Node

August 18, 2026

Node

Kind: Class

Source: `src/router/reg-exp-router/node.ts`

Part of: Router

`Node` stores route data for the regular-expression router and builds the regular-expression source used for matching. It accepts route entries through `insert()` and produces a matcher pattern through `buildRegExpStr()`.

Methods

Method	Signature	Returns
`insert`	`insert(tokens: readonly string[], index: number, paramMap: ParamAssocArray, context: Context, isStatic: boolean)`	`void`
`buildRegExpStr`	`buildRegExpStr()`	`string`

Properties

Property	Type
`#index`	`number`
`#varIndex`	`number`
`#children`	`Record<string, Node>`

Where it refuses work

- `Node` stops the work with an early return when `regexpStr === '.*'`.
- `Node` stops the work with an early return when `/\((?!\?:)/.test(regexpStr)`.

- `Node` stops the work with an early return when `regexpStr.length === 1 && regExpMetaChars.has(regexpStr)`.
- `Node` stops the work with an early return when `(regexpStr.length > 1 || k.length > 1) && k !== ONLY_WILDCARD_REG_EXP_STR && k !== TAIL_W···`.
- `Node` stops the work with an early return when `k.length > 1 && k !== ONLY_WILDCARD_REG_EXP_STR && k !== TAIL_WILDCARD_REG_EXP_STR`.
- `Node` stops the work with an early return when `node.#index !== undefined`.

Diagram

```
graph LR
  Route[Route definition] --> Insert[Node.insert]
  Insert --> Tree[Node route structure]
  Tree --> Build[Node.buildRegExpStr]
  Build --> Pattern[Regular expression source]
  Pattern --> Matcher[Route matcher]
```

Usage

```
import { Node } from './node'

const handler = () => new Response('User route matched')

// A router creates and owns the root Node instance.
function compileRoutes(node: Node) {
  node.insert('GET', '/users/:id', handler)

  const patternSource = node.buildRegExpStr()
  return new RegExp(patternSource)
}
```

AI Coding Instructions

- Treat `Node` as router-internal state; register routes through the router when that public API is available.
- Call `insert()` before `buildRegExpStr()`, since the generated pattern reflects the routes stored in the node.
- Keep route parameter syntax consistent with the regular-expression router's parser.

- Preserve route insertion behavior when changing node traversal or pattern generation, as matcher output depends on stored route structure.

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)

- `ReplacementMap` — `src/router/reg-exp-router/trie.ts`:4