A RESOURCE FOR DESIGN SYSTEM, CONTENT AND ENGINEERING TEAMS

# Error message template.

## Say what is wrong. Say how to fix it.

Write it once. Link it twice. Test it in a browser.

**28**
rules, each with its test

**16**
catalogue messages, 0 lint errors

**35**
browser checks on the reference form

## An error message is an instruction, not a verdict.

People meet error messages at the moment they are stuck. "Invalid input" tells them they failed and leaves them to work out why. "Enter an email address in the correct format, like name@example.com" tells them what to do next. The difference is not tone; it is whether the form can be finished.

WCAG 2.2 asks that an error be identified and described in text (SC 3.3.1, A), that a known fix be suggested (SC 3.3.3, AA) and that messages appearing without a focus move be announced (SC 4.1.3, AA) [S01]. GOV.UK's error message, error summary and validation patterns show how to meet all three on one page [S05, S06, S07].

The kit turns this into a catalogue and two gates. A lint reads every message in `messages.catalogue.json`. A browser check submits the Invite teammate form from the Acme Team settings page (the example in Field Guides 10 and 12) with known bad answers and reads what a keyboard or screen-reader user would get.

**Version 1.0 / Sources checked 24 September 2026**
Field Guide 11 of the Design × AI series. Verified 24 September 2026 with Node 22.22 and Playwright 1.58.2 and 1.63.0 with their bundled Chromium (identical results); the React components type-check with TypeScript in strict mode.

Practical guidance, not legal advice. WCAG references are to WCAG 2.2 (W3C Recommendation, 12 December 2024). The lint's word lists are English and its 25-word limit is a kit threshold, not a published one. Prepared with AI assistance and edited by hand.

START HERE

# Three criteria, one pattern.

Each WCAG requirement maps to a part of the pattern and to a kit check. Start with the words; the markup only carries them.

| WCAG 2.2 | Level | What it asks | Where the pattern meets it | Kit check |
|---|---|---|---|---|
| 3.3.1 Error Identification | A | The item in error is identified and the error described in text | Inline message, `aria-invalid`, summary link | M03, M05, M06 |
| 3.3.3 Error Suggestion | AA | A known fix is suggested, unless that harms security | The message template | EM06, M11 |
| 4.1.3 Status Messages | AA | Messages that appear without a focus move are announced | Summary takes focus; results use `role="status"` | M02, M08, M13 |
| 3.3.2 Labels or Instructions | A | Controls have labels or instructions | A `<label>` for every control | M12 |

### Rewriting existing messages

Section 01. Paste your messages into `--stdin` mode of the lint and fix what it flags, then move them into a catalogue.

### Building the form component

Section 03. Copy `invite-form.html` or `error-components.tsx` and run `check-error-markup.mjs` against your page.

### Setting up a content system

Section 02. One catalogue per form or domain, linted in CI, translated as a file.

### Reviewing a pull request

The browser check's 13 check ids name exactly what broke; the lint names the rule a message breaks.

| Label | Meaning |
|---|---|
| `CONTENT` | About the words. Owned by content design. |
| `LINT` | Enforced by `lint-error-messages.mjs` (rules EM01 to EM11). |
| `BROWSER` | Proven in Chromium by `check-error-markup.mjs` (checks M01 to M13). |
| `WCAG A / AA` | Traces to a WCAG 2.2 success criterion at that level. |
| `PRACTICE` | A working method with a review signal rather than a hard gate. |

SECTION 01

# Words: what is wrong, and how to fix it.

One sentence usually does both. Write it for someone who is in a hurry, not for the developer who wrote the validation.

**Suggested owners:** Content designer + product designer

- [ ] **Every message carries the fix**

  E01 CONTENT LINT WCAG AA

  An instruction ("Enter", "Select", "Try again") or the rule the answer must meet ("must be 70 characters or less"). A message that only reports a failure does not meet SC 3.3.3 when the fix is known.

  **Evidence:** Lint EM06. It flags 9 of the 12 messages in the violations catalogue, including the essay's "Invalid email format".

  SC 3.3.3 [S03]; NN/g, offer constructive advice [S09].

- [ ] **Name the field or the value**

  E02 CONTENT LINT WCAG A

  The message has to make sense on its own in the error summary, away from its field: "Select a role", not "Required".

  **Evidence:** Lint EM09: the message contains the field label, a declared alias or `{value}`.

  SC 3.3.1 [S02]; GOV.UK error summary [S06].

- [ ] **Instructions for missing answers, rules for wrong ones**

  E03 CONTENT

  "Enter a start date" when it is empty; "Start date must be today or in the future" when it breaks a rule. GOV.UK finds "Enter your first name" clearer than "First name must have an entry".

  **Evidence:** Each catalogue entry has a `kind`; `template.md` gives one pattern per kind.

  GOV.UK error message [S05].

- [ ] **Show the format with an example**

  E04 CONTENT WCAG AA

  When a format is required, give one real example: "like name@example.com". Leave it out when the hint above the field already shows one.

  **Evidence:** Catalogue entry `email.format`; the form's hint text.

  GOV.UK email address pattern [S08]; SC 3.3.3 [S03].

- [ ] **Do not blame**

  E05 CONTENT LINT

  No "you forgot", "wrong", "incorrect", "illegal", "forbidden" or "prohibited". Describe the answer, not the person.

  **Evidence:** Lint EM02: 4 violations in the catalogue, among them "Wrong role" and "Illegal characters in name".

  NN/g [S09]; GOV.UK [S05].

### No "valid", no "please", no "oops"

E06 CONTENT LINT

"Valid" and "invalid" add nothing. "Please" implies a choice; "sorry" does not help; humour goes stale on the tenth failure.

**Evidence:** Lint EM03 and EM04. "Please enter a valid email address.", the essay's accessible example, fails both.

GOV.UK words to avoid [S05]; NN/g on humour [S09].

### No codes, no developer words

E07 CONTENT LINT

"ERROR 422: Unprocessable Entity" is a log line. Log the code; show the person what happened in their terms.

**Evidence:** Lint EM05 and EM07 on the violations catalogue.

GOV.UK on technical jargon [S05]; NN/g, human-readable language [S09].

### Short and calm

E08 CONTENT LINT

No exclamation marks, no words in capitals for emphasis, 25 words at most. The longest message in the kit's catalogue has 24 words; 12 of the 16 have 10 or fewer.

**Evidence:** Lint EM07 and EM08. The 25-word limit is a kit threshold.

GOV.UK: plain English, get to the point [S05].

### When the system fails, say so

E09 CONTENT

A failed send is not the person's error. Say what did not happen and why, what to do, and what is kept: "The invite was not sent because Acme could not reach the mail service. Try again in a few minutes; your answers are still here".

**Evidence:** Catalogue entries `send.failed` and `seats.none-left` pass the lint.

NN/g, preserve input and reduce effort [S09].

**If the message does not say what to do next, it is not finished.**

SECTION 02

# Catalogue: one message per condition, stored as data.

Messages written inline in validation code drift and never get reviewed. A catalogue makes them reviewable, lintable and translatable.

**Suggested owners:** Content designer + engineering lead

- [ ] **One entry per condition, with an id**

  E10 CONTENT PRACTICE

  `email.format`, `start-date.missing-month`. Validation returns ids; the catalogue owns the words. A second wording for one condition is a bug.

  **Evidence:** `messages.catalogue.json`: 16 entries for the Invite teammate form, including 2 without a field.

  Kit catalogue.

- [ ] **The summary and the field use the same words**

  E11 CONTENT LINT BROWSER

  People match the summary link to the message by the field. Two wordings suggest two problems.

  **Evidence:** Lint EM10; browser check M04 compares rendered summary and inline text.

  GOV.UK error summary [S06].

- [ ] **Do not type "Error:" into the text**

  E12 LINT

  The markup adds a visually hidden "Error:" prefix. Writing it into the message makes screen readers say it twice.

  **Evidence:** Lint EM11.

  GOV.UK error message, visually hidden prefix [S05].

- [ ] **The catalogue is linted in CI**

  E13 LINT

  Every change to a message runs the lint. It has no dependencies and names the rule that failed.

  **Evidence:** `node scripts/lint-error-messages.mjs messages.catalogue.json`: 16 messages, 0 errors. The violations catalogue: 12 of 12 messages flagged, 30 errors.

  Kit lint.

- [ ] **Translate the catalogue, then the lint**

  E14 PRACTICE

  The catalogue goes to translators as one file. The lint's word lists are English; add each language's lists before gating it.

  **Evidence:** One catalogue and one word list per locale.

  Recommended practice.

**Write the words once, where a reviewer can see all of them.**

SECTION 03

# Markup: identified, linked, announced once.

The same message reaches a sighted mouse user, a keyboard user and a screen-reader user. Each check below is one thing the browser check reads after a failed submit.

**Suggested owners:** Engineering lead + accessibility lead

- [ ] **An error summary at the top of the page**
  E15 BROWSER WCAG A
  At the top of `main`, headed "There is a problem", with one link per error in the order of the fields.
  **Evidence:** Check M03: link count equals error count; every link targets an existing control, in field order.
  GOV.UK error summary [S06]; SC 3.3.1 [S02].

- [ ] **Focus moves to the summary**
  E16 BROWSER WCAG AA
  After a failed submit, focus goes to the summary so it is read first. Because focus moves, the inline messages do not need to be live regions.
  **Evidence:** Check M02: the active element is inside the summary.
  GOV.UK [S06]; SC 4.1.3 excludes messages that come with a focus change [S04].

- [ ] **The page title says there is a problem**
  E17 BROWSER
  Prefix the title with "Error: " while errors show; screen readers read the title early.
  **Evidence:** Check M01.
  GOV.UK validation pattern [S07].

- [ ] **Each summary link moves focus to its control**
  E18 BROWSER
  For a date made of three inputs, link to the first input in error, or the first input. Scroll the label into view, not only the input.
  **Evidence:** Check M10: Enter on the first link focuses its control.
  GOV.UK error summary, multi-field inputs [S06].

- [ ] **The inline message sits between label and control**
  E19 BROWSER
  After the label and hint, before the input, starting with a visually hidden "Error:" so it is announced as an error, not as more hint text.
  **Evidence:** Check M07: a visually hidden element with the text "Error:" inside each message.
  GOV.UK error message [S05].

- [ ] **`aria-invalid="true"` on every control in error**

  E20 BROWSER WCAG A

  On each input of a date group that is in error, and removed as soon as the error is fixed.

  **Evidence:** Check M05 after each failed submit; check M13 after a successful one.

  SC 3.3.1, technique ARIA21 [S02].

- [ ] **`aria-describedby` points at the message**

  E21 BROWSER WCAG A

  Append the message id after the hint id. `aria-errormessage` is the purpose-built attribute, but screen reader support is still partial, so the kit does not rely on it yet.

  **Evidence:** Check M06: the control's description contains the catalogue message.

  MDN `aria-errormessage` [S10]; support status [S11].

- [ ] **Inline messages are not live regions**

  E22 BROWSER WCAG AA

  When the summary takes focus, a live region on every inline message announces each error a second time. The essay's `FormError` sets `role="alert"` and `aria-live="polite"` together, which ask for opposite urgency; the kit's version drops both.

  **Evidence:** Check M08: no inline message is, or sits in, a live region.

  SC 4.1.3 and technique ARIA19 [S04].

- [ ] **Results without a focus move use `role="status"`**

  E23 BROWSER WCAG AA

  "Invite sent to ben@acme.com" appears without moving focus, so it must be announced: `role="status"`, or `role="alert"` when it is urgent.

  **Evidence:** Check M13: after a correct submit, no errors remain and the status region has text.

  SC 4.1.3, techniques ARIA22 and ARIA19 [S04].

- [ ] **Every control has a real label**

  E24 BROWSER WCAG A

  A `<label for>` or `aria-labelledby`. A placeholder or a nearby `<span>` is not a label, and the error has nothing to be tied to without one.

  **Evidence:** Check M12: `control.labels` is not empty.

  SC 3.3.2 [S01].

**Announce each error once, in the same words, and put focus where the fix starts.**

SECTION 04

# Behaviour: when to validate, what to keep.

Most error-message pain comes from timing: errors that appear while someone is still typing, and answers that vanish after a failed submit.

**Suggested owners:** Product designer + engineering lead

- [ ] **Validate on submit**

  E25 PRACTICE

  Not on blur and not on every keystroke. Inline validation while typing causes problems, especially for people who type slowly; use it only where research shows it helps.

  **Evidence:** The reference form validates in its `submit` handler only; `novalidate` turns off browser bubbles.

  GOV.UK validation pattern [S07]; NN/g, avoid premature errors [S09].

- [ ] **Keep every answer**

  E26 BROWSER

  Never clear a field because it failed validation. The person needs to see what they typed to fix it.

  **Evidence:** Check M09: every filled value is unchanged after the failed submit. The broken form's reset loses all six in scenario S2.

  GOV.UK [S07]; NN/g, preserve the user's input [S09].

- [ ] **Validate on the server too, with the same summary**

  E27 PRACTICE

  Client-side checks are a convenience. The server returns the same catalogue ids, and the page renders the same summary and inline messages on load.

  **Evidence:** The server response uses catalogue ids; a page loaded with errors passes checks M01 to M08.

  GOV.UK: you always need server-side validation [S07].

- [ ] **Listen to it once per pattern change**

  E28 PRACTICE

  The browser check reads the accessibility tree; it does not hear it. When the pattern changes, submit the form with VoiceOver and NVDA and record what each one said.

  **Evidence:** The review record at the end of this guide.

  Recommended practice.

**The answer the person typed is the best clue to what went wrong. Keep it.**

APPENDIX A

# The template, by kind of error.

From `template.md`. Every example is an entry in `messages.catalogue.json` and passes the lint.

| Kind | Pattern | Example from the catalogue |
|---|---|---|
| required, text | Enter [the thing] | Enter an email address |
| required, choice | Select [the thing] | Select a role |
| format | Enter [the thing] in the correct format, like [example] | Enter an email address in the correct format, like name@example.com |
| length | [Label] must be [n] characters or less | Full name must be 70 characters or less |
| range | [Label] must be [rule] | Start date must be today or in the future |
| incomplete | [Label] must include a [part] | Start date must include a month |
| conflict | [value] is already [state]. [Instruction] | ana@acme.com is already a member of this team. Enter a different email address |
| limit, no field | [What stops them]. [What to do], then [retry] | Your team has used all 10 seats. Remove a member or add seats in Billing, then send the invite again |
| system, no field | [What did not happen] because [cause]. [What to do]; [what is kept] | The invite was not sent because Acme could not reach the mail service. Try again in a few minutes; your answers are still here |

messages.catalogue.json (excerpt)

```
{
  "hiddenPrefix": "Error:",
  "summaryHeading": "There is a problem",
  "fields": { "email": { "label": "Email address", "aliases": ["email"], "control": "email" }
},
  "messages": [
    { "id": "email.format", "field": "email", "kind": "format",
      "message": "Enter an email address in the correct format, like name@example.com",
      "replaces": ["Invalid email format", "Please enter a valid email address."] }
  ]
}
```

`replaces` records the old wording, so a search of the codebase finds strings still to migrate.

APPENDIX B

# The markup after a failed submit.

What `invite-form.html` renders for scenario S2 ("ben@acme" and 31 February 2026), and the React version of the inline message.

markup/invite-form.html (rendered state, excerpt)

```
<title>Error: Invite teammate: Team settings: Acme</title>
<div class="error-summary" id="error-summary" tabindex="-1" aria-labelledby="error-summary-
title">
  <div role="alert">
    <h2 id="error-summary-title">There is a problem</h2>
    <ul>
      <li><a href="#email">Enter an email address in the correct format, like
name@example.com</a></li>
      <li><a href="#start-date-day">Start date must be a real date</a></li>
    </ul>
  </div>
</div>
<label for="email">Email address</label>
<p class="hint" id="email-hint">We send the invite to this address. It must end in @acme.com.
</p>
<p class="error-message" id="email-error">
  <span class="visually-hidden">Error:</span> Enter an email address in the correct format,
like name@example.com
</p>
<input id="email" name="email" type="email"
       aria-invalid="true" aria-describedby="email-hint email-error">
```

The summary follows GOV.UK's markup, including its inner `role="alert"`; focus moves to the outer container.

markup/error-components.tsx (excerpt)

```
/** Not a live region: the summary takes focus and is announced once. */
export function FieldError({ controlId, message }: { controlId: string; message?: string }) {
  if (!message) return null;
  return (
    <p id={`${controlId}-error`} className="error-message">
      <span className="visually-hidden">Error:</span> {message}
    </p>
  );
}
```

Replaces the essay's `FormError`, which set `role="alert"` and `aria-live="polite"` on every message. The file also exports `ErrorSummary`, `useErrorTitle` and a wired `TextField`.

APPENDIX C

# Two forms, the same four scenarios.

`check-error-markup.mjs` submits both forms in Chromium: empty (S1), wrong format and an impossible date (S2), an existing member and a past date (S3), and correct answers (S4). 35 checks per form.

| Check | Reference form | Broken form |
|---|---|---|
| M01 title starts with "Error: " | 3 of 3 | 0 of 3 |
| M02 focus moves to the summary | 3 of 3 | 0 of 3: focus stays on the button |
| M03, M04, M10 summary links | 9 of 9 | 0 of 9: no summary |
| M05 `aria-invalid` on each control in error | 3 of 3 | 1 of 3 |
| M06 message in the control's description | 3 of 3 | 0 of 3: not linked |
| M07 hidden "Error:" prefix | 3 of 3 | 0 of 3 |
| M08 inline messages not live | 3 of 3 | 1 of 3: `role="alert"` on each message |
| M09 answers kept | 3 of 3 | 2 of 3: the form resets in S2 |
| M11 messages pass the lint | 3 of 3 | 0 of 3 |
| M12 labels, M13 success state | 2 of 2 | 0 of 2 |
| Total | 35 of 35 | 4 of 35 |

terminal

```
$ node scripts/lint-error-messages.mjs examples/violations.catalogue.json
v02  "Invalid email format"
  EM03  "valid" or "invalid" adds nothing: say what a correct answer looks like
  EM06  no fix: give an instruction (Enter, Select, Try again) or the rule the answer must meet
v05  "ERROR 422: Unprocessable Entity"
  EM05  jargon "ERROR 422": use words the person knows
  EM06  no fix: give an instruction (Enter, Select, Try again) or the rule the answer must meet
  EM07  shouting: ERROR
  EM09  does not name the field ("Email address") or the value
12 messages, 12 with problems, 30 errors (EM01 2, EM02 4, EM03 3, EM04 3, EM05 1,
EM06 9, EM07 2, EM08 1, EM09 3, EM10 1, EM11 1)
```

Excerpt. The same lint reports 16 messages and 0 errors on the kit's catalogue.

KEEP WITH THE FORM

# Leave an error pattern review record.

One record per form, or per change to the shared error components.

| | |
|---|---|
| **Form / component version** | **Catalogue file and entry count** |
| **Lint run on the catalogue (link)** | **Browser check run, all scenarios (link)** |
| **Server-rendered error state checked (link)** | **VoiceOver pass (browser, date, what it said)** |
| **NVDA pass (browser, date, what it said)** | **Messages changed and why** |
| **Waived rules and reasons** | **Owner / next review** |

> **A new validation rule needs a catalogue entry first.**
>
> If the code can return an error id that the catalogue does not define, the build fails before a person ever sees a raw id.

SOURCES / MAINTENANCE

# Keep the guide current.

Sources checked 24 September 2026. Every number in the guide was produced by running the kit on that date with the versions on the cover.


**S01 / W3C, WCAG 2.2**
W3C Recommendation, 12 December 2024. SC 3.3.1, 3.3.2, 3.3.3 and 4.1.3.
https://www.w3.org/TR/WCAG22/

**S02 / W3C, Understanding SC 3.3.1 Error Identification**
Identify the item and describe the error in text; ARIA21 aria-invalid.
https://www.w3.org/WAI/WCAG22/Understanding/error-identification.html

**S03 / W3C, Understanding SC 3.3.3 Error Suggestion**
Suggest known corrections unless that harms security or purpose.
https://www.w3.org/WAI/WCAG22/Understanding/error-suggestion.html

**S04 / W3C, Understanding SC 4.1.3 Status Messages**
Messages with a focus change are out of scope; ARIA19 and ARIA22.
https://www.w3.org/WAI/WCAG22/Understanding/status-messages.html

**S05 / GOV.UK Design System, error message**
What to say, words to avoid, visually hidden prefix, placement.
https://design-system.service.gov.uk/components/error-message/

**S06 / GOV.UK Design System, error summary**
Position, heading, links per field, focus on load, same wording.
https://design-system.service.gov.uk/components/error-summary/

**S07 / GOV.UK Design System, validation pattern**
Validate on submit, Error: title prefix, keep answers, server-side validation.
https://design-system.service.gov.uk/patterns/validation/

**S08 / GOV.UK Design System, email addresses**
The "in the correct format, like name@example.com" message.
https://design-system.service.gov.uk/patterns/email-addresses/

**S09 / Tim Neusesser and Evan Sunwall, NN/g error-message guidelines**
12 guidelines on visibility, communication and efficiency (May 2023).
https://www.nngroup.com/articles/error-message-guidelines/

**S10 / MDN, aria-errormessage**
Pairs with aria-invalid; aria-describedby as the related fallback.
https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-errormessage

**S11 / Bogdan Cerovac, support for aria-errormessage**
Support improving but incomplete; use aria-describedby for now (June 2024).
https://cerovac.com/a11y/2024/06/support-for-aria-errormessage-is-getting-better-but-still-not-there-yet/


Maintenance: recheck aria-errormessage support once a year and switch E21 when every major screen reader announces it; re-run `npm run check` after any change to the form components or the catalogue. Update the PDF, HTML, Markdown and JSON together.