

JSON

La notación de objeto Javascript

Intercambio de datos



El mundo está conectado

Intercambio de datos

HOLA

OHAYO



HELLO

OLÁ

El mundo está conectado

Intercambio de datos



El mundo está conectado

Intercambio de datos



El mundo está conectado

Características deseables intercambio de datos

- Independencia de Sistema Operativo
- Independencia del lenguaje de programación
- Facilidad de transmisión por redes
- ...

Ficheros de texto plano

Intercambio de datos

<XML>

{ JSON }

Lenguajes para intercambio de datos

{JSON}

(JavaScript Object Notation)

<http://json.org/json-es.html>

Ejemplo JSON

```
{
  "bebida": "Cerveza",
  "inicioProduccion": "3500 a.c.",
  "ingredientes": ["lupulo", "cebada", "agua", "trigo", "malta"],
  "marcas": [
    {
      "nombre": "San Miguel",
      "tipo": "pilsen",
      "procedencia": "España"
    }, {
      "nombre": "Kölsch",
      "tipo": "Estilo Alemán",
      "procedencia": "Colonia"
    }, {
      "nombre": "Guinness",
      "tipo": "Negra",
      "procedencia": "Irlanda"
    }
  ]
}
```

```
<bebidas>
  <bebida>Cerveza</bebida>
  <inicioProduccion>3500 a.c.</inicioProduccion>
  <ingredientes>
    <ingrediente>lupulo</ingrediente>
    <ingrediente>cebada</ingrediente>
    <ingrediente>agua</ingrediente>
    <ingrediente>trigo</ingrediente>
    <ingrediente>malta</ingrediente>
  </ingredientes>
  <marcas>
    <marca>
      <nombre>San Miguel</nombre>
      <tipo>pilsen</tipo>
      <procedencia>España</procedencia>
    </marca>
    [... NO ME CABEN MÁS MARCAS ...]
  </marcas>
</bebidas>
```

```
{
  "bebida": "Cerveza",
  "inicioProduccion": "3500 a.c.",
  "ingredientes": ["lupulo", "cebada", "agua", "trigo",
    "malta"],
  "marcas": [
    {
      "nombre": "San Miguel",
      "tipo": "pilsen",
      "procedencia": "España"
    }, {
      "nombre": "Kölsch",
      "tipo": "Estilo Alemán",
      "procedencia": "Colonia"
    }, {
      "nombre": "Guinness",
      "tipo": "Negra",
      "procedencia": "Irlanda"
    }
  ]
}
```


Algunas características

XML

- Permite intercambio entre distintos sistemas
- Sencillo de leer por las personas
- Abierto
- Extensible
- Da importancia al significado
- Simplificado
- Soporta muchos lenguajes

JSON

- Permite intercambio entre distintos sistemas
- Es posible leerlo por humanos, aunque menos fácil
- Abierto
- No es extensible
- Da importancia a los datos
- Aún más simple
- Soporte nativo

Ventajas comparativas entre XML y JSON

XML

- Variaciones del XML para diferentes usos. Varios lenguajes.
- Lectura para humanos
- Existen reglas de validación de documentos
- Posiblemente mayor soporte

JSON

- Más ligero en la mayoría de los casos
- Permite implementar cualquier tipo de datos
- Más ligero de implementar
- Es código Javascript!
- Amplio soporte para Ajax en la mayoría de frameworks

¿Para qué sirve JSON?

Para intercambio de datos... ¿no?

¿Para qué sirve JSON?

Para intercambio de datos... ¿no?

- Servicios web (con API REST)
- Separar el Frontend del backend

Servicios web

Tecnología que utiliza un conjunto de protocolos y estándares que sirven **para intercambiar datos entre aplicaciones...**

http://es.wikipedia.org/wiki/Servicio_web

API REST

REST, **RE**presentational **S**tate **T**ransfer, es un tipo de **arquitectura de desarrollo web** que se apoya totalmente en el estándar HTTP.

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

Por lo tanto REST es el tipo de arquitectura más **natural y estándar para crear APIs para servicios orientados a Internet**.

http://es.wikipedia.org/wiki/Representational_State_Transfer

<http://www.desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html>

API REST

Hay cientos de APIs en Internet que te devuelven los datos en JSON.

- **Twitter**
- **Youtube**
- **Facebook**
- **Flickr**
- ...

Comunicaciones

Backend --> Frontend

JSON desempeña un papel fundamental en la arquitectura de las aplicaciones web modernas.

Front-end

HTML

CSS

JS

Back-end

PHP

NodeJS

MySQL

Frontend - Backend

Comunicaciones Cliente ---> Servidor

Tienes los mecanismos, **POST**, **GET** para enviar datos del cliente (navegador) al servidor.

Nada te impide enviar datos en JSON pero los vas a tener que meter en la solicitud HTTP, generalmente por POST. Por ello lo común es enviar los datos en pares clave valor usando los mecanismos habituales de POST.

Si te envías un JSON al servidor, tendrás que convertir los datos a objetos nativos PHP, usando **json_decode()** como ya se explicó.

Comunicaciones Servidor ---> Cliente

Más útil es devolver los datos en JSON desde PHP al navegador, porque así la respuesta de tu aplicación es independiente del "front".

En aplicaciones "avanzadas" se evita devolver datos en HTML, porque estás forzado a utilizar los datos de la manera como fueron devueltos.

Con Javascript recibes los datos de la aplicación y haces lo que necesites. Incluso en el JSON puedes enviar datos de varios tipos para usar de maneras distintas.

Comunicaciones Servidor ---> Cliente



Argentina



Chile



Colombia

Seleccione...

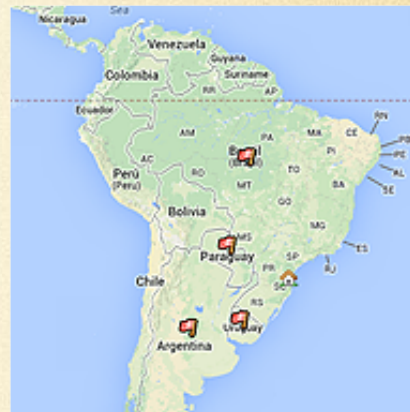
España

Argentina

Colombia

Francia

- Albania
- Alemania
- Andorra
- Angola



Puedo hacer uso de países en el "front" de una aplicación web para distintas cosas. ¿Son vistas distintas? o recibo un JSON con los países y en el front me encargo de procesarlo y convertirlo en la salida que me interesa.

Backend → frontend web



1 Backend → varios frontend

