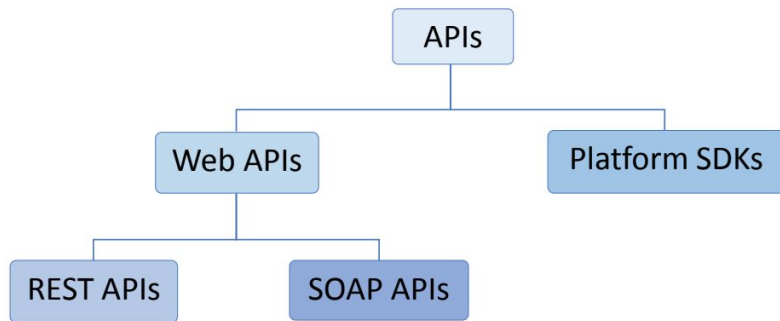


INGENIERÍA DE SOFTWARE APLICADA 2021

jeremias.castillo@um.edu.ar

REST & API FIRST APPROACH

API - Definición y Tipos



API – Application Programming Interface

“Colección de funciones de software que proveen un conjunto coherente de funcionalidad”

- **SDK**: específico para un lenguaje o plataforma (Ej: iOS SDK, SPRING)
- **WEB API**: independiente del lenguaje o plataforma, utilizado para aplicaciones web y funciona bajo el protocolo HTTP.
- **SOAP** – Simple Object Access Protocol: Protocolo de intercambio de mensajes estructurados en implementación de webservices. Utiliza XML.
- **REST – Representational State Transfer**: Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP (generalmente utiliza JSON)

REST - Arquitectura

- **Interfaz uniforme:**
 - La interfaz de basa en recursos (por ejemplo el recurso Empleado (Id, Nombre, Apellido, Puesto, Sueldo)
 - El servidor mandará los datos (vía html, json, xml...)
 - La representación del recurso que le llega al cliente, será suficiente para poder cambiar/borrar el recurso.
- **Peticiones sin estado.**
- **Cacheable.**
- **Separación de cliente y servidor:**
 - Separación de responsabilidades donde al cliente no interviene en como se obtienen los datos y al servidor no le interesa la interfaz de usuario.
- **Sistema de Capas**
 - El cliente puede estar conectado mediante la interfaz al servidor o a un intermediario, para el es irrelevante y desconocido.
 - El uso de capas o servidores intermedios puede servir para aumentar la escalabilidad (sistemas de balanceo de carga, cachés) o para implementar políticas de seguridad
- **Código bajo demanda (opcional) :**
 - Los servidores pueden ser capaces de aumentar o definir cierta funcionalidad en el cliente transfiriéndole cierta lógica que pueda ejecutar

REST - Recursos

- Los recursos son los **bloques fundamentales de construcción** en los sistemas web. Un recurso es cualquier cosa expuesta a la web (documentos, videos, dispositivos, etc)
- Las **características fundamentales** de los recursos son:
 - Un recurso puede ser una colección de objetos o un objeto individual
 - Un recurso es definido por su URI (Uniform Resource Identifier). Una URI identifica solo a un recurso pero un recurso puede estar en más de una URI.
 - Los recursos están manipulados mediante operaciones CRUD (Create, Read, Update, Delete), las cuales están mapeadas a verbos del protocolo HTTP.

Method	URI	Description
GET	/customers	List all customers in the collection.
POST	/customers	Create a new customer.
GET	/customers/{customerId}	Retrieve a customer specified by ID.
PUT	/customers/{customerId}	Update a customer specified by ID.
DELETE	/customers/{customerId}	Delete a customer specified by ID.

REST - API endpoints

- Request:

GET http://api.ecommerce.com/v1/customers/{customerId}/orders?year=2017

↑
HTTP Method

↑
Base URI

↑
Resource URI

↑
Path Parameter

↑
Query Parameter

Accept: application/json

↑
Request Header

↑
Media Type

- Response:

Status: 200 OK

↑
HTTP Status Code

Content Type: application/json

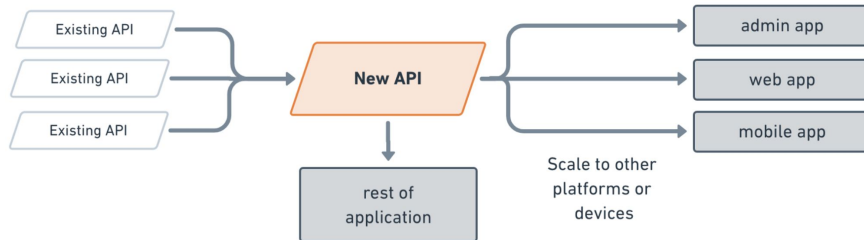
↑
Response Header

↑
Media Type

Response Body →

```
[{
  "orderId" : "132",
  "customerId" : "86597",
  "productName" : "REST_API_Textbook_v1",
  "productDescription" : "REST API Textbook 1-st edition",
  "priceTotal" : 29.99,
  "placedAt" : "2017-01-10T23:43:33.741Z",
  "completed" : true,
  "completedAt" : "2017-01-11T05:11:00.562Z"
}, {
  "orderId" : "134",
  "customerId" : "86597",
  "productName" : "Swagger_Textbook_v2",
  "productDescription" : "Swagger and OpenAPI Textbook 2-nd edition",
  "priceTotal" : 18.99,
  "placedAt" : "2017-03-05T23:43:33.741Z",
  "completed" : true,
  "completedAt" : "2017-03-06T05:11:00.562Z"
}]
```

API FIRST - ¿Qué significa?



- Se diferencia de los enfoques “**Consumer First / Code First**” donde se programa la lógica de negocios, UI y Backend para luego montar una API.
- En el enfoque **API First** la definición de la API es lo primordial y se realiza antes de escribir cualquier código.
- Esta metodología es muy útil cuando muchas apps, de **diversos tipos y orígenes**, deben interactuar con la lógica que presenta la API.

API FIRST - Tareas



Design

Design and model APIs according to specification-based standards



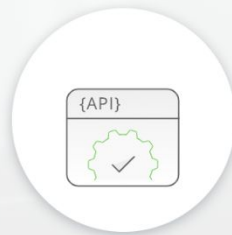
Build

Build stable, reusable code for your API in almost any language



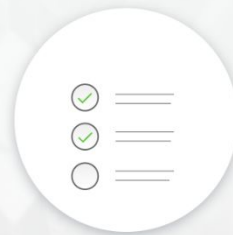
Document

Improve developer experience with interactive API documentation



Test

Perform simple functional tests on your APIs without overhead



Standardize

Set and enforce API style guidelines across your API architecture

El diseño de una API implica su especificación y correspondiente documentación.

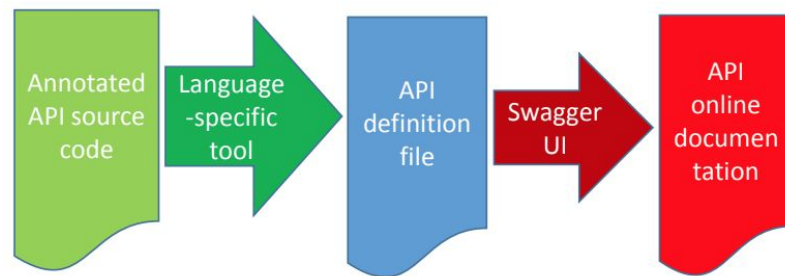
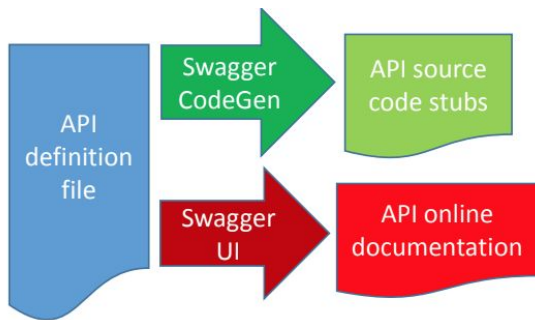
API FIRST - Especificación y Documentación

Para poder especificar una api existen diferentes formas de realizarlo:
RAML, API Blueprint, OpenAPI/Swagger.



1. Diseñar la API y especificarla en un archivo.
2. Generar el código fuente (stubs) desde el archivo de especificación.
3. Generar la documentación online

1. Diseñar la API y escribir código.
2. Agregar anotaciones (@) al código fuente.
3. Generar el archivo de definición de la API desde las anotaciones.
4. Generar la documentación online



API FIRST - Swagger Example OAS3

The screenshot displays the Swagger Editor interface, which is used for creating and editing OpenAPI specifications. The left pane shows the raw JSON specification, and the right pane provides a visual overview of the API.

Swagger Editor (Supported by SMARTBEAR)

File ▾ Edit ▾ Insert ▾ Generate Server ▾ Generate Client ▾

```
1 openapi: 3.0.0
2 servers:
3   # Added by API Auto Mocking Plugin
4   - description: SwaggerHub API Auto Mocking
5     url: https://virtserver.swaggerhub.com/isa-um/blog/1.0.0
6 info:
7   description: This is a simple API
8   version: "1.0.0"
9   title: blog API
10  contact:
11    email: you@your-company.com
12  license:
13    name: Apache 2.0
14    url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
15 tags:
16   - name: admins
17     description: Secured Admin-only calls
18   - name: developers
19     description: Operations available to regular developers
20 paths:
21   /trend:
22     get:
23       tags:
24         - developers
25       summary: searches trends
26       operationId: searchInventory
27       description: |
28         By passing in the appropriate options, you can search for
29         available trends in the system
30       parameters:
31         - in: query
32           name: range
33           description: number of days to calculate trend
34           schema:
35             type: integer
```

blog API 1.0.0 OAS3

This is a simple API

[Contact the developer](#)
[Apache 2.0](#)

Servers

https://virtserver.swaggerhub.com/isa-um/blog/1.0.0 - SwaggerHub API Auto Mocking ▾

admins Secured Admin-only calls ^

developers Operations available to regular developers ^

GET /trend searches trends ▾

Schemas ^

<https://editor.swagger.io/>