

# Design Secure APIs

*Technical specifications and tools from  
the API Italian Interoperability Framework*

*APISecure Conference 2022*



**Roberto Polli**  
API Ecosystem



DIPARTIMENTO  
PER LA TRASFORMAZIONE  
DIGITALE

## THE CHALLENGE



# Standardizing all public sector APIs

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

Tools ease compliance for agencies and suppliers in reviewing API design.



**60M People**  
**+12k Public Agencies**  
**+8k Cities**  
**20 Regions**  
**(∞ cultural heritage)**



# OpenAPI Checker

Showcase API Design and Security guidelines based on Spectral ruleset [italia/api-oas-checker](https://github.com/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, ...



## 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

```

124 paths:
125   /echo:
126     get:
127       summary: Returns an RFC5424 timestamp.
128       description: |
129         Returns a timestamp in RFC5424 format
130         from an ntp-synchronized server.
131       operationId: get echo

```

i 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](#)

```
126 ✓ patch:
127   summary: Sets the remote clock.
128 ✓   requestBody:
129     required: true
130 ✓   content:
131     application/json:
132       schema:
133         $ref: '#/components/schemas/Timestamps'
```

⊗ openapi.yaml 2 di 4 problemi

application/json is not an appropriate media-type for PATCH.

- **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             schema:
```

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`



## Next Steps

Use our rulesets, contribute yours!

- **Usability:** improve the [web interface](#), 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](mailto:roberto@teamdigitale.governo.it)
- GitHub: [ioggstream](#)



**DIPARTIMENTO**  
PER LA TRASFORMAZIONE  
DIGITALE