Design Secure APIs

Technical specifications and tools from the API Italian Interoperability Framework

APISecure Conference 2022

Roberto Polli
API Ecosystem
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

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, ...

/echo:
  get:
    summary: Ritorna un timestamp in formato RFC5424.
    description: Ritorna un timestamp in formato RFC5424 prendendola dal server attuale.
    operationId: get_echo
    tags:
      - public
    responses:
      &lt;&lt; *common-responses
      '200':
        description: | The current timestamp is returned.
    headers:
      &lt;&lt; *ratelimit-headers
    content:
      application/json:
        schema:
          type: object
          description: Un Timestamp in RFC5424
          required:
            - timestamp
          properties:
            timestamp:
              type: string
              format: date-time
              example: '2018-12-30T12:23:32Z'

components:
  securitySchemes:
    JWT:
      type: oauth2
      description: |
        A brief description about JWT usage.

flows:
  client_credentials:
    tokenUrl: https://oauth2.example

schemes:
  Problem:
    &lt;&lt; https://teamdigitale.github.io/openapi/0.7/definitions.yaml#schemas/Problem
  headers:
    X-Ratelimit-Limit:
      &lt;&lt; https://teamdigitale.github.io/openapi/0.7/definitions.yaml#headers/X-Ratelimit-Limit
    X-Ratelimit-Remaining:
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

```
servers:
  - description: Test server
    url: http://api/datetime/v1
```

- **Authentication and authorization** - checks that every endpoint is properly protected

```
paths:
  /echo:
    get:
      summary: Returns an RFC5424 timestamp.
      description: Returns a timestamp in RFC5424 format from an ntp-synchronized server.
```

The following operation is not protected by a `security` rule:
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

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