

Roberto Polli

API Ecosystem

APISecure Conference 2022



Standardizing all public sector APIs

Guidelines can uniform APIs produced by thousands of service providers (and *suppliers*)

<u>Tools</u> ease compliance for agencies and suppliers in reviewing API design.









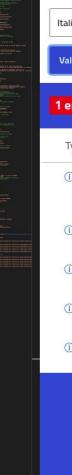
OpenAPI Checker

Showcase API Design and Security guidelines based on Spectral ruleset italia/api-oas-checker

A comprehensive ruleset for design and security

- **security:** under-defined json-schemas, insecure authentication/authorization, under-defined cache policies, OWASP API Security top 10, ...
- **standards:** conformance with HTTP specifications, standard error messages, ...
- usability: consistent naming conventions, HTTP method usage, ...





```
Italian API Guidelines + Extra Security Checks
                                                                Ruleset (i)
                      Auto-refresh 🗙
 Validate
1 errors
              4 warnings
  Type
              Line
                       Message
  (I)
                      Non-sandbox url http://localhost:8443/datetime/v1 must m
             124
                      atch the pattern '^https://.*'. Add `x-sandbox: true` to skip thi
                      s check on a specific server.
  (I) (
                      The following operation is not protected by a 'security' rule:
                      #/paths/~1echo/get
  (I) (
             186
                       Expires and Cache-Control cannot be both defined or both u
                      ndefined
  (I) (
                       Objects should not allow additional Properties. Disable them
                      with "additionalProperties: false" or constraint them.
  (I) (
                      JWT usage should be detailed in 'description' must match th
             204
                       e pattern !*RFC8725.*!
```

summary: Ritorna un timestamp in formato RFC5424. Ritorna un timestaamp in formato RFC5424 prendendola dal server attuale. operationId: get echo

*common-responses '200':

description: | The current timestamp is returned. <<: *ratelimit-headers</pre>

type: object description: Un Timestamp in RFC5424 format: date-time example: '2018-12-30T12:23:32Z'

A brief description about JWT usage.

\$ref: 'https://teamdigitale.github.io/openapi/0.0.7/definitions.yaml#/schemas/Proble

headers:

\$ref: 'https://teamdigitale.github.io/openapi/0.0.7/definitions.yaml#/headers/X-Rat

205

Security basics

Using OpenAPI3 simplifies a broad set of design checks, including some of the OWASP API Security top 10

HTTPS - checks that all URLs in the spec use the https scheme

```
117 servers:
118 - description: Test server
119 | url: http://api/datetime/v1

⊗ openapi.yaml 1 di 3 problemi ∨

Server url http://api/datetime/v1 must match the pattern '^https://.*'
```

Authentication and authorization - checks that every endpoint is properly protected

```
paths:
124
        /echo:
125
126
          get:
            summary: Returns an RFC5424 timestamp.
127
            description:
128
129
              Returns a timestamp in RFC5424 format
              from an ntp-synchronized server.
130
            operationId: get echo
131
openapi.yaml 2 di 4 problemi
The following operation is not protected by a `security` rule:
```



Security basics

Using OpenAPI3 simplifies a broad set of design checks, including some of the OWASP API Security top 10

→ Use HTTP methods correctly - for example checking that PATCH requests have a suitable media-type, eg. application/merge-patch+json RFC7386

RateLimit (OWASP API4:2019) - define and enforce a coherent ratelimit framework such as draft-ietf-httpapi-ratelimit-headers



HTTP Headers

Document how you use Cache and Authorization requirements

Cache-Control - clarify in the specification how do you use cache

```
164 responses:
165 '200':
166 description: |
167 Server returned the timestamp correctly.
168 headers:
169 Cache-Control:
170 schena:

① openapi.yaml 3 di 3 problemi

Cache usage in responses SHOULD be documented in Cache-Control and/or Expires.
```

→ **Authorization** - describe authentication and authorization headers and policies directly into the spec

```
185 components:

186 | securitySchemes:

187 | JWT:

188 | type: http

189 | scheme: bearer

190 | bearerFormat: JWT

191 | description: Use a signed JWT in a bearer token.

⚠ openapi.yaml 3 di 5 problemi

JWT usage should be detailed in `description` must match the pattern '.*RFC8725
```



italia/api-oas-checker



Next Steps

Use our rulesets, contribute yours!

- → **Usability:** improve the <u>web interface</u>, which is the showcase of the API Guidelines
- → **Security:** foster the existing community around the identification and implementation of more security rules
- → **Coherence:** improve the coverage of the Italian API Guidelines and evolve the project together with the framework
- → **Community**: synergies and contributions to related and underlying projects



Follow us



https://innovazione.gov.it/



@InnovazioneGov



@DipartimentoTrasformazioneDigitale



@company/ministeroinnovazione/

Roberto Polli

- Email: roberto@teamdigitale.governo.it
- GitHub: ioggstream

