Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

APIs y servicios web

IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R.
2do Semestre 2020

API

Application Programming Interface

API

¿Qué es?

Es una interfaz o intermediario entre una aplicación o software y quien desee interactuar con la aplicación.

- **Se dice que una aplicación “expone” funcionalidades mediante una API**
- **Quien desee acceder a estas funcionalidades tiene que “consumir” la API**

API

¿Qué es?

Establecer comunicación entre aplicaciones, ya que quien consume la API puede perfectamente ser otra aplicación

```
document.querySelector('.some-class')
```

```
GET https://api.github.com/orgs/IIC2513-2020-2/repos
```

API Web

API que utiliza HTTP

API Web

¿Qué es?

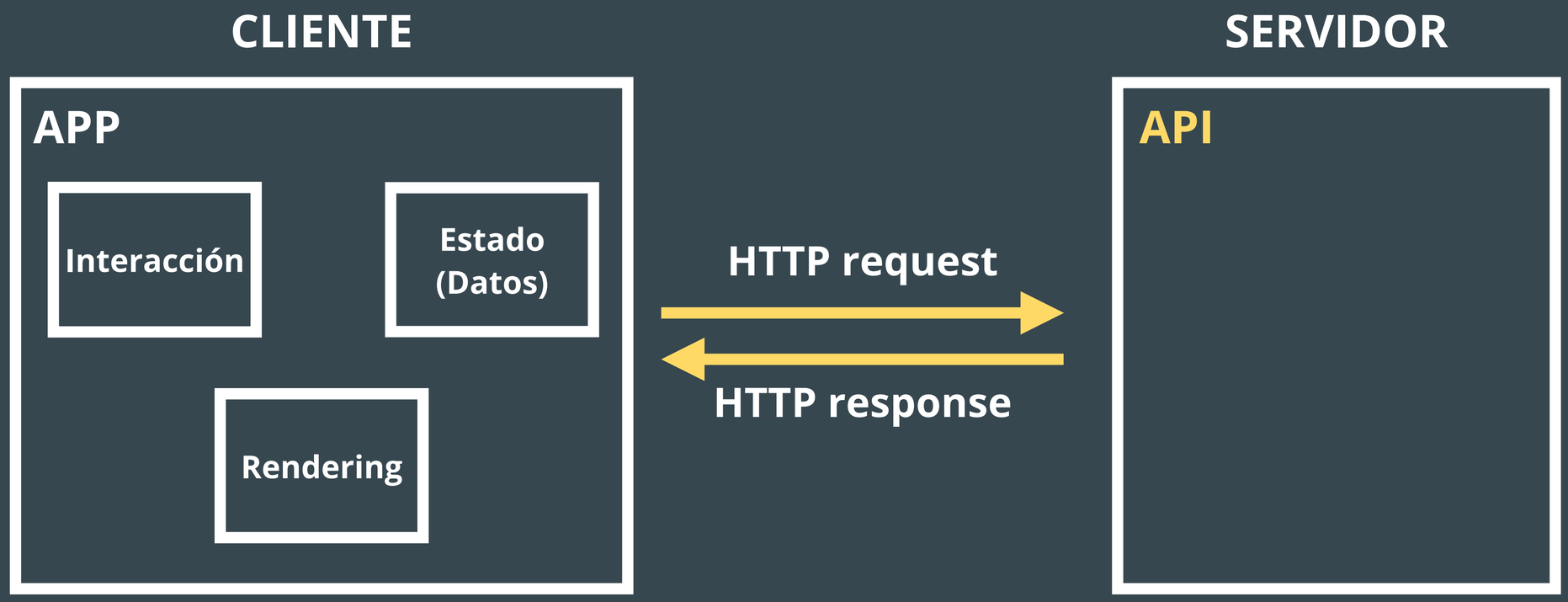
Es una API que se accede utilizando la infraestructura Web, es decir, el protocolo HTTP

- En su forma más genérica, el cliente hace un request y recibe datos como respuesta
- No necesariamente un documento HTML

API Web

¿Dónde está inserta?

Las URLs suelen llamarse endpoints



API Web

Características

Se conocen de forma genérica como "web services"

- Las respuestas contienen datos estructurados que son presentados en una manera estándar. Posibles de ser interpretados por diferentes tipos de aplicaciones
- Los formatos más utilizados para una API son JSON y XML

Arquitecturas WS

Cómo especificar un servicio web

Arquitecturas web services

¿Qué es?

Existen diferentes arquitecturas o estilos para especificar un servicio web

Lo que tienen en común:

- **Scope (qué vamos a consultar)**
- **Acción/operación (cómo vamos a consultar)**
- **Datos (con qué vamos a consultar)**

Arquitectura RPC

Remote Procedure Call

Arquitectura RPC

¿Qué es?

Llamar a un procedimiento (función) de forma remota

El medio utilizado es internet (puede ser HTTP o incluso SMTP)

- **Analogía: llamar a una función en JavaScript**
- **Necesitamos nombre del procedimiento y argumentos**

XML-RPC

Ejemplo

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodCall>
   <methodName>sample.sum</methodName>
   <params>
      <param>
         <value><int>17</int></value>
      </param>

      <param>
         <value><int>13</int></value>
      </param>
   </params>
</methodCall>
```

Fuente: TutorialsPoint

SOAP

Ejemplo

```
POST /Quotation HTTP/1.0
Host: www.xyz.org
Content-Type: text/xml; charset = utf-8
Content-Length: nnn

<?xml version = "1.0"?>
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-envelope"
   SOAP-ENV:encodingStyle = "http://www.w3.org/2001/12/soap-encoding">

   <SOAP-ENV:Body xmlns:m = "http://www.xyz.org/quotations">
      <m:GetQuotation>
         <m:QuotationsName>MiscroSoft</m:QuotationsName>
      </m:GetQuotation>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fuente: TutorialsPoint

JSON-RPC

Ejemplo

```
POST / HTTP/1.1
HOST: api.example.com
Content-Type: application/json

{“jsonrpc": "2.0", "method": "subtract", "params": {"subtrahend": 23,
"minuend": 42}, "id": 3}
```

Fuente: JSON-RPC Spec

Arquitectura REST

REpresentational **S**tate **T**ransfer

Arquitectura REST

¿Qué es?

Estilo arquitectónico que define una **serie de principios** (o restricciones) que debe cumplir un **servicio web**

Disertación publicada por **Roy Fielding** en el año 2000 para conseguir grado de PhD

Arquitectura REST

Características

Utiliza los conceptos de recursos y representaciones de estos recursos

Se apoya fuertemente en los métodos HTTP para especificar acciones

Arquitectura REST

Principios

- Client-server
- Stateless
- Cacheable
- Uniform interface
  - Resource-based
  - Manipulation of Resources Through Representations
  - Self-descriptive Messages
  - Hypermedia as the Engine of Application State (HATEOAS)
- Layered system
- Code on Demand (opcional)

RESTful WS

¿Cómo se ven?

RESTful web services

¿Cómo se ven?

REST se apoya fuertemente en HTTP

- Scope en **path**
- Operación en **método HTTP**
- Datos en **body**

RESTful web services

¿Cómo se ven?

```
GET     /users
POST    /users
GET     /users/2
PUT     /users/2
PATCH   /users/2
DELETE  /users/2
```

Ejemplo RESTful WS

Github API

Referencias

- Perry Eising (Medium) - "What exactly IS an API?"
- W3C - "Web Services overview - Design Issues"
- Phil Sturgeon (Smashing Magazine) - "Understanding RPC Vs REST For HTTP APIs"
- Roy Fielding - “Architectural Styles and the Design of Network-based Software Architectures”
- REST API Tutorial - “What Is REST?”