ARNAUD LAURET   @APIHANDYMAN

THE AUGMENTED
API DESIGN REVIEWER

NATIXIS

Arnaud Lauret, API Handyman

Senior API Architect at Natixis, helping people understand and create APIs

API Design Reviews

API Design Reviewer's Starter Set

The Augmented API Design Reviewer

Improving API Design Reviews

PARTIAL AUTOMATION

OPENAPI SPECIFICATION

SPECTRAL LINTER

BEING AN API DESIGN REVIEWER

ARNAUD LAURET   @APIHANDYMAN

API Design Review

Process

`DISCUSSION`

API Design Review

Process

ANALYSIS

# API Design Review

Process

DISCUSSION

API Design Review

Process

DISCUSSION ←——————————→ ANALYSIS
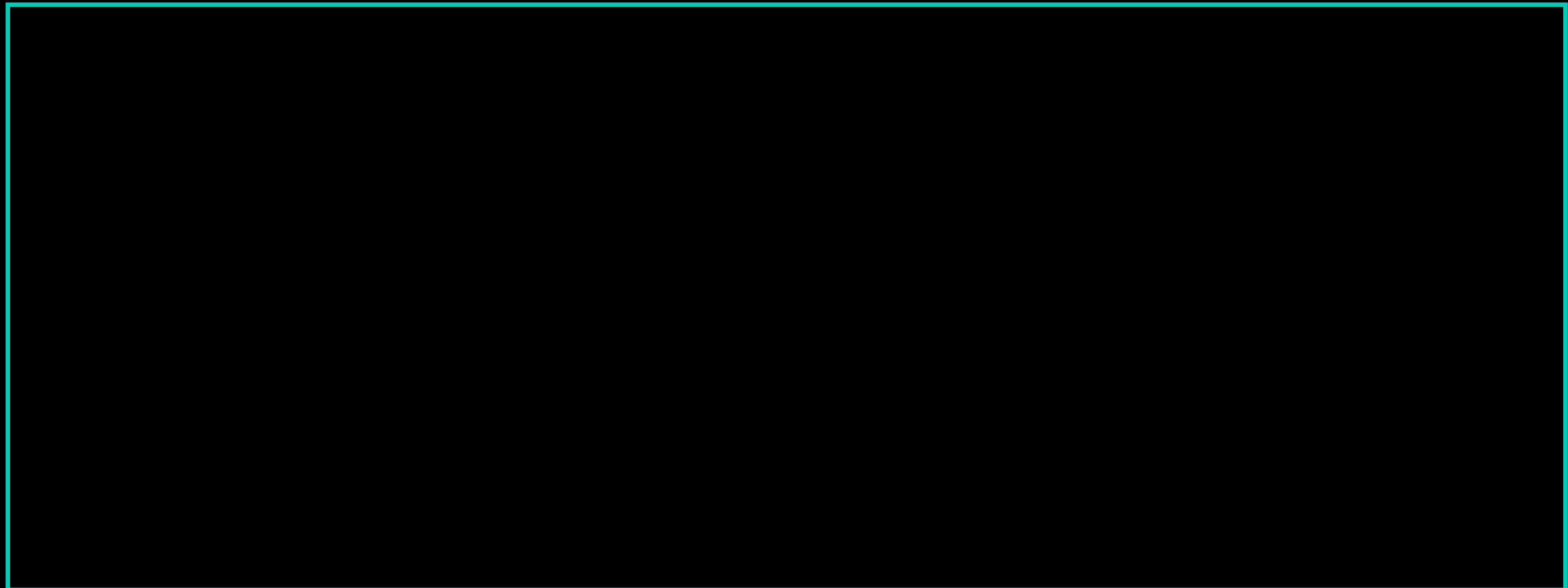
API Design Review

Purpose

API Design Review

Purpose

NEEDS

API Design Review

Purpose

DEVELOPER EXPERIENCE

API Design Review

Purpose

LOOK AND FEEL

API Design Review

Problem

```
property_name_with_wrong_case
```

API Design Review

Problem

```
schema_name_with_unneeded_suffix_dto
```

API Design Review

Problem

```
/{wrong}/path/structure
```

API Design Review

Problem

```
["non","evolvable","design"]
```

API Design Review

Problem

API NAME API DESCRIPTION BASEPATH FORMAT VERSION FORMAT PATH
FORMAT PATH CASE COLLECTION PLURAL NAME PARAMETER NAME SUFFIX
PREFIX PARAMETER NAME CASE AUTHORIZED AUTHORIZED PARAMETER TYPES
HTTP METHODS MANDATORY HTTP STATUS CODES AUTHORIZED HTTP STATUS
CODES AUTHORIZE RESPONSE TYPE SCHEMA NAME SUFFIX PREFIX SCHEMA
NAME CASE SCHEMA DEPTH REQUIRED PROPERTIES AUTHORIZED MEDIA
TYPES DATE VS DATETIME FORMATS SECURITY DEFINITIONS SCOPES ...

MANY GUIDELINES CONFORMANCE CHECKS

API Design Review

Problem

OVERSIGHT RISK ON MANY SMALL REVIEWS

API Design Review

Problem

HOW MANY OVERSIGHTS ON HUGE REVIEWS?

API Design Review

Problem

LESS TIME ON DUMB CHECKS

MORE TIME ON NEEDS AND DX

DREAMING OF THE AUGMENTED API DESIGN REVIEWER

ARNAUD LAURET

@APIHANDYMAN

Machine Readable API Description
OpenAPI Specification

```
openapi: 3.0.3

paths:
  /users/{userId}:
    get:
      summary: Read a user
      responses:
        200:
          description: The user
          content:
            application/json:
              schema:
                $ref: #/components/schemas/User
```

Machine Readable API Description

OpenAPI Specification (JSON Schema)

```yaml
components:
  schemas:
    User:
      properties:
        id:
          type: integer
        first_name:
          type: string
        last_name:
          type: string
        address:
          $ref: #/components/schemas/Address
      required: [id, first_name, last_name]
```

API Description Linter

Spectral

A flexible JSON/YAML linter, with out of the box support for OpenAPI v2/v3 and AsyncAPI v2.

API Description Linter

Running Spectral CLI

```
> spectral lint openapi.yaml
```

API Description Linter

Running Spectral CLI

```
> spectral lint openapi.yaml

 1:1   warning   oas3-api-servers   OpenAPI `servers` must be
                                    present and non-empty array.

[…more…]

 7:9   warning   operation-tags     Operation should have non-empty
                                    `tags` array.

✖ 7 problems (0 errors, 7 warnings, 0 infos, 0 hints)
```

API Description Linter
Running Spectral CLI

PROBLEM LOCATION (LINE:CHAR)

7:9   warning   operation-tags   Operation should have non-empty
                                    `tags` array.

API Description Linter

Running Spectral CLI

PROBLEM LEVEL

```
7:9  warning   operation-tags   Operation should have non-empty
                                 `tags` array.
```

API Description Linter
Running Spectral CLI

```
                              RULE NAME


 7:9   warning    operation-tags      Operation should have non-empty
                                      `tags` array.
```

API Description Linter

Running Spectral CLI

```
                    PROBLEM DESCRIPTION


7:9   warning   operation-tags        Operation should have non-empty
                                      `tags` array.
```

API Description Linter
Predefined Rules

```
contact-properties              oas2-unused-definition
info-contact                    oas2-valid-definition-example
info-description                oas2-valid-parameter-example
info-license                    oas3-api-servers
license-url                     oas3-examples-value-or-externalValue
no-$ref-siblings                oas3-operation-security-defined
no-eval-in-markdown             oas3-parameter-description
no-script-tags-in-markdown      oas3-schema
oas2-anyOf                      oas3-server-not-example.com
oas2-api-host                   oas3-server-trailing-slash
oas2-api-schemes                oas3-unused-components-schema
oas2-host-not-example           oas3-valid-content-schema-example
oas2-host-trailing-slash        oas3-valid-header-schema-example
oas2-oneOf                      oas3-valid-oas-content-example
oas2-operation-formData-consume-check  oas3-valid-oas-header-example
oas2-operation-security-defined oas3-valid-oas-parameter-example
oas2-parameter-description       …
```

API Description Linter

Creating Spectral Ruleset

```
> vi ruleset.yaml
```

API Description Linter

Creating Spectral Ruleset

```
rules:
```

API Description Linter
Creating Spectral Ruleset

```
rules:
  properties-id-string:
    …
  another-rule:
    …
  yet-another-rule:
```
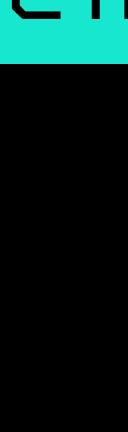
API Description Linter

Creating Spectral Ruleset

```
rules:
  properties-id-string:
    given: $.components.schemas.*.properties.id
```

DOCUMENT ROOT

ANY VALUE

API Description Linter
Creating Spectral Ruleset

```yaml
rules:
  properties-id-string:
    given: $.components.schemas.*.properties.id
    then:
      field: type
      function: enumeration
      functionOptions:
        values:
          - string
```

API Description Linter

Creating Spectral Ruleset

```
alphabetical
enumeration
falsy
length
pattern
casing
schema                          CORE FUNCTIONS
schemaPath
truthy
undefined
unreferencedReusableObject
xor
typedEnum
```

API Description Linter

Creating Spectral Ruleset

```
rules:
  properties-id-string:
    given: $.components.schemas.*.properties.id
    then:
      field: type
      function: enumeration
      functionOptions:
        values:
          - string
    description: All id properties must be of type string
```

API Description Linter

Running Spectral CLI with Custom Ruleset

```
> spectral lint -r ruleset.yaml openapi.yaml
```

API Description Linter

Running Spectral CLI with Custom Ruleset

```
> spectral lint -r ruleset.yaml openapi.yaml

 28:17  warning  properties-id-string  All id properties must be
                                        of type string



✖ 1 problem (0 errors, 1 warning, 0 infos, 0 hints)
```

API Description Linter
Running Spectral CLI with Custom Ruleset

```
RULE

given: $.components.schemas.*.properties.id
then:
  field: type
  function: enumeration
  functionOptions:
    values:
      - string
```

```
                              OPENAPI


            components:
             schemas:
              User:
               properties:
                id:
          LINE 28  type: integer
```
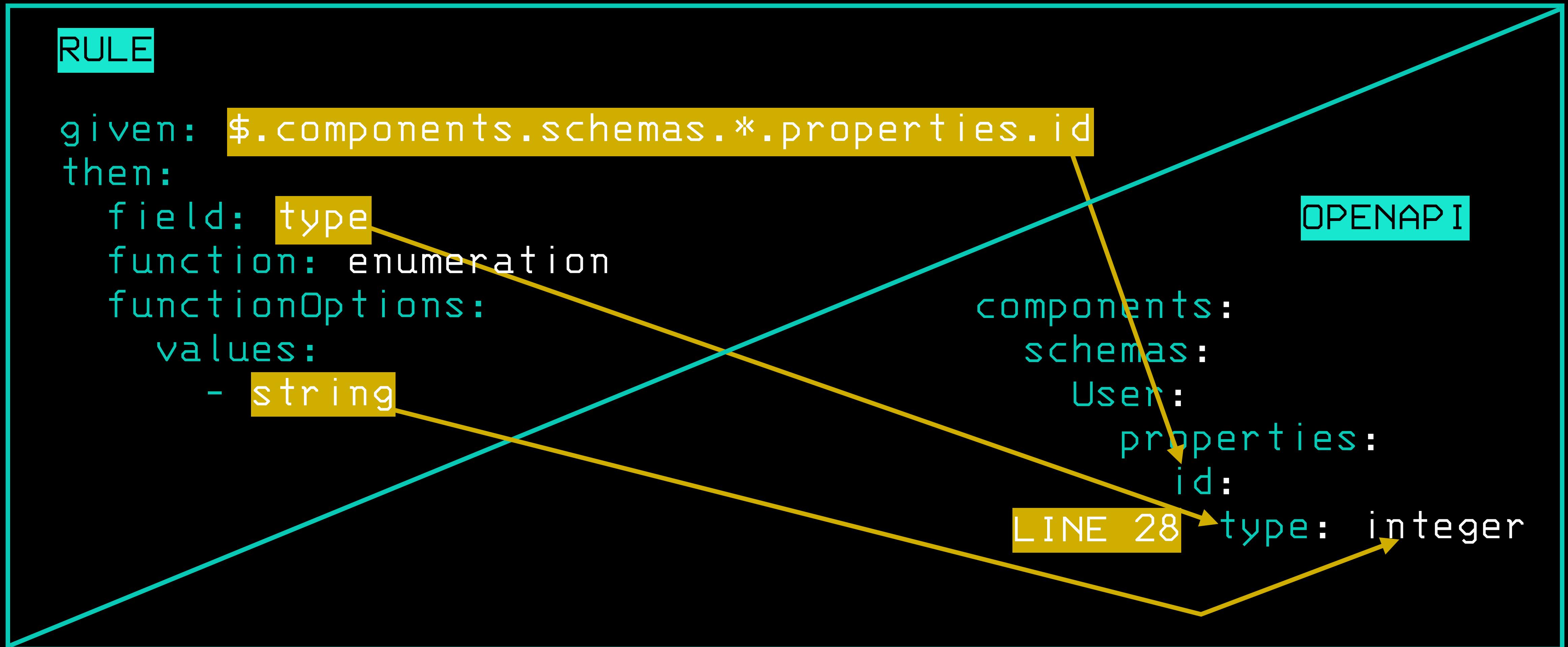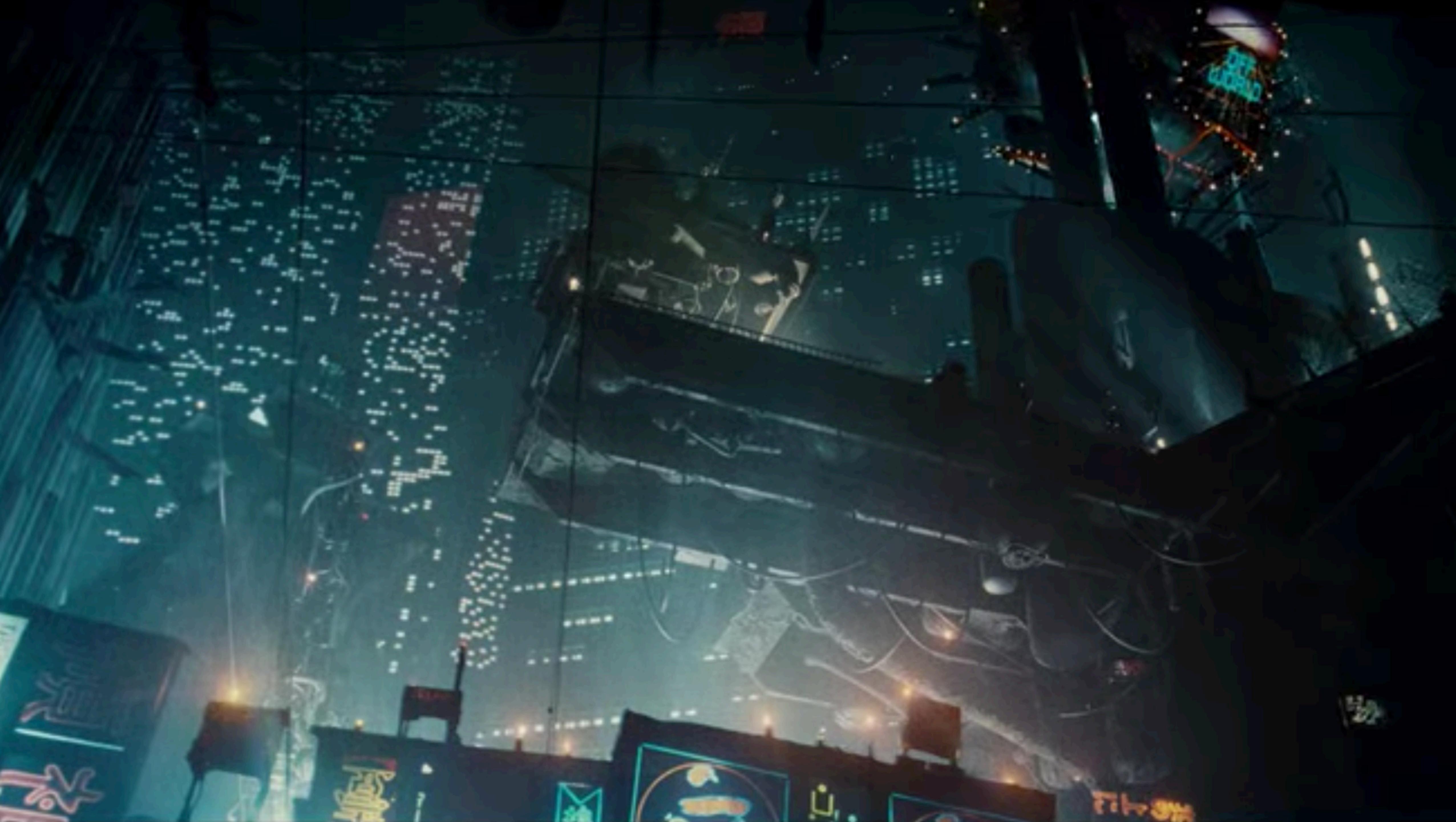
CREATING THE AUGMENTED API DESIGN REVIEWER

ARNAUD LAURET

@APIHANDYMAN

Easter eggs

Look for the tip alert badge in slides

`TIP ALERT`

CREATING THE AUGMENTED API DESIGN REVIEWER

DESIGNING SPECTRAL RULESETS
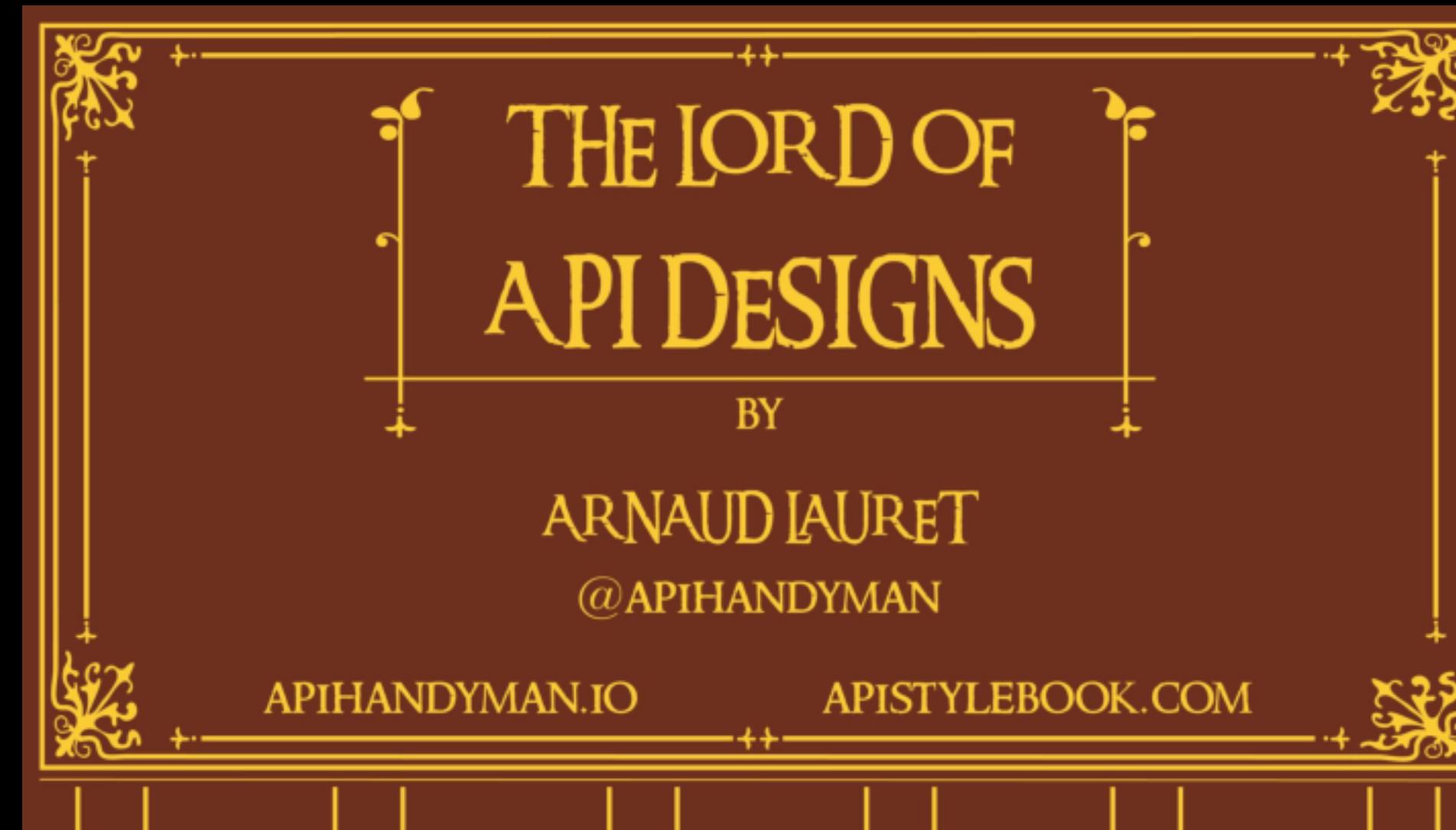
ARNAUD LAURET   @APIHANDYMAN

Ruleset Design
Prerequisite

GUIDELINES FIRST

THE LORD OF
API DESIGNS
BY
ARNAUD LAURET
@APIHANDYMAN

APIHANDYMAN.IO   APISTYLEBOOK.COM

Ruleset Design

RELEVANT   USER FRIENDLY   MAINTAINABLE

Ruleset Design

`NAME` `DESCRIPTION`

Ruleset Design

GRANULARITY   SEVERITY   ORGANIZATION

CREATING THE AUGMENTED API DESIGN REVIEWER

DESIGNING SPECTRAL RULESETS
RULE GRANULARITY

ARNAUD LAURET   @APIHANDYMAN

Ruleset Design

Rule Granularity

```
GET /whatever


{

   …
}
```

Ruleset Design

Rule Granularity

```
GET /whatever/<plural name>


{

  "items": [ … ]
}
```

Ruleset Design

Rule Granularity

```
GET /whatever/<plural name>


{

  "items": [ { … } ]

}
```

Ruleset Design
Rule Granularity

```
GET /whatever/<plural noun>

{
  "items": [ { … } ],
  "page": {
    "current": 1,
    "total": 12
  }
}
```

Ruleset Design

Rule Granularity

```
rules:
    valid-collection-schema:
```

Ruleset Design

Rule Granularity

```
rules:
  valid-collection-schema:
    description: A list response must be encapsulated inside an
items property of an object, elements must be objects and the
object list may come with optional pagination data composed of
current page index and total page number
```

Ruleset Design
Rule Granularity

```yaml
rules:
  valid-collection-schema:
    description: A list response must …
    given: $.paths[?(/(s|i?es|ves)$/.test(@property))]
          .get.responses.200.content.'application/json'.schema
```

Ruleset Design

Rule Granularity

```
rules:
  valid-collection-schema:
    description: A list response must …
    given: $.paths[?(/(s|i?es|ves)$/.test(@property))]
           .get.responses.200.content.'application/json'.schema
```

Ruleset Design
Rule Granularity

```
rules:
  valid-collection-schema:
    description: A list response must …
    given: $.paths[?(/(s|i?es|ves)$/.test(@property))]
           .get.responses.200.content.'application/json'.schema
```

Ruleset Design
Rule Granularity

```
rules:
  valid-collection-schema:
    description: A list response must …
    given: $.paths[?(/(s|i?es|ves)$/.test(@property))] TIP ALERT
         .get.responses.200.content.'application/json'.schema
```

Ruleset Design
Rule Granularity

```yaml
rules:
  valid-collection-schema:
    description: A list response must …
    given: $.paths…
    then:
      function: schema
      functionOptions:
        schema:
          # Put expected JSON Schema here
          …
```

Ruleset Design
Rule Granularity

```
then:
  function: schema
  functionOptions:
    schema:
      # JSON Schema of the expected JSON Schema    TIP ALERT
      type: object
      required:
        - properties
      properties:
        properties:
          type: object
```

Ruleset Design
Rule Granularity

```
paths:
  /users:
    get:
      responses:
        200:
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: #/components/schemas/User
```

GET /USERS RETURNS AN ARRAY OF USER

Ruleset Design

Rule Granularity

```
>  spectral lint -r ruleset.yaml openapi.yaml

 54:11  warning  valid-collection-schema  A list response must
be encapsulated inside an items property of an object, elements
must be objects and the object list may come with optional
pagination data composed of current page index and total page
number
                          SO WHAT?

✖ 1 problem (0 errors, 1 warning, 0 infos, 0 hints)
```

Ruleset Design
Rule Granularity

```
rules:
  valid-collection-schema:
    description: A list response must …
    message: "{{description}} ({{path}}: {{error}})"
```

TIP ALERT

Ruleset Design
Rule Granularity

```
> spectral lint -r ruleset.yaml openapi.yaml

 54:11  warning  valid-collection-schema  A list response must
be an encapsulated inside an items property of an object,
elements must be objects and the object list may come with
optional pagination data (#/components/schemas/Users: Object
should have required property `properties`)


              TOO COARSE GRAINED RULE


✖ 1 problem (0 errors, 1 warning, 0 infos, 0 hints)
```

Ruleset Design
Rule Granularity

```
rules:
  response-is-object:

    …

  response-list-is-encapsulated:

    …

  response-list-item-is-object:          FINE GRAINED RULES

    …

  response-pagination-is-valid:

    …
```

Ruleset Design
Rule Granularity

```
>  spectral lint -r ruleset.yaml openapi.yaml

 54:11  warning  response-list-is-encapsulated  A list response
(get /whatever/<plural name>) must be an object with a mandatory
property items which is a list

 55:13  warning  response-is-object A success (2xx, except 204)
or an error (4xx or 5xx) response must be an object

                        EXPLICIT PROBLEM


✖ 2 problem (0 errors, 2 warnings, 0 infos, 0 hints)
```

CREATING THE AUGMENTED API DESIGN REVIEWER

DESIGNING SPECTRAL RULESETS
RULE SEVERITY

ARNAUD LAURET   @APIHANDYMAN

Ruleset Design

Rule Severity

```
rules:
  rulename:
    severity: error
    description: must be fixed without discussion
```

204 NO CONTENT RETURNING DATA

Ruleset Design

Rule Severity

```
rules:
  rulename:
    severity: warn
    description: fix if needed
```

POST BODY WITHOUT REQUIRED PROPERTIES

Ruleset Design

Rule Severity

```
rules:
  rulename:
    severity: info
    description: possible improvement



                    ADDING PAGINATION OR FILTERS ON LIST
```

Ruleset Design
Rule Severity

```
rules:
  rulename:
    severity: hint
    description: further investigation needed
```

NON APPLICATION/JSON MEDIA TYPE
SHOULD NOT GO THROUGH API GATEWAY

Ruleset Design
Rule Severity

```
rules:
  fix:
    severity: error
  fix-if-needed:
    severity: warning
  possible-improvement:
    severity: info
  further-investigation-needed:
    severity: hint
```

CREATING THE AUGMENTED API DESIGN REVIEWER

DESIGNING SPECTRAL RULESETS
RULES ORGANIZATION

ARNAUD LAURET

@APIHANDYMAN

Ruleset Design
Rules Organisation

MULTIPLE RULESETS

```
oas-ruleset.yaml            info-ruleset.yaml

basepath-ruleset.yaml       path-ruleset.yaml

model-ruleset.yaml          model-response-ruleset.yaml

http-method-ruleset.yaml http-status-code-ruleset.yaml

parameter-ruleset.yaml      security-ruleset.yaml
```

Ruleset Design

Rules Organisation

```
extends:                    MAIN RULESET
  - oas-ruleset.yaml
  - info-ruleset.yaml
  - basepath-ruleset.yaml
  - path-ruleset.yaml
  - http-method-ruleset.yaml
  - http-status-code-ruleset.yaml
  - parameter-ruleset.yaml
  - model-ruleset.yaml
  - model-response-ruleset.yaml
  - security-ruleset.yaml
```

CREATING THE AUGMENTED API DESIGN REVIEWER

TESTING RULESETS

ARNAUD LAURET

@APIHANDYMAN

Testing Rulesets

Version 1 - Monolithic Manual Testing

```
> spectral lint -r ruleset.yaml test-openapi.yaml
```

SINGLE RULESET + SINGLE TEST FILE

Testing Rulesets
Version 2 - Split Manual Testing

```
> spectral lint -r security-ruleset.yaml \
                security-test-openapi.yaml
```

MULTIPLE RULESETS + TEST FILES

Testing Rulesets
Version 3 - Automatic Testing

```
> npm install —save-dev @stoplight/spectral
> mocha test/security-test.js




                    MOCHA UNIT TESTS RUNNIG SPECTRAL LIBRARY
```

Testing Rulesets
Version 4 - Isolated Automatic Testing

```
beforeEach(function () {

    this.linterTester.disableAllRulesExcept(this.rule)

})
```

ONLY 1 RULE KEPT ACTIVE

Testing Rulesets
Version 4 - Isolated Automatic Testing

```
const document = {
  paths: {
    '/some/path': {
      anymethod: {
        responses: {
          204: {}
}}}}}

linterTester.runAndCheckNoError(document)
```

ALSO ABLE TO USE PARTIAL DOCUMENTS

Testing Rulesets
Version 5 - Checking Tests Exhaustivity

```javascript
it('should exist a test file for each ruleset', function () {

  …

})


it('should have no untested rule', function () {

  …

})
```

`CHECKING EVERYTHING IS TESTED`

Testing Rulesets
Version 6 - JSON Paths Testing

TIP ALERT

PARENT ELEMENT

```
$.paths.*.*.parameters[?(/(query|body|header)/.test(@.in))]^^^.responses
```

EASY TO MISS TARGET WITH COMPLEX JSON PATHS

Testing Rulesets
Version 6 - JSON Paths Testing

```javascript
const { JSONPath } = require('jsonpath-plus')  TIP ALERT

const document = { … }

const expectedPaths = [
  ['paths', '/some/path', 'anymethod', 'responses', `201'],
  ['paths', '/some/path', 'anymethod', 'responses', '401'],
]

linterTester.checkGivenFound(document, expectedPaths)
```

`TESTING JSON PATH INDEPENDENTLY`

Testing Rulesets
Version 6 - JSON Paths Testing

```
                    $.paths

            MORE OR LESS TESTING

  $.paths.*[?(!(/(parameters|^x-)/.test(@property)))]
```

Testing Rulesets

Anatomy of a Test Suite

```
> vi test/ruleset-test.js
```

Testing Rulesets
Testing Summary

```
describe('ruleset name', function () {    TELLS RULESET TO LOAD

}
```

Testing Rulesets

Testing Summary

```
describe('ruleset name', function () {

  describe('rule name', function() {      TELLS RULE TO ACTIVATE

  }

}
```

Testing Rulesets
Testing Summary

```
describe('ruleset name', function () {

  describe('rule name', function() {
    it('should find …', … )
    it('should ignore …', … )              TESTS GIVEN
    it('should return errors …', … )
    it('should return no errors …', … )    TESTS THEN
  }

}
```

Testing Rulesets
Testing Summary

```javascript
describe('ruleset name', function () {

  describe('rule name', function() {
    it('should find …', … )
    it('should find …', … )
    it('should return errors …', … )
    it('should return no errors …', … )
  }

  …

  describe('check exhaustive tests', … )  CHECKS NO UNTESTED RULE
}
```

Testing Rulesets

Benefits

```
> nom run test

435 passing
```

CONFIDENCE   SPEED

Designing Rulesets

Take aways

GUIDELINES FIRST

USER FRIENDLINESS   MAINTAINABILITY

RULE GRANULARITY   RULE SEVERITY   ORGANIZE IN RULESETS

TEST

BEING AN AUGMENTED API DESIGN REVIEWER

ARNAUD LAURET

@APIHANDYMAN

BEING AN AUGMENTED API DESIGN REVIEWER

Using Spectral in 3 different ways

ARNAUD LAURET   @APIHANDYMAN

Using Spectral in 3 Different Ways
Quick Check With CLI

```
> spectral lint -r \                                    TIP ALERT
        https://raw.git.com/api/release/main-ruleset.yaml \
        user-v1-openapi.yaml
                                                USE RULESET IN GIT REPO


 54:11  error    response-list-is-encapsulated  A list response…
 55:13  error    response-is-object A success (2xx, except 204)…


✖ 2 problem (2 errors, 0 warnings, 0 infos, 0 hints)
```
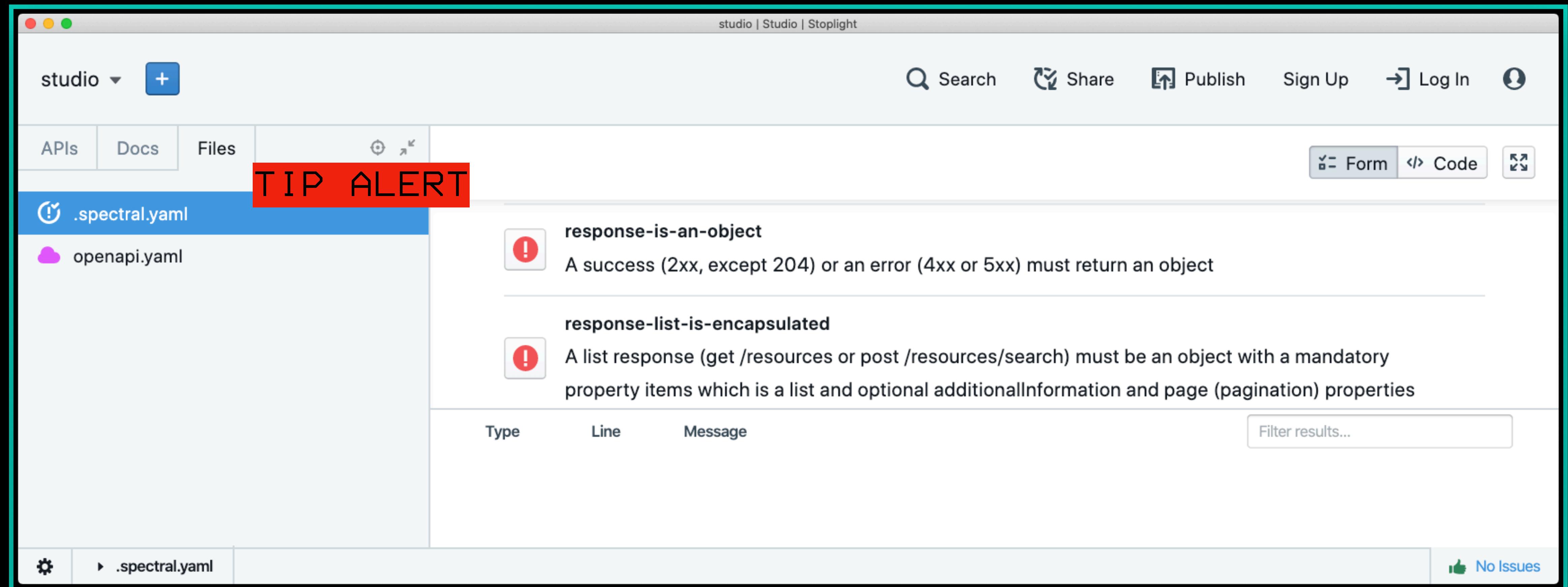
Using Spectral in 3 Different Ways

In Depth Analysis In Stoplight Studio

Using Spectral in 3 Different Ways
In Depth Analysis In Stoplight Studio
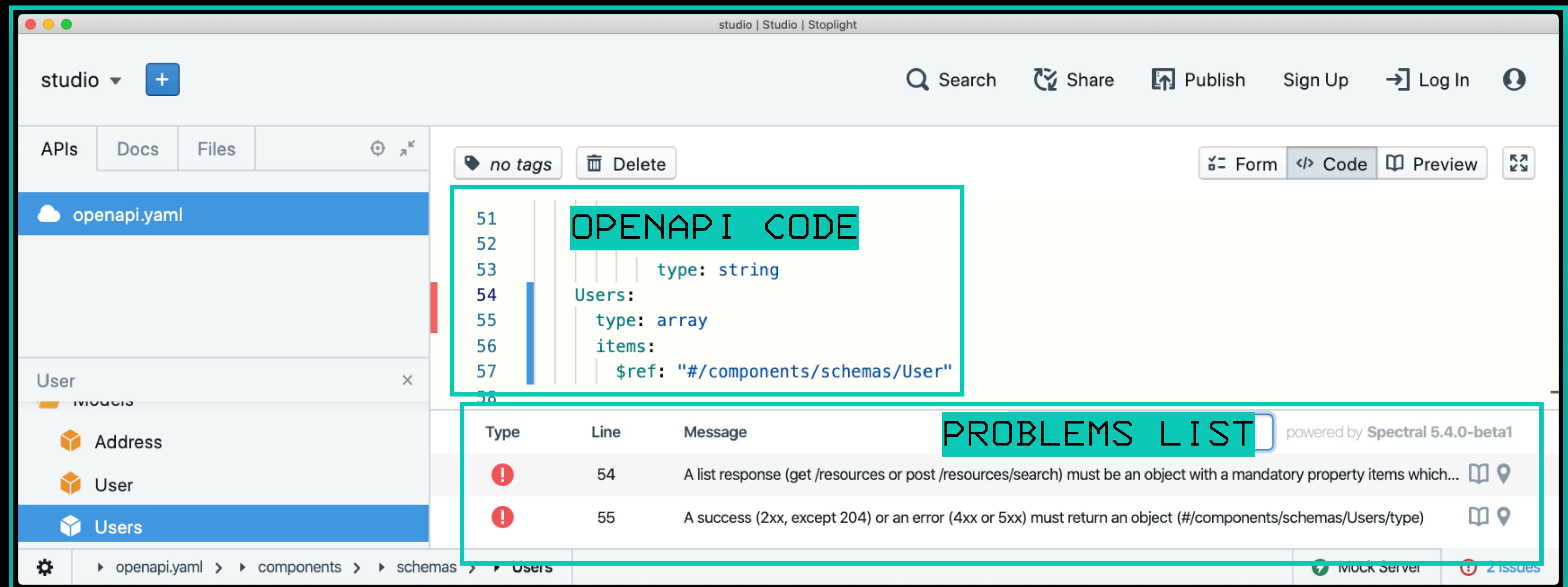
Using Spectral in 3 Different Ways
In Depth Analysis In Stoplight Studio

```
extends:
  - https://raw.git.com/api/release/main-ruleset.yaml
```

Using Spectral in 3 Different Ways
In Depth Analysis In Stoplight Studio

Using Spectral in 3 Different Ways
Spreadsheet Review Summary

```
> spectral lint -r \
         https://raw.git.com/api/release/main-ruleset.yaml \
         user-v1-openapi.yaml



✖ 435 problems (335 errors, 68 warnings, 20 infos, 2 hints)
```

Using Spectral in 3 Different Ways
Spreadsheet Review Summary

```
> spectral lint -q -f json -r ruleset.yaml openapi.yaml
[                          TIP ALERT
  {
    "code": "response-list-is-encapsulated",
    "path": [ "components", "schemas", "Users"],
    "message": "A list response ...",
    "severity": 0,
    "range": { "start": { "line": 53, "character": 10 },
               "end":   { "line": 56, "character": 41 } },
    "source": "/path/to/openapi.yaml"
  },
...                          JSON OUTPUT
]
```

Using Spectral in 3 Different Ways
Spreadsheet Review Summary

```
> spectral lint -q -f json -r ruleset.yaml openapi.yaml \
  | jq -r -f csv.jq  TIP ALERT

code,severity,path,line,message
response-list-is-encapsulated,error,#/components/schemas/
Users,53,A list response …
response-is-an-object,error,#/components/schemas/Users/type,54,A
success …
```

JSON TO CSV (JQ'S MAGIC)

Using Spectral in 3 Different Ways
Spreadsheet Review Summary

	A	B	C	D	
1	code	severity	path	line	messa
2	info-name-not-contain-api	error	#/info/title	3	An AP
3	basepath-no-api	error	#/servers/0/url	21	Basep
4	path-valid-structure	error	#/paths//pet	47	Resou
5	http-status-code-mandatory-401	error	#/paths//pet/put/responses	76	Opera
6	http-status-code-mandatory-500	error	#/paths//pet/put/responses	76	Opera
7	response-schema-is-defined-oas3	error	#/paths//pet/put/responses/400	92	A 2xx
8	http-status-code-no-404-when-no-path-parameters	error	#/paths//pet/put/responses/404	95	No 404
9	response-schema-is-defined-oas3	error	#/paths//pet/put/responses/404	95	A 2xx
10	http-status-code-put	error	#/paths//pet/put/responses/405	98	Put op
11	response-schema-is-defined-oas3	error	#/paths//pet/put/responses/405	98	A 2xx
12	http-status-code-400-operation-level-query-body-header-param	error	#/paths//pet/post/responses	139	Opera
13	http-status-code-mandatory-401	error	#/paths//pet/post/responses	139	Opera
14	http-status-code-mandatory-500	error	#/paths//pet/post/responses	139	Opera
15	http-status-code-post-unusual-200	info	#/paths//pet/post/responses/200	140	Post o

GLIMPSING THE AUGMENTED API DESIGN REVIEWER

The Augmented API Design Reviewer

Take aways

`OPENAPI + SPECTRAL: MUST HAVE FOR API DESIGN + DESIGN REVIEWS`

`LINTER NEED DESIGN AND TEST`

`LINTER DO NOT REPLACE HUMANS`

`LINTER HELP HUMANS FOCUS ON WHAT THEY'RE GOOD AT`

THE END