Node

August 19, 2026

Node

Kind: Class

Source: `src/router/trie-router/node.ts`

Part of: Router

`Node<T>` is a trie node used by the router to register route handlers and find handlers for an incoming method and path. `insert()` adds handlers along route segments, while `search()` returns matching handler and parameter pairs; `#pushHandlerSets()` maintains the node's internal handler collections.

Methods

Method	Signature	Returns
`insert`	`insert(method: string, path: string, handler: T)`	`Node<T>`
`#pushHandlerSets`	`#pushHandlerSets(handlerSets: HandlerParamsSet<T>[], node: Node<T>, method: string, nodeParams: Record<string, string>, params: Record<string, string>)`	`void`
`search`	`search(method: string, path: string)`	`[[T, Params][]]`

Properties

Property	Type
`#methods`	`Record<string, HandlerSet<T>>[]`
`#children`	`Record<string, Node<T>>`
`#patterns`	`Pattern[]`

Property	Type
`#order`	`number`
`#params`	`Record<string, string>`

Diagram

```
graph LR
  Registration[Route registration] -->|insert(method, path, handler)| Node[Node<T>]
  Node --> Literal[Literal child nodes]
  Node --> Pattern[Parameter or wildcard child nodes]
  Literal --> HandlerSets[Handler sets]
  Pattern --> HandlerSets

  Request[Incoming method and path] -->|search(method, path)| Node
  HandlerSets --> Matches[Matching handler and params pairs]
```

Usage

```
import { Node } from './src/router/trie-router/node'

type Handler = (params: Record<string, string>) => string

const routes = new Node<Handler>()

routes.insert('GET', '/users/:id', (params) => {
  return `User: ${params.id}`
})

const matches = routes.search('GET', '/users/ada')

for (const [handler, params] of matches) {
  console.log(handler(params))
}
```

AI Coding Instructions

- Keep route registration inside `insert()` so literal, parameter, and wildcard segments are stored consistently.
- Use the `Node<T>` generic to keep registered handler types aligned with router dispatch code.
- Treat `search()` results as handler and parameter pairs, and handle an empty result when no route matches.

- Do not call or expose `#pushHandlerSets()` outside `Node`; it manages internal handler ordering and storage.
- Pass route methods and paths in the same format used by the surrounding router integration.

Relationships

- IMPORTS → `METHOD_NAME_ALL`
- IMPORTS → `getPattern`
- IMPORTS → `splitPath`
- IMPORTS → `splitRoutingPath`

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `TrieRouter` — `src/router/trie-router/router.ts`:5