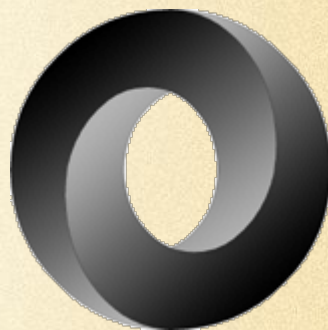


JSON

Para el desarrollo de aplicaciones web con PHP

JSON



- 1) Aprender a usar JSON en PHP (y Javascript)
- 2) Incorporar ejemplos con JSON en el flujo de trabajo del MVC



#phpPRO



"Conociendo y usando Jota-son en PHP en @EscuelaIT / @deswebcom"

JSON

- Qué es JSON
- Intercambio de datos
 - a. Servicios web con API REST
 - b. Comunicaciones Backend ---> Frontend

¿Qué es JSON?

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un **formato ligero de intercambio de datos**.

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

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

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"
    }
  ]
}
```

XML Equivalente al JSON de ejemplo

```
<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"
    }
  ]
}
```


Ventajas comparativas 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 JS

Ejemplo

Producir JSON desde PHP, a partir de:

1. Array
2. Array asociativo
3. Objeto

¿Para qué sirve JSON?

Para intercambio de datos... ¿no?

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

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**
- ...

Ejemplo

Consumir JSON desde PHP

Aprendamos lo que es un API REST y cómo alimentar una aplicación PHP por medio de datos que recibimos por JSON.

<http://openweathermap.org/api>

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

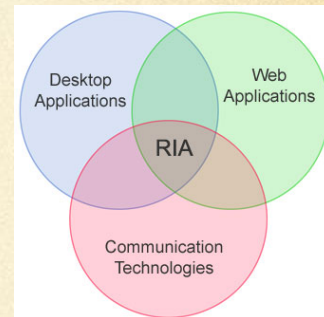
Frontend - Backend

Refrescando conceptos...

Antes de continuar veamos un par de conceptos y técnicas de desarrollo en aplicaciones modernas.

RIA Internet Rich Applications

Aplicaciones de Internet enriquecidas
Aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.



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

Ajax

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas [...] se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible **realizar cambios sobre las páginas sin necesidad de recargarlas**, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

<http://es.wikipedia.org/wiki/AJAX>

¿Ajax es PHP?



Ejemplo

Aprendamos a hacer Ajax con jQuery.

- Ajax simple con load()
- Enviar un formulario por Ajax y a mostrar la respuesta que nos envía PHP después de procesarlo.

<http://api.jquery.com/>

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... ▼

Seleccione...

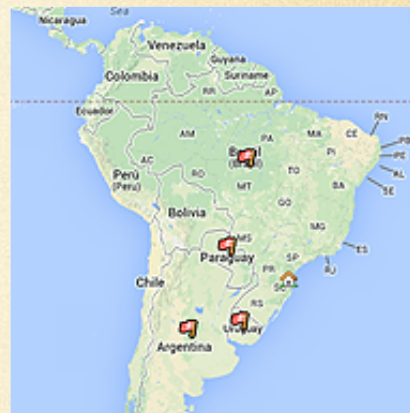
España

Argentina

Colombia

Francia

- Albania
- Alemania
- Andorra
- Angola

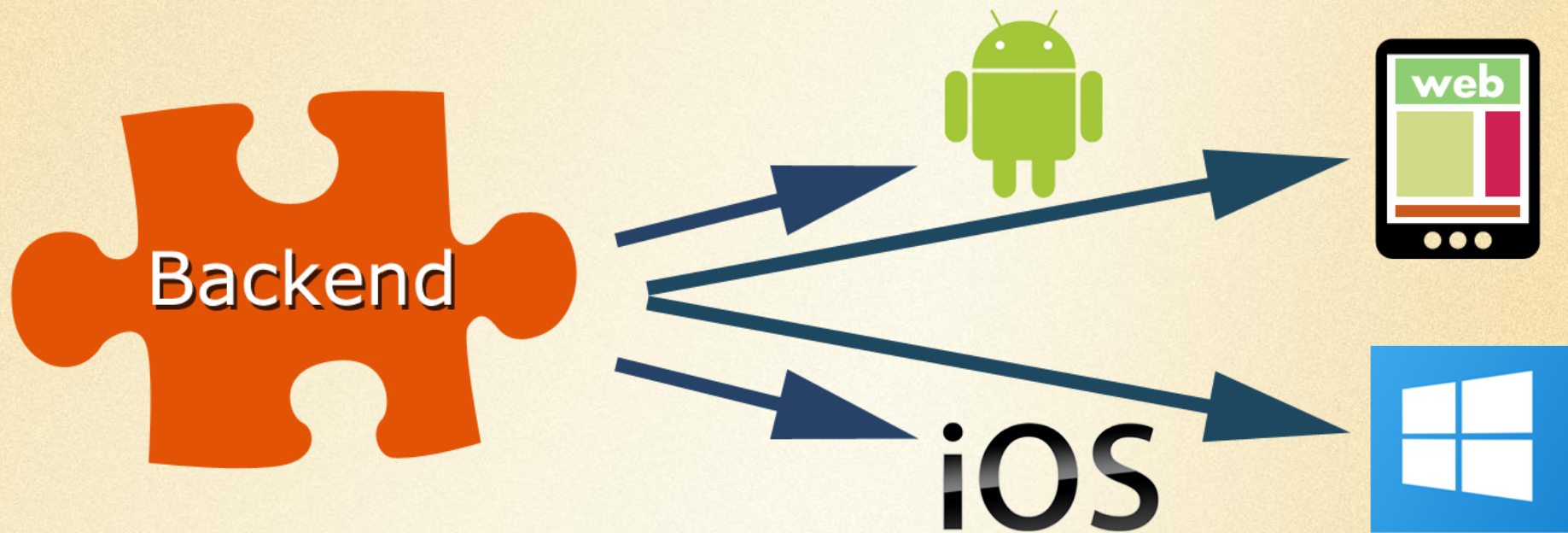


Puedo hacer uso de países en el "front" de una aplicación web para distintas cosas. ¿Son vistas distintas? o recibo un JSON con los países y en el front me encargo de procesarlo y convertirlo en la salida que me interesa.

Backend → frontend web



1 Backend → varios frontend



Ejemplo

Aprendamos a generar un dato JSON desde PHP y enviarlo al cliente para procesarlo.

<http://jqueryui.com/autocomplete/>