

Servicios Telemáticos Avanzados

4.- Representación de datos: XML y JSON

OpenCourseWare 2014

Maidier Huarte y Gorka Prieto
Escuela Técnica Superior de Ingeniería de Bilbao
Departamento de Ingeniería de Comunicaciones
Universidad del País Vasco (UPV/EHU)

Servicios Telemáticos Avanzados:

4.- XML y JSON.odp



Copyright © 2014 Maider Huarte Arrayago, Gorka Prieto Agujeta

Servicios Telemáticos Avanzados: 4.- XML y JSON.odp lana, Maider Huartek eta Gorka Prietok egina, Creative Commons-en Attribution-NonCommercial-Share Alike 4.0 International License baimenaren menpe dago. Baimen horren kopia bat ikusteko, <http://creativecommons.org/licenses/by-nc-sa/4.0/> webgunea bisitatu edo gutun bat bidali ondoko helbidera: Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

Servicios Telemáticos Avanzados: 4.- XML y JSON.odp by Maider Huarte and Gorka Prieto is licensed under a Creative Commons Attribution-NonCommercial-Share Alike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or, send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

XML y JSON

ÍNDICE

1.- Introducción

2.- XML

2.1.- Introducción

2.2.- Sintaxis

2.3.- Validación

2.4.- JAXB

3.- JSON

3.1.- Introducción

3.2.- Sintaxis

3.3.- AJAX

1.- Introducción

- Transporte de datos entre aplicaciones
 - HTTP
 - HTML
 - XML
 - JSON

2.- XML

2.1.- Introducción

- ¿Qué es XML?
 -
 -
 -
 -
- Extensión de ficheros:
- Tipo MIME:
- Ejemplos de utilización
 - Server-Side Scripting
 - Client-Side Scripting: **AJAX**

2.- XML

2.1.- Introducción

- Tecnologías basadas en XML
 - xhtml:
 - Utilizado en JavaEE
 - Versión más estricta de html, basada en xml
 - XSD: xml Schema, alternativa a DTD
 - SOAP
 - WSDL
 - ...

2.- XML

2.2.- Sintaxis

- Documentos xml
 - Texto simple
 - Declaración xml
 - Elementos estructurados: etiquetas
 - No etiquetas predefinidas
 - ▶ Definición libre de etiquetas
 - ▶ Definición libre de estructura

Ejemplo XML

2.- XML

2.2.- Sintaxis

- Sintaxis xml
 - Comentarios: `<!-- comentario de varias lineas -->`
 - Etiquetas: apertura y cierre
 - Declaración xml
 - Nombres
 - Anidamiento
 - Elemento raíz
 - Atributos de elementos: `nombre="valor"`

Ejemplo XML

2.- XML

2.2.- Sintaxis

- Sintaxis xml
 - Caracteres especiales

Carácter	Sustitución	Razón
<	<	Delimitación de etiquetas
>	>	Delimitación de etiquetas
“	"	Valores de atributos
'	'	Valores de atributos
&	&	Sustitución de caracteres conflictivos

2.- XML

2.2.- Sintaxis

• Ejemplos: Ficheros xml básicos

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesion SYSTEM "Sesion.dtd">
<sesion idSesion="1" idUsuario="25">
  <operacion>
    <op1>12</op1>
    <op>-</op>
    <op2>2</op2>
    <res>10</res>
  </operacion>
  <operacion>
    <op1>34</op1>
    <op>+</op>
    <op2>5</op2>
    <res>39</res>
  </operacion>
</sesion>
```

SesionMatematicas1.java

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesion SYSTEM "froga3.dtd">
<sesion>
  <idSesion>1</idSesion>
  <idUsuario>25</idUsuario>
  <operacion>
    <op1>12</op1>
    <op>-</op>
    <op2>2</op2>
    <res>10</res>
  </operacion>
  <operacion>
    <op1>34</op1>
    <op>+</op>
    <op2>5</op2>
    <res>39</res>
  </operacion>
</sesion>
```

SesionMatematicas2.java

[Volver a 6](#)

[Volver a 7](#)

[Volver a 10](#)

2.- XML

2.2.- Sintaxis

- Nomenclatura de etiquetas
 - Letras, números, etc
 - Números, caracteres de puntuación
 - Espacios
 - “xml”, “Xml”, etc
- Recomendaciones para nomenclatura
 - Nombres descriptivos: ' _ '
 - Cortos y simples
 - No utilizar:
 - No utilizar:

[Ejemplo XML](#)

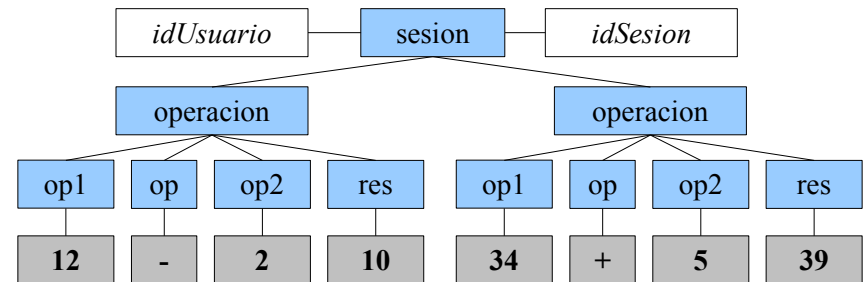
2.- XML

2.2.- Sintaxis

- Texto interno vs atributos
 - Varios valores
 - Valores estructurados
 - Extensible
 - Identificación de elementos
 - Combinación
- XML y DOM
 - Elemento raíz: árbol
 - Hijos, padres
 - Hermanos

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesion SYSTEM "Sesion.dtd">
<sesion idSesion="1" idUsuario="25">
  <operacion>
    <op1>12</op1>
    <op>-</op>
    <op2>2</op2>
    <res>10</res>
  </operacion>
  <operacion>
    <op1>34</op1>
    <op>+</op>
    <op2>5</op2>
    <res>39</res>
  </operacion>
</sesion>
```

SesionMatematicas1.java



2.- XML

2.2.- Sintaxis

- XML Namespaces
 - Conflictos de nomenclatura
 - Declaración: **xmlns[:prefijo]="URL"**
 - Prefijos
 - URL
 - Ubicación de declaraciones xmlns
 -
 -

2.- XML

2.2.- Sintaxis

- Ejemplo: NameSpaces

```
<root xmlns:h="http://www.w3.org/TR/html4/" xmlns:f="http://www.w3schools.com/furniture">
<h:table>
  <h:tr>
    <h:td>Manzanas</h:td>
    <h:td>Peras</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>Mesa de cocina</f.name>
  <f:width>80</f.width>
  <f:length>120</f.length>
</f.table>

</root>
```

EjemploTable.xml

```
<table xmlns="http://www.w3.org/TR/html4/">
<tr>
  <td>Manzanas</td>
  <td>Peras</td>
</tr>
</table>
```

Tabla.xml

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>Mesa de cocina</name>
  <width>80</width>
  <length>120</length>
</table>
```

Mesa.xml

2.- XML

2.3.- Validación

- Bien formado vs válido
 - Bien formado
 - Válido
 - Patrón de validación
 - ▶ DTD
 - ▶ XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesion SYSTEM "froga3.dtd">
<sesion>
  <idSesion>1</idSesion>
  <idUsuario>25</idUsuario>
  <operacion>
    <op1>12</op1>
    <op>-</op>
    <op2>2</op2>
    <res>10</res>
  </operacion>
  <operacion>
    <op1>34</op1>
    <op>+</op>
    <op2>5</op2>
    <res>39</res>
  </operacion>
</sesion>
```

SesionMatematicas2.java

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="T3-5_3" transaction-type="JTA">
    <jta-data-source>java:jboss/datasources/MySQLDS</jta-data-source>
  <!-- Clases Entity -->
  <properties>
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
    <property name="hibernate.hbm2ddl.auto" value="update" />
  </properties>
</persistence-unit>
</persistence>
```

persistence.java

2.- XML

2.4.- JAXB

- Java Architecture for XML Binding (JAXB)
 - Permite mapear clases Java a XML
 - Anotaciones+marshallers
 - Incluido a partir de Java6
- Proceso
 - Anotar beans
 - @XmlRootElement, @XmlElement
 - Admiten nombre: @XmlElement(name="item")
 - Obtener contexto JAXB
 - JAXBContext.newInstance(Item.class)

2.- XML

2.4.- JAXB

- Proceso (*continuación*)
 - Obtener marshallers
 - .createMarshaller()
 - .createUnmarshaller()
 - Serializar/Deserializar
 - .marshal(object, dest)
 - .unmarshal(src)

2.- XML

2.4.- JAXB

```
@XmlRootElement
public class ItemBean {
    private long ref;
    private String name;

    @XmlElement
    public long getRef() {
        return ref;
    }

    public void setRef(long ref) {
        this.ref = ref;
    }

    @XmlElement
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

ItemBean.java

2.- XML

2.4.- JAXB

```
@XmlRootElement
public class ItemList {
    private List<ItemBean> items;

    @XmlElement(name = "item")
    public List<ItemBean> getItems() {
        return items;
    }

    public void setItems(List<ItemBean> items) {
        this.items = items;
    }

    public static void save( ItemList itemList, String path ) throws IOException, JAXBException {
        JAXBContext jaxbContext = JAXBContext.newInstance(ItemList.class);
        Marshaller jaxbMarshaller = jaxbContext.createMarshaller();
        jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
        jaxbMarshaller.marshal(itemList, new File(path));
    }
}
```

ItemList.java

3.- JSON

3.1.- Introducción

- ¿Qué es JSON?
 -
 -
- JSON vs XML
 - Parecidos
 - Diferencias
- Extensión de ficheros:
- Tipo MIME:

3.- JSON

3.2.- Sintaxis

- Documentos JSON: texto simple
- Sintaxis
 - Datos: nombre, valor
 - Separación entre datos:
 - Objetos:
 - Arrays
- Declaración de datos
 - “*nombre*”:”*valor*”
 - Valores

3.- JSON

3.2.- Sintaxis

• Ejemplo: Sintaxis JSON

```
{
  "sesion" :
  {
    "idSesion" : "1",
    "idUsuario" : "25",
    "operacion" :
    [
      {"op1":"12", "op":"-", "op2":"2", "res":"10"},
      {"op1":"34", "op":"+", "op2":"5", "res":"39"}
    ]
  }
}
```

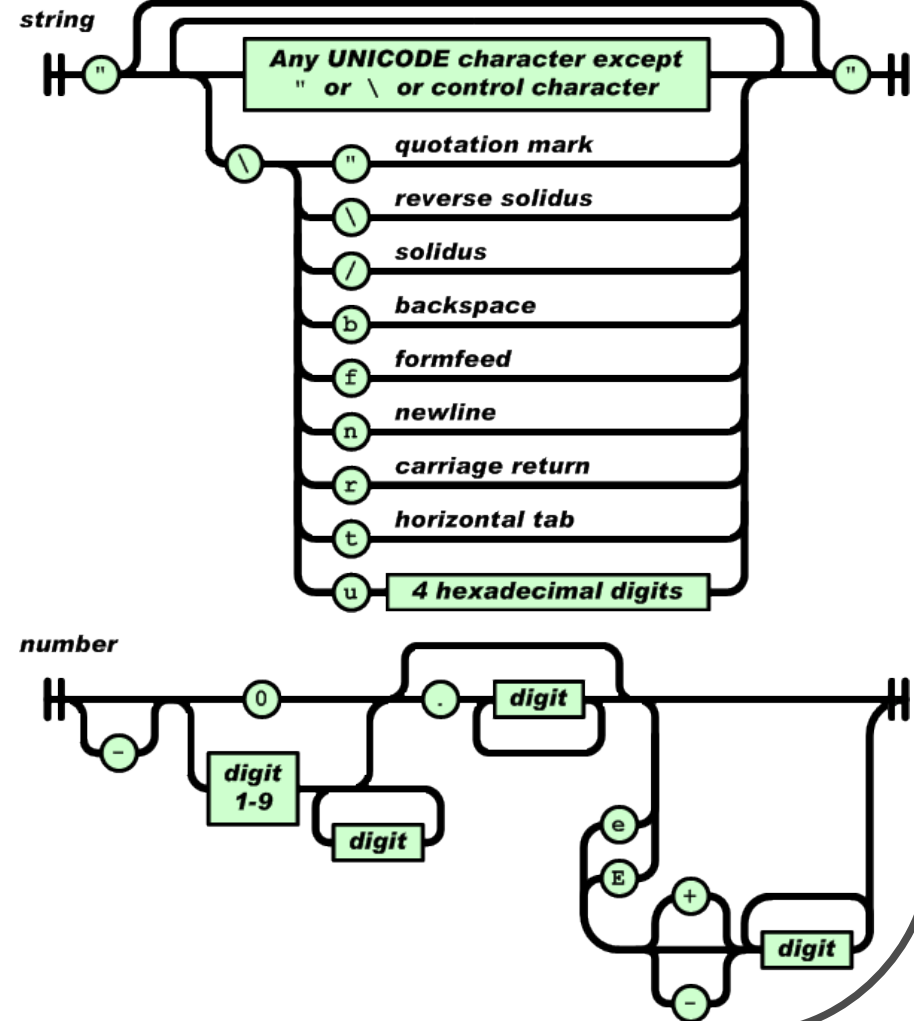
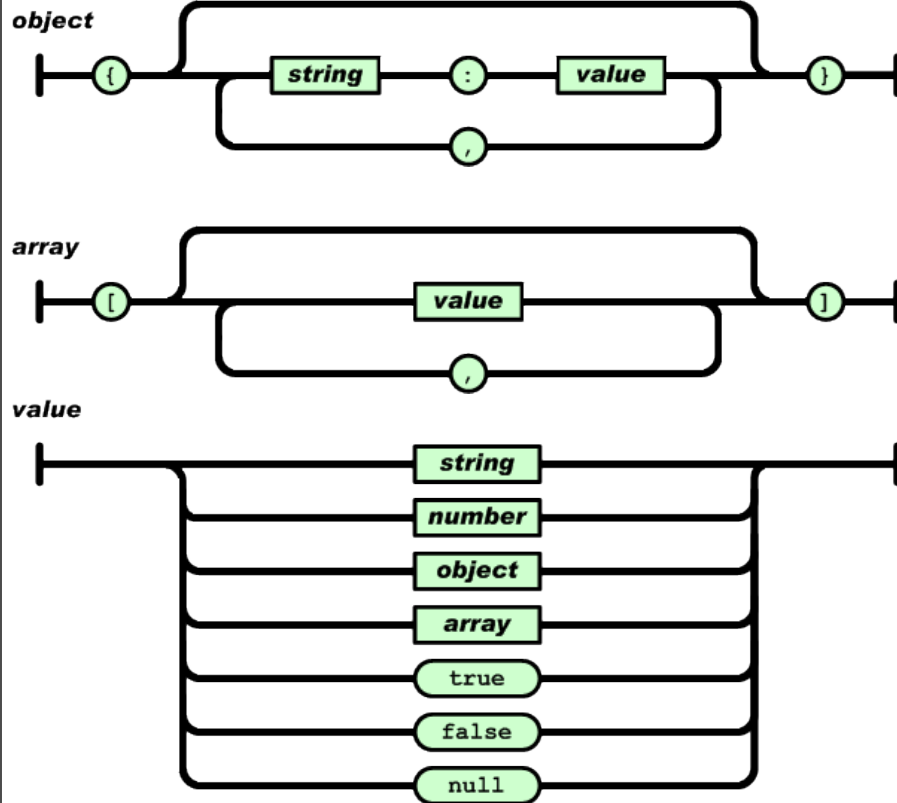
SesionMatematicas2.json

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesion SYSTEM "froga3.dtd">
<sesion>
  <idSesion>1</idSesion>
  <idUsuario>25</idUsuario>
  <operacion>
    <op1>12</op1>
    <op>-</op>
    <op2>2</op2>
    <res>10</res>
  </operacion>
  <operacion>
    <op1>34</op1>
    <op>+</op>
    <op2>5</op2>
    <res>39</res>
  </operacion>
</sesion>
```

SesionMatematicas2.java

3.- JSON

3.2.- Sintaxis



<http://www.json.org/>

3.- JSON

3.3.- AJAX

- AJAX
 - Asynchronous JavaScript And XML
 - Cambiar visualización de página
 -
 -
 - Objeto XMLHttpRequest
 - `open(method,url,async)`, `send()`, `send("")`, `.responseText`
- Procesar código JSON de la respuesta
 - `eval("código_JSON")`
 - **`JSON.parse("código_JSON")`**

3.- JSON

3.3.- AJAX

• Ejemplo: JSON+AJAX

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Coordenadas</title>
<script>
function obtenerCoordenadas()
{
    dir=document.getElementById("direccion").value;
    dirSinEspacios=sustituirEspacios(dir);

    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.open("GET","http://maps.googleapis.com/maps/api/geocode/json?address="+dirSinEspacios+"&sensor=false",false);
    xmlhttp.send();
    jsonRespuesta=xmlhttp.responseText;

    var jsO=eval("(" + jsonRespuesta + ")");
    document.getElementById("latitud").innerHTML="" + jsO.results[0].geometry.location.lat;
    document.getElementById("longitud").innerHTML="" + jsO.results[0].geometry.location.lng;
}
<!-- RESTO DE LINEAS EN LA PÁGINA SIGUIENTE -->
```

Coordenadas.html

3.- JSON

3.3.- AJAX

• Ejemplo: JSON+AJAX

```
<!-- RESTO DE LINEAS EN LA PÁGINA ANTERIOR -->
function sustituirEspacios(direccion)
{
    palabras=direccion.split(" ");
    l=palabras.length;
    sinEspacios=palabras[0];
    for(i=1;i<l;i++)
        sinEspacios+=" "+palabras[i];

    return sinEspacios;
}
</script>

<script>
</script>
</head>
<body>
<h1>APLICACIÓN PARA OBTENER COORDENADAS</h1>
<b>Latitud:</b><label id="latitud"></label><br>
<b>Longitud:</b><label id="longitud"></label><br>
<input id="direccion">
<button onclick="obtenerCoordenadas();">Obtener Coordenadas</button>

</body>
</html>
```

Coordenadas.html