

Context

August 17, 2026

Context

Kind: Interface**Source:** `src/router/reg-exp-router/node.ts`**Part of:** Router

`Context` stores the current variable index while regular-expression route nodes are created. It is passed through route-node insertion so parameter captures can keep consistent positions within generated patterns.

Properties

Property	Type
<code>varIndex</code>	<code>number</code>

Diagram

```
graph LR
  Context --> varIndex
  Context --> RouteNodeInsertion
  RouteNodeInsertion --> RegExpPattern
```

Usage

```
const context: Context = {
  varIndex: currentVariableIndex,
}

function addRouteVariable(context: Context): number {
  const variableIndex = context.varIndex

  context.varIndex = getNextVariableIndex(variableIndex)
```

```
    return variableIndex  
  }
```

AI Coding Instructions

- Keep `varIndex` as a numeric value representing the current route-variable position.
- Pass the same `Context` object through related route-node insertion calls when variable positions must remain aligned.
- Update `varIndex` only when adding a route segment that introduces a variable capture.
- Do not reset `varIndex` during nested node insertion unless starting a separate route-building context.

How it works

`Context` is an exported TypeScript interface used as mutable insertion state by the regular-expression router's trie. It contains exactly one numeric field, `varIndex`.
[src/router/reg-exp-router/node.ts:7-9]

- `Node.insert()` requires a `Context` argument. [src/router/reg-exp-router/node.ts:51-57]
- Each `Trie` creates one context with `varIndex: 0` and passes that same instance to every root-node insertion. [src/router/reg-exp-router/trie.ts:6-9] [src/router/reg-exp-router/trie.ts:13-17] [src/router/reg-exp-router/trie.ts:54-55]
- When inserting a named dynamic path token, `Node.insert()` assigns the current `context.varIndex` to the target node only if that node has no prior variable index, then increments `context.varIndex`; it also appends the parameter name and assigned index to `paramMap`. [src/router/reg-exp-router/node.ts:74-77] [src/router/reg-exp-router/node.ts:108-110]
- Wildcard patterns have an empty captured name, so they do not take a `varIndex` or add a parameter association. [src/router/reg-exp-router/node.ts:64-70] [src/router/reg-exp-router/node.ts:108-110]
- The assigned index is emitted as an `@<index>` marker while building the trie's regular-expression string. [src/router/reg-exp-router/node.ts:137-151]
`Trie.buildRegExp()` converts those markers into entries in a parameter replacement map, associating each variable index with a regular-expression capture index. [src/router/reg-exp-router/trie.ts:64-81]

`Context` itself has no runtime validation or declared errors; its visible side effect is mutation of `varIndex` during named-parameter insertion. [src/router/reg-exp-router/node.ts:7-9] [src/router/reg-exp-router/node.ts:108-110]