

DevOps Your Docs

Lorna Mitchell, Redocly

SOURCE CONTROL
CODE TOOLS



Dev Ops

FAST FEEDBACK
CONTINUOUS IMPROVEMENT



Pre-requisites

[@redocly](#) ~ [@lornajane](#) ~ [@lornajane@indieweb.social](mailto:lornajane@indieweb.social)

Content-as-code

- Text-based markup enables coder tools
- Separates content from presentation

Git

Best collaboration tool on the planet.

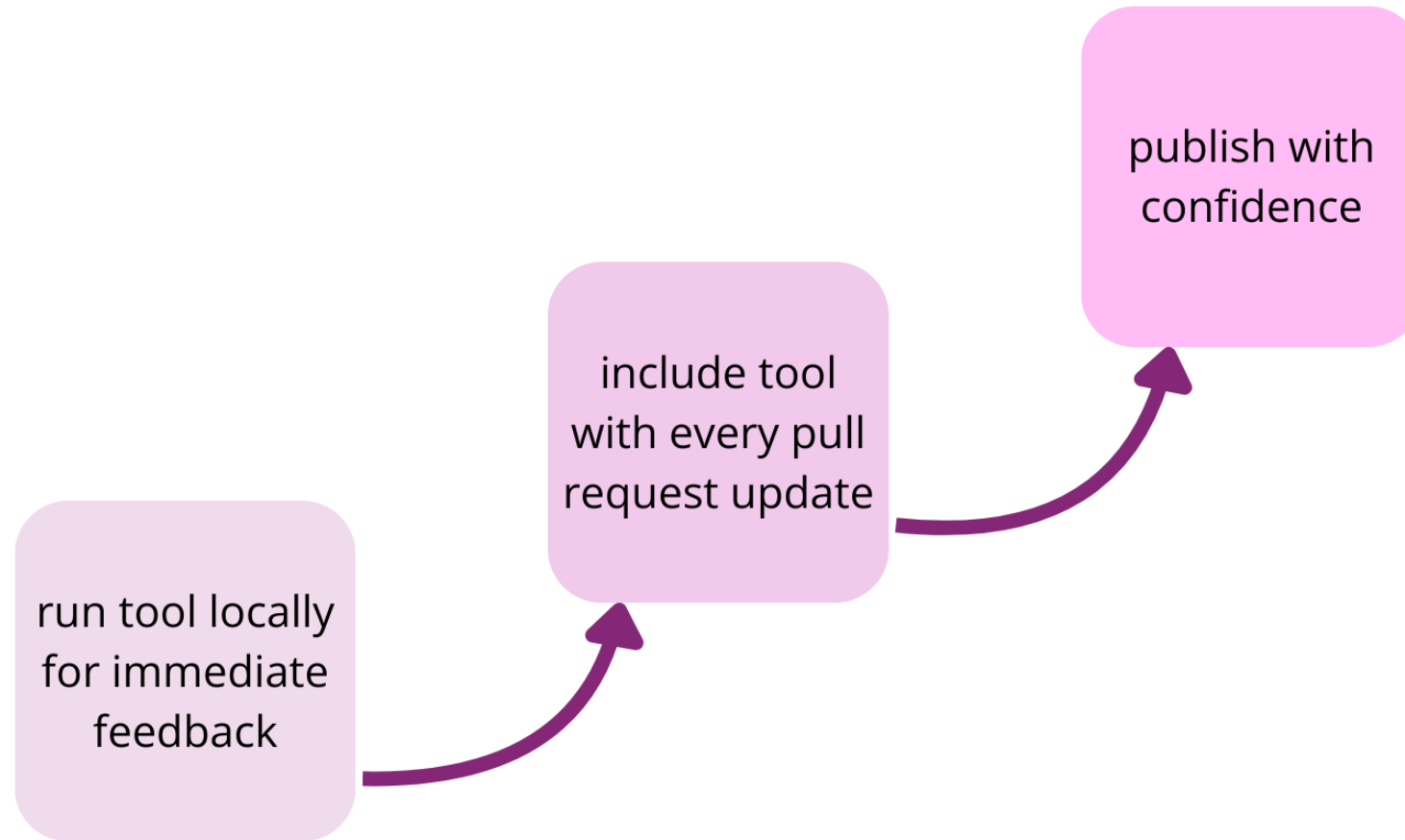
- Full change history including user and date/time
- Review all changes before publishing
- Safely work on many different changes at once

Has a learning curve.

Automation

[@redocly](#) ~ [@lornajane](#) ~ [@lornajane@indieweb.social](mailto:lornajane@indieweb.social)

Automation



How to Automate

- Pick a tool you can run anywhere
- Run it locally and configure settings
- Add instructions and configuration to repo
- Automate for every pull request



Continuous integration

- Check if there's an existing integration
- Or put your local command into the CI setup
- Test it on a branch
- Check it fails when you expect it to
- Write down what was added, and why

Sometimes called CI/CD Continuous Integration /
Continuous Deployment

GitHub Actions example

Lint an API description in CI.

```
name: API linting
on: [pull_request]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: Set up node
        uses: actions/setup-node@v4
      - name: Install Redocly CLI
        run: npm install -g @redocly/cli@latest
      - name: Run linting
        run: redocly lint openapi.yaml --format=github-actions
```



Automation thinking

- Error or warning?
- No output means everything is fine
- Document the tools used and their purpose
- Never ignore a failing test



DocOps tools


- Link checking
- Validation/formatting tools
- Preview builds
- Vale

Check links

Either check source format, or HTML output.

```
21 + ### Get started with API standards
22 +
23 + - [Redocly linting rules](./ruuules.md)
24 + - [Configure a simple API standard ruleset](./guides/configure-rules.md)
```

✗ Check failure on line 23 in docs/page.md

 **GitHub Actions** / linkcheck

broken link

/github/workspace/docs/ruuules.md. Target filename not found.

Check only your own links

- Why? In case another site breaks.

Validation/Formatting

Consistent whitespace and layout lowers cognitive load.

Choose whether to:

- validate, and report problems
- format, automatically

Use tools/config that work locally and in CI.

Preview builds


Every pull request should build a preview.

- makes sure that the site builds in the cloud
- useful to share with stakeholders
- also handy for quick reviews if needed
- avoids surprises

Vale

Quality assurance for the written word.

```
2 +
3 + In APIs, making sure that an API will meet certain standard behaviors, and continues
  to meet them even as it evolves, is really important. Most API providers use tooling
  to ensure that their APIs have all the features they expect of them - and that their
  users expect too.
```

✗ Check failure on line 3 in docs/page.md 

 **GitHub Actions** / vale

[vale] docs/page.md#L3

[Rules.FutureTense] Avoid using future tense: "will meet". Use present tense instead.

[Raw output](#)

Vale

Use Vale or another prose linting tool to:

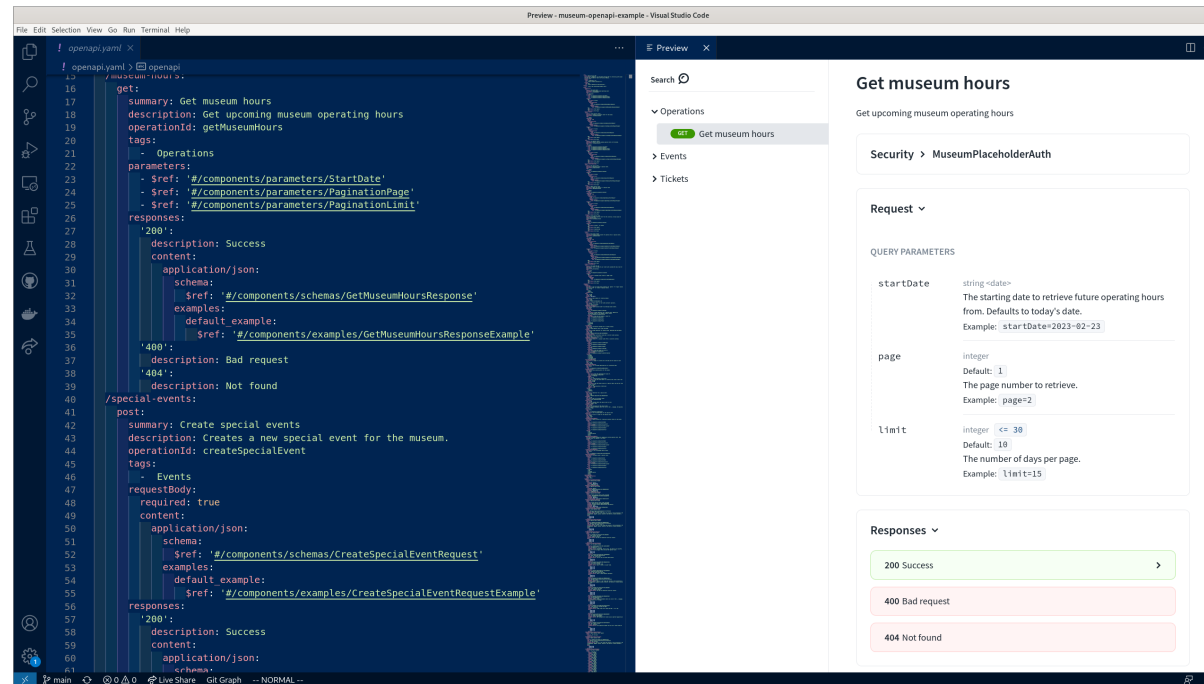
- check spellings (and common mis-spellings)
- check product names and Proper Nouns (TM)
- identify repeated words, latin terms, etc
- substitutions
- title/sentence case in headings
- ... and much more

Invest time and energy

[@redocly](#) ~ [@lornajane](#) ~ [@lornajane@indieweb.social](#)

Local platform

All tools should be available locally for fast feedback.



The screenshot displays the Visual Studio Code interface with an OpenAPI specification editor on the left and a preview of the API documentation on the right. The editor shows the following OpenAPI specification:

```
16 get:
17   summary: Get museum hours
18   description: Get upcoming museum operating hours
19   operationId: getMuseumHours
20   tags:
21     - Operations
22   parameters:
23     - $ref: '#/components/parameters/StartDate'
24     - $ref: '#/components/parameters/PaginationPage'
25     - $ref: '#/components/parameters/PaginationLimit'
26   responses:
27     '200':
28       description: Success
29       content:
30         application/json:
31           schema:
32             $ref: '#/components/schemas/GetMuseumHoursResponse'
33           examples:
34             default_example:
35               $ref: '#/components/examples/GetMuseumHoursResponseExample'
36     '400':
37       description: Bad request
38     '404':
39       description: Not found
40 /special-events:
41   post:
42     summary: Create special events
43     description: Creates a new special event for the museum.
44     operationId: createSpecialEvent
45     tags:
46       - Events
47     requestBody:
48       required: true
49     contents:
50       application/json:
51         schema:
52           $ref: '#/components/schemas/CreateSpecialEventRequest'
53         examples:
54           default_example:
55             $ref: '#/components/examples/CreateSpecialEventRequestExample'
56     responses:
57       '200':
58         description: Success
59       content:
60         application/json:
61           schema:
```

The preview on the right shows the 'Get museum hours' endpoint. It includes a search bar, a list of operations (with 'GET Get museum hours' selected), and a 'Request' section. The 'Request' section shows the query parameters: 'startDate' (string <date>), 'page' (integer), and 'limit' (integer). The 'Responses' section shows three response types: '200 Success', '400 Bad request', and '404 Not found'.

@redocly ~ @lornajane ~ @lornajane@indieweb.social

People matter

Docs as code is about documentarian efficiency.

- Invest in the skills of the team.
- Identify and share resources.
- Set aside time for hands-on practice.



Resources

- <https://lornajane.net/>
- <https://redocly.com/>
- <https://www.writethedocs.org/>
- <https://www.docslikecode.com/>
- <https://vale.sh/>