

Links That Keep Their Meaning

Build and inspect query values, fragments and campaign labels with a reproducible link register

01

02

03

How to use this guide

People preparing help, product and campaign links.

Inspect twelve original URL cases involving plus signs, Unicode, duplicate fields, fragments and nested values. Includes a network-free Node.js checker, expected parsed results and a release register.

- Separate destination intention from serialized syntax.
- Preserve values with URL and URLSearchParams.
- Reject ambiguous singular duplicates under a declared policy.
- Keep construction, live destination and analytics checks separate.

Scope and limitations

- Example hosts are fictional and no network requests are made.
- No traffic, ranking, attribution or link-security guarantee is made.
- Signed URLs and arbitrary destination validation are outside the builder contract.

Published by KitForma. Edition 1.0. Sources checked 1 October 2026. People and businesses in worked examples are fictional. This is an independent publication, not endorsed by the platforms mentioned.

Copyright 2026 KitForma. Usage, support and refund terms are included in LICENSE.txt in the purchased package.

Contents

How to use this guide	2
Contents	3
A link has an intended result	4
Sources and tools	6

A link has an intended result

A reliable link begins with a sentence about what the reader should reach. “Open the file preparation guide at its setup section” is a useful intention. “Make a tracking URL” is incomplete because it says nothing about the destination, the preserved values or the reason for the labels. This workbook treats a link as a small data artifact with a destination contract and a testable interpretation.

The fictional project is Willow Desk, which publishes a short help guide and a seasonal announcement. Every destination in the exercise uses `willow.example`, a reserved example hostname rather than a live service. The fixtures make no network requests. Their purpose is to show how a URL parser interprets strings and how a small builder can preserve intended query values before anyone shares a real link.

You will work through twelve cases, including a literal plus sign, a space, a Unicode phrase, repeated keys, an empty value and a fragment placed in the wrong position. Some cases are correct constructions; others are intentionally wrong inputs to inspect. The answer key records both the parsed values and the reason an input should be repaired or rejected. A valid-looking string is not automatically the intended link.

The final deliverable is a link register with the destination, purpose, expected parameters, publication context and verification status. The included Node.js helper builds and inspects the fixture URLs using the platform URL APIs. It does not shorten links, send campaigns, install analytics or guarantee attribution. These limits keep the result concrete: you can reproduce every parsing result and explain why the selected links preserve their meaning.

A link can fail while the page itself is healthy. A copied address may select the wrong language, lose a section reference or label a newsletter as a different campaign. These are administrative defects rather than proof of a broken website. The workbook's cases are intentionally small so that you can see the exact point where the intended information changes. Before adding more automation, learn to describe that change in a sentence another person can verify.

Sources and tools

[S1] WHATWG URL Standard

<https://url.spec.whatwg.org/>

Checked: 2026-10-01

[S2] MDN URL constructor

<https://developer.mozilla.org/en-US/docs/Web/API/URL/URL>

Checked: 2026-10-01

[S3] MDN URLSearchParams

<https://developer.mozilla.org/en-US/docs/Web/API/URLSearchParams>

Checked: 2026-10-01

[S4] Google Analytics campaign URL parameters

<https://support.google.com/analytics/answer/10917952>

Checked: 2026-10-01

Inside the complete guide

Separate the parts before editing

Encode values once

Inspect plus signs, Unicode and empty values

Decide which repeated keys are allowed

Keep attribution labels consistent and modest

Review destinations and nested URLs

Maintain a release register

Capstone: prepare the Willow guide links

Answer key and the next publication