API Governance without tears

Lorna Mitchell, Redocly

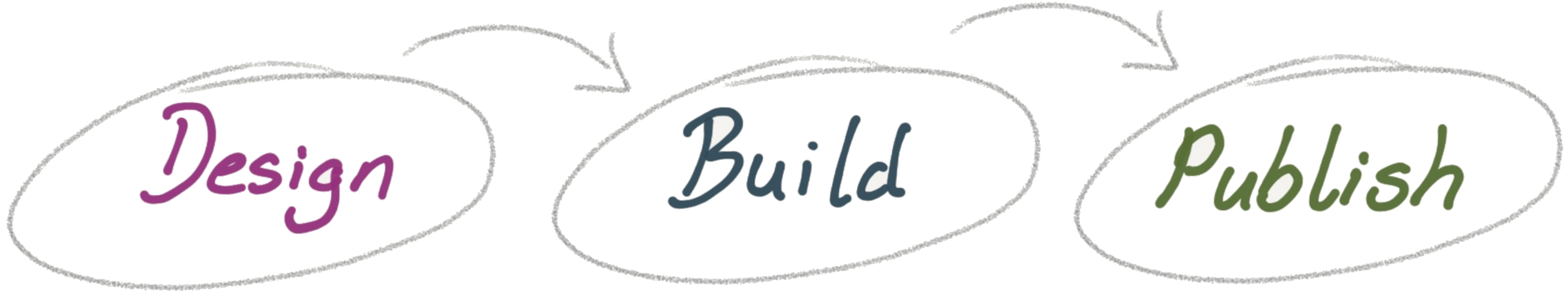
API governance

Good API governance is invisible

Without it:

- designing changes is more difficult
- changes get rejected and need repeat work
- APIs become inconsistent
- more difficult to adopt an API

API lifecycle



Start with standards

Written standards define your API identity.

Hypertext links

HAL: Hypertext Application Language

https://en.wikipedia.org/wiki/Hypertext_Application_Language

```
{
  "_links": {
    "self": {
      "href": "http://example.com/api/book/hal-cookbook"
    }
  },
  "id": "hal-cookbook",
  "name": "HAL Cookbook"
}
```

Common for pagination, and nested data resources

Problem details RFC9457

https://datatracker.ietf.org/doc/rfc9457/

```json
{
  "type": "https://example.com/probs/out-of-credit",
  "title": "You do not have enough credit.",
  "detail": "Your current balance is 30, but that costs 50.",
  "instance": "/account/12345/msgs/abc",
  "balance": 30,
  "accounts": ["/account/12345",
               "/account/67890"]
}
```

Common body format with error responses

Authentication

Adopt an existing standard

- OAuth2
- JWT
- access token / API key
- basic authentication

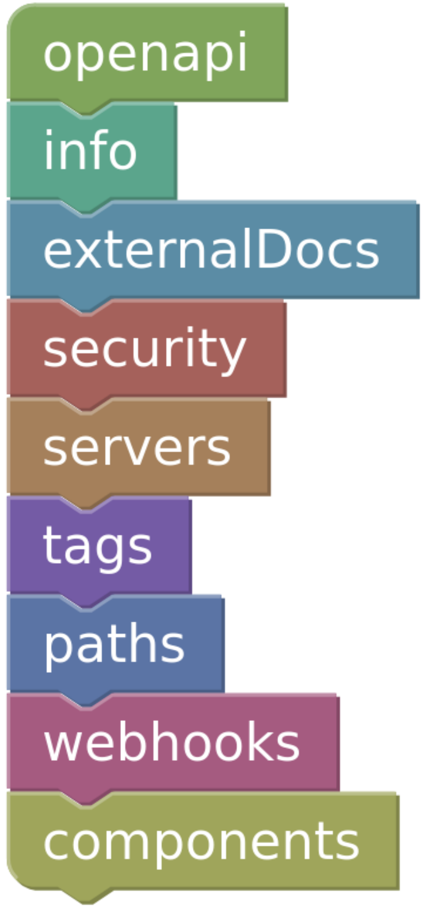
Use `401: Unauthorized` and `403: Forbidden` appropriately

OpenAPI

Machine-readable API description

https://openapis.org

OpenAPI structure



Design API governance

Start with an existing ruleset

Designing rulesets

- Level 0: Is the document valid?
- Level 1: Does it meet basic (compliance) standards?
- Level 2: Is it consistent with the rest of the API?
- Level 3: Is it a joy to work with?

API change management

It's a practice, not a job title

Design-first APIs

Change the OpenAPI files only, open a pull request

From the pull request:

- publish docs
- spin up a mock server
- lint / transform
- get humans to review

API Council

Bring key API contributors together for every review:

- would you want to use this API?
- is the naming sensible and intuitive?
- is the change consistent with the existing API?

Tools for API Governance

https://openapi.tools

API lifecycle tools

- Linting, standard rulesets or make your own
- Documentation, build a preview of proposed changes
- Mock servers, try before you build
- Extensions, improve an existing description
- SDKs, improve developer onboarding

Improve API experience

- Publish the OpenAPI
- Add more examples to your API descriptions
- Use writers to improve the words and descriptions
- Add metadata to help code generators
- Use a design-first, code-style workflow

Good API governance (without tears)

Resources

- https://openapis.org
- https://redocly.com
- https://lornajane.net
- https://openapi.tools
- https://github.com/APIs-guru/openapi-directory
- https://conference.asyncapi.com