

JAVASCRIPT

REST



¿Qué es una API?

2

- **Definición:** Application Programming Interfaces, o *interfaz de programación de aplicaciones*.
- Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, **permitiendo la comunicación entre dos aplicaciones** de software a través de un conjunto de reglas.



REST. ¿Qué es? (I)

3

- ¿Qué es una API (o servicio, o *web service*) REST?
- Un servicio REST no tiene estado (es *stateless*), lo que quiere decir que, entre dos llamadas cualesquiera, el servicio pierde todos sus datos.
- Para mantener un estado se requiere algún sitio (generalmente memoria) donde guardar todos los estados de todos los clientes.



REST. ¿Qué es? (II)

4

- A más clientes, más memoria, hasta que al final podemos llegar a no poder admitir más clientes, no por falta de CPU, sino de memoria.
- Sea cual sea la tecnología con la que desarrolles para web, seguro que conoces que puedes crear lo que se llama una “sesión”.
- Los datos de la sesión se mantienen entre peticiones y son por cada usuario.
- El clásico ejemplo de sesión puede ser un carrito de la compra, entre las distintas peticiones web, la información del carrito de compra se mantiene.



REST. ¿Qué es? (III)

5

- Es algo idóneo para atender muchas peticiones de datos.
- Los recursos de gasto en el servidor son mínimos.
- La usabilidad de la app es mucho mayor.



REST. Arquitectura (I).

6



REST. Arquitectura (II).

7

- El cliente inicia siempre una solicitud.
- Nuestro lenguaje del lado del cliente es el que solicitará al servidor un recurso al servidor.
- El servidor y el cliente web se comunican en un formato de intercambio de información, habitualmente JSON.
- Lo importante es que el cliente no recibe HTML para renderizar, sino simplemente los datos que se han generado como respuesta.
- El servidor no escribe HTML, sino únicamente genera los datos para enviarlos al cliente.



REST. Ventajas (I)

8

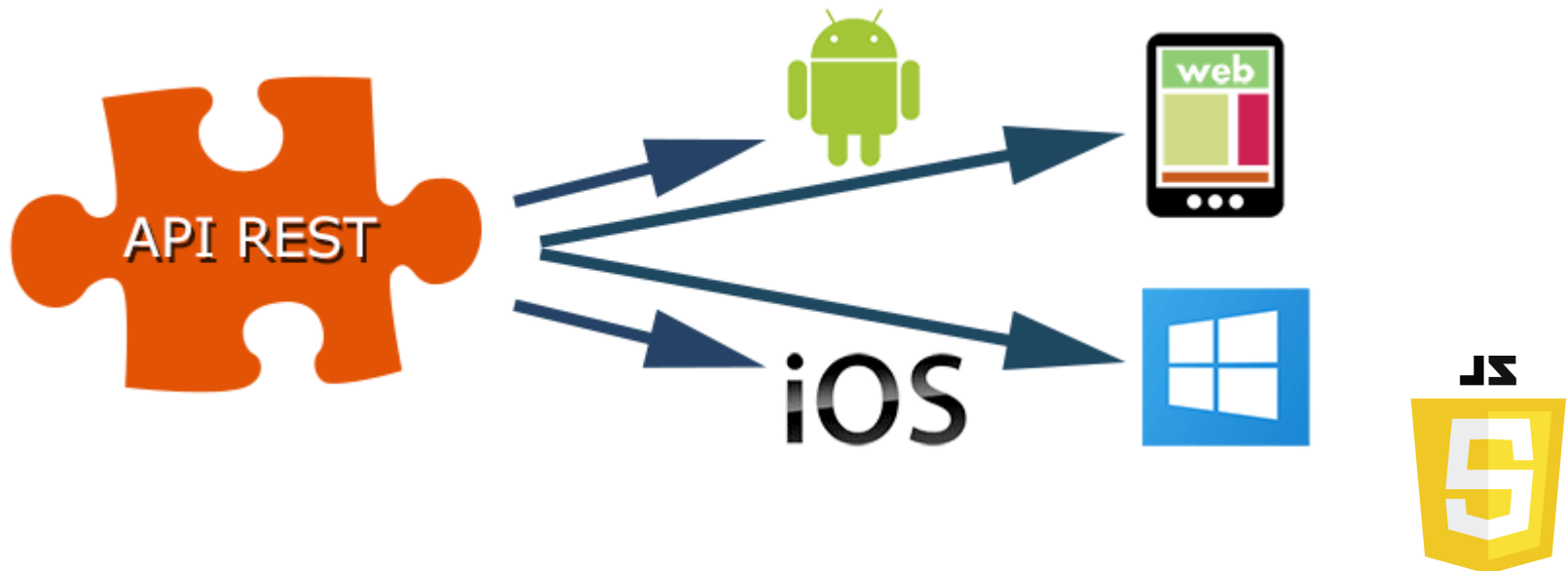
- Al ser sistemas independientes (solo se comunican con un lenguaje de intercambio como JSON) puedes desarrollarlos proyectos autónomos, equipos autónomos.
- Al cliente le da igual cómo está hecha la API y al servidor le da igual qué vas a hacer con los datos que te ha proporcionado.



REST. Ventajas (II)

9

- ❑ Multiplataforma.
- ❑ Independencia de lenguajes y sistemas.
- ❑ Tendríamos un recurso válido para cualquier otro tipo de app.



REST. Ventajas (III)

10

- ❑ Escalabilidad. La app puede crecer todo lo que necesite en cada momento.
- ❑ La API puede responder a otros tipos de operaciones o puede versionarse tanto como se desee.
- ❑ También la programación del lado del cliente puede crecer todo lo necesario con el tiempo, incluso, crear otros frontales, no solo web, también de Apps para cualquier dispositivo.



REST. Recursos necesarios.

11

- Un servidor.
- Una aplicación de gestión de la API en el lado del servidor.
- Una base de datos.
- Uno o varios clientes si se quiere disponer de ella para desarrollar apps.



JSON

12

- ❑ JSON (acrónimo de JavaScript Object Notation), es un formato de texto sencillo para el intercambio de datos.
- ❑ Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.
- ❑ Hoy en día es el lenguaje de intercambio de datos estándar cliente/servidor.

JSON: Sintaxis

13

- Los tipos de datos disponibles en JSON son:
 - ▣ **Números:** Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos. Ejemplo: 123.456
 - ▣ **Cadenas:** Representan secuencias de cero o más caracteres. Se ponen entre doble comilla. Ejemplo: "Hola"
 - ▣ **Booleanos:** Representan valores booleanos y pueden tener dos valores: true y false
 - ▣ **null:** Representan el valor nulo.
 - ▣ **Array:** Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes. Ejemplo ["juan","pedro","jacinto"]
 - ▣ **Objetos:** Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena entre comillas dobles. El valor puede ser de cualquier tipo.

JSON: Ejemplo

14

```
{
  "departamento":8,
  "nombredepto":"Ventas",
  "director": "Juan Rodríguez",
  "empleados":[
    {
      "nombre":"Pedro",
      "apellido":"Fernández"
    },
    {
      "nombre":"Jacinto",
      "apellido":"Benavente"
    }
  ]
}
```