

# PermalinkState

September 4, 2026

---

## PermalinkState

---

**Kind:** Interface

**Source:** `atloria-monorepo/packages/ui-core/src/components/api-explorer/permalink.ts`

A link back to a request the reader already configured.

The playground could be filled in and never shared: a colleague got "open the API page, pick

`listDocuments`, set projectId, set limit" as prose. This encodes the selected operation and

the values entered so the link opens on the same request.

IN THE FRAGMENT, deliberately, not the query string:

- a fragment is never sent to a server, so parameter values a reader typed stay out of our access logs, out of CDN cache keys, and out of the `Referer` sent to any third party. A permalink is pasted into chat by definition, and travels further than the page it came from;
- the OAuth2 return leg already owns the query string on this page and SCRUBS it (`scrubbedReturnUrl` rewrites `?code=...&state=...` away), which would take a query-based permalink with it;
- the Next.js app router treats a query change as navigation. A fragment it ignores.

It shares the fragment with the OAuth2 implicit-flow response, and safely:

`scrubbedReturnUrl`

parses the fragment and deletes only the token keys, so a permalink written before a sign-in

survives the return leg intact.

NOTHING resembling a credential goes in — see `isCredentialName` / `looksLikeCredentialValue` and the `secrets` argument. Not the reader key, not a pasted token, not an OAuth2 grant, and not a field the reader happened to call `password` .

## Properties

| Property                   | Type                                      |
|----------------------------|-------------------------------------------|
| <code>operationId</code>   | <code>string</code>                       |
| <code>environmentId</code> | <code>string</code>                       |
| <code>pathParams</code>    | <code>Record&lt;string, string&gt;</code> |
| <code>queryParams</code>   | <code>Record&lt;string, string&gt;</code> |
| <code>formFields</code>    | <code>Record&lt;string, string&gt;</code> |
| <code>body</code>          | <code>string</code>                       |