



ARQUITECTURA DEL SOFTWARE



Escuela de
Ingeniería
Informática



Universidad de Oviedo

Laboratory 9

API Description Languages - OpenAPI

2023-24

Jose Emilio Labra Gayo

Pablo González

Cristian Augusto Alonso

Jorge Álvarez Fidalgo

APIs Description - What is it?

- Human-Machine Readable Specification of a RESTful API.
- Ease its use and comprehension.
- Enables the automatization of:
 - Documentation creation.
 - Code SDKs.
 - Test Case creation.
 - Mocking and Stubbing.

APIs description languages

- Web Services Description Language (WSDL, commonly used in SOAP).
- API Blueprint.
- RAML (supports OpenAPI).
- Open Data Protocol (OData).
- **OpenAPI.**



OpenAPI - What is it?

- Description of HTTP-based APIs (commonly RESTful) that eases its usage and comprehension by the consumer (user-application).
- Comprised of a YAML file (API specification) that delineates inputs, outputs, and guidelines for accessing and consuming it.
- This specification can be manually created or automatically generated using a tool directly from the code itself.

OpenAPI

API Specification



- A self-contained resource that defines the elements of the API. It must include at least:
 - One path field.
 - One components or webhook field.
- In the specification, we can find:
 - Path Schemes: These specify different parameters (e.g., the number of questions in /questionservice/{})
 - Resource Types: Including text-plain, JSON, text + JSON, html + JSON...
 - HTTP Status Codes: Used to indicate results, such as 500 for errors or 200 for success

API Specification- Example

openapi: 3.0.0

info:

title: Gateway API

description: Gateway OpenApi specification.

version: 0.1.9

servers:

- url: http://localhost:8000
description: Development server

paths:

...

OpenAPI version

Information: name,
description and API version

Servers-Host on which the API
is made available

gatewayservice/openapi.yaml

```

openapi: 3.0.0
info:
  title: Gateway API
  description: Gateway OpenApi specification.
  version: 0.1.9
servers:
  - url: http://localhost:8000
    description: Development server
paths:
  /users:
    post:
      summary: Adds a new user to the database.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                username:
                  type: string
                  description: The user ID.
                password:
                  type: string
                  description: The user password.
      responses:
        '200':
          # status code
          description: OK
        '400':
          # status code
          description: The user cannot be added
          content:
            application/json:
              schema:
                type: object
                properties:
                  error:
                    type: string
                    description: Information about the error.

```

API Specification- Example

...
paths:
 /adduser:

Paths: one for each available endpoint

post:
 summary: Adds a new user to the database.

Type of CRUD operation

requestBody:
required: true
 content:

Request body specification:
 (if is mandatory): content,
 type and scheme

application/json:
 schema:
 type: object
 properties:

username:
type: string
description: The user ID.

password:
type: string
description: The user password.

Type and description of each attribute

responses:

```

gatewayservice/openapi.yaml

openapi: 3.0.0
info:
  title: Gateway API
  description: Gateway OpenApi specification.
  version: 0.1.9
servers:
  - url: http://localhost:8000

paths:
  /adduser:
    post:
      summary: Adds a new user to the database.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                username:
                  type: string
                  description: The user ID.
                password:
                  type: string
                  description: The user password.
      responses:
        '200':
          # status code
          description: OK
        '400':
          # status code
          description: The user cannot be added
          content:
            application/json:
              schema:
                type: object
                properties:
                  error:
                    type: string
                    description: Information about the error.
    
```



API Specification- Example

...

responses:

```
'200': # status code
  description: OK
'400': # status code
  description: The user cannot be added
  content:
    application/json:
      schema:
        type: object
        properties:
          error:
            type: string
            description: Information about the error.
```

Path outputs (responses):
descriptions, types, and
content with their schemes
for each

gatewayservice/openapi.yaml

```
openapi: 3.0.0
info:
  title: Gateway API
  description: Gateway OpenApi specification.
  version: 0.1.9
servers:
  - url: http://localhost:8000
    description: Development server
paths:
  /adduser:
    post:
      summary: Adds a new user to the database.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                username:
                  type: string
                  description: The user ID.
                password:
                  type: string
                  description: The user password.
      responses:
        '200': # status code
          description: OK
        '400': # status code
          description: The user cannot be added
          content:
            application/json:
              schema:
                type: object
                properties:
                  error:
                    type: string
                    description: Information about the error.
```



Good Practices

- Defining the specification before its implementation:
 - Enables us to focus on functionality and identify issues earlier.
 - Ease interoperability between different types of services.
- Synchronize/update the specification with the API changes [1]

[1] Alberto Martin-Lopez, Sergio Segura, and Antonio Ruiz-Cortés. FSE22. Online testing of RESTful APIs: promises and challenges.

Swagger



- Toolkit that ease the API specification deployment
- Enables:
 - Read the specification in a easy and clear way.
 - Check the different endpoints and hosts where the API is made available.
 - Available in the Gateway service (gatewayservice) of wiq_o:
<http://localhost:8000/api-doc>

The screenshot shows a web browser window with the URL "localhost:8000/api-doc/#". The page header features the Swagger logo and the text "Supported by SMARTBEAR". The main content area displays "Gateway API" with version tags "0.1.9" and "OAS 3.0". Below this, it says "Gateway OpenApi specification." There is a "Servers" section with a dropdown menu showing "http://localhost:8000 - Development server". Underneath, there is a "default" section with three API endpoints listed: a POST endpoint for "/adduser" (Adds a new user to the database.), a GET endpoint for "/health" (the Healthcheck of the service.), and a POST endpoint for "/login" (Logs the user into the system.). Each endpoint is represented by a colored box (green for POST, blue for GET) and a dropdown arrow on the right.