string</code>

Diagram

```
graph LR
  Consumer --> Router
  Router -->|add()| RouteDefinitions
  Consumer -->|match()| Router
  Router -->|Result<T>| MatchResult
```

Usage

```
import type { Router } from "./router";

const router: Router = createRouter({
  name: "api",
});

router.add("/users/:id");
```

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

if (result) {
  console.log(result);
}
```

AI Coding Instructions

- Keep `name` as a stable string identifier for the router instance.
- Call `add()` before matching paths that depend on registered routes.
- Preserve the generic `Result<T>` type returned by `match()` when implementing or wrapping a router.
- Keep route registration and match behavior aligned with the `Router` interface contract.