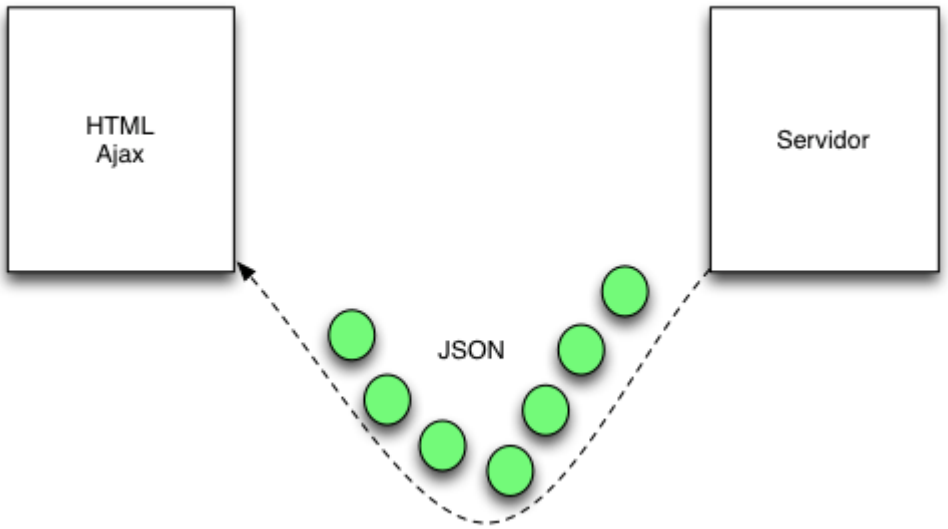
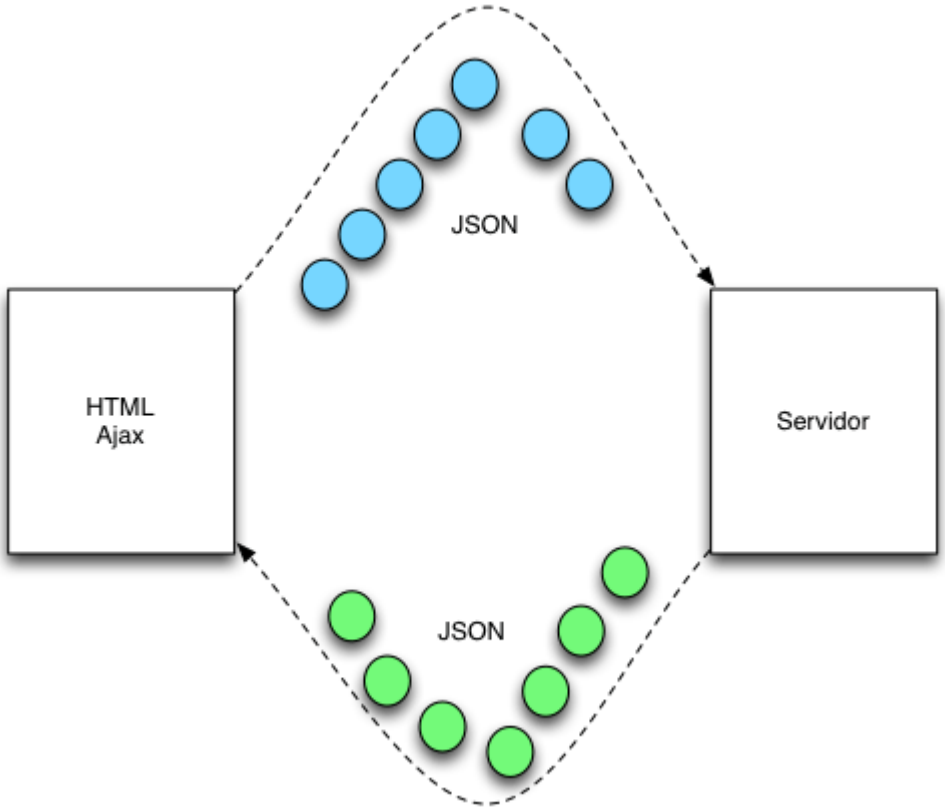
REST JSON son dos acrónimos que cada día se usan más. Estamos ya muy acostumbrados a recibir datos JSON en nuestras aplicaciones AJAX y HTML5 desde el servidor. De hecho se ha convertido en un standard de facto.



REST JSON y Envío

Aunque es cierto que esta forma de trabajar la tenemos ya muy integrada con nuestro negocio. También es bastante cierto que el envío de datos no siempre lo realizamos en JSON. Sin embargo esta práctica cada día se esta imponiendo más debido a que genera homogeneidad en la forma de trabajar con los datos.Hay frameworks como Angular.js que lo tiene completamente integrado.



REST JSON y Java

Vamos a ver como podemos abordar algo de este estilo desde el mundo Java. Para ello nos vamos a apoyar en JQuery y en una petición AJAX sencilla a la cual solicitamos que los datos se envíen en JSON.

```
var persona={"nombre":"pedro","apellidos":"perez"};

$.ajax({
url: "rest/servicioPersonas/personas",
type: "POST",
data: JSON.stringify(persona),
contentType: "application/json",
```

```
complete: resultado
});

&amp;amp;nbsp;
```

Una vez hecho esto tendremos que configurar un servicio REST para que pueda consumir datos en formato JSON. Para ello voy a utilizar
Apache CXF
<pre>y configurarlo en un Tomcat7 . Se puede ver como configurarlo en el siguiente artículo</pre>

<a title="Apache CXF y REST (Configuración)"
href="https://www.arquitecturajava.com/apache-cxf-y-rest/"> Apache CXF Configuración.
<pre> Una vez configurado vamos a dar de alta el servicio.

```
package com.arquitecturajava.serviciosexternos;

import javax.ws.rs.Consumes;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/servicioPersonas/")
@Produces("application/json")
```

```
public class ServicioPersonasJSON {

@POST
@Path("/personas")
@Consumes(MediaType.APPLICATION_JSON)
public void addPersona(Persona persona) {

System.out.println(persona.getNombre()+ persona.getApellidos());
}
}
```

En este caso el ejemplo es trivial ya que unicamente añadimos la anotación @Consumes y le especificamos que los datos son recibidos en formato JSON una vez lo hemos configurado el framework de JAX-RS se encargará de mapearlos a la clase Persona. En este caso simplemente imprimimos los datos por pantalla.

```
may 24, 2014 7:40:31 PM org.springframework.web.context.ContextLoade
INFO: Root WebApplicationContext: initialization completed in 964 ms
may 24, 2014 7:40:31 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
may 24, 2014 7:40:31 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
may 24, 2014 7:40:31 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 2817 ms
pedroperez
```

Vamos a echar un veo a los ficheros de configuración que Apache CXF necesita para este ejemplo en concreto funcione.

web.xml

```
&amp;lt;dependencies&amp;gt;
```

```
&amp;lt;?xml version="1.0" encoding="UTF-8"?&amp;gt;
&amp;lt;web-app version="2.5"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"&amp;gt;
&amp;lt;listener&amp;gt;
&amp;lt;listener-
class&amp;gt;org.springframework.web.context.ContextLoaderListener&amp
;lt;/listener-class&amp;gt;
&amp;lt;/listener&amp;gt;
&amp;lt;servlet&amp;gt;
&amp;lt;servlet-name&amp;gt;CXFServlet&amp;lt;/servlet-name&amp;gt;
&amp;lt;servlet-
class&amp;gt;org.apache.cxf.transport.servlet.CXFServlet&amp;lt;/servl
et-class&amp;gt;
&amp;lt;load-on-startup&amp;gt;1&amp;lt;/load-on-startup&amp;gt;
&amp;lt;/servlet&amp;gt;
&amp;lt;servlet-mapping&amp;gt;
&amp;lt;servlet-name&amp;gt;CXFServlet&amp;lt;/servlet-name&amp;gt;
&amp;lt;url-pattern&amp;gt;/rest/*&amp;lt;/url-pattern&amp;gt;
&amp;lt;/servlet-mapping&amp;gt;
&amp;lt;/web-app&amp;gt;

&amp;amp;nbsp;
```

applicationContext.xml

```
&amp;lt;?xml version="1.0" encoding="UTF-8"?&amp;gt;
&amp;lt;beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jaxrs="http://cxf.apache.org/jaxrs"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://cxf.apache.org/jaxrs
http://cxf.apache.org/schemas/jaxrs.xsd"&gt;

&amp;lt;import resource="classpath:META-INF/cxf/cxf.xml" /&amp;gt;
&amp;lt;import resource="classpath:META-INF/cxf/cxf-servlet.xml"
/&amp;gt;

&amp;lt;jaxrs:server id="servicioRest" address="/"&amp;gt;
&amp;lt;jaxrs:serviceBeans&amp;gt;

&amp;lt;bean
class="com.arquitecturajava.serviciosexternos.ServicioPersonasJSON"
/&amp;gt;

&amp;lt;/jaxrs:serviceBeans&amp;gt;
&amp;lt;jaxrs:providers&amp;gt;
&amp;lt;ref bean="jsonProvider"/&amp;gt;
&amp;lt;/jaxrs:providers&amp;gt;
&amp;lt;jaxrs:features&amp;gt;

&amp;lt;/jaxrs:features&amp;gt;
&amp;lt;/jaxrs:server&amp;gt;
&amp;lt;bean id="jsonProvider"
class="org.codehaus.jackson.jaxrs.JacksonJsonProvider"/&amp;gt;
&amp;lt;/beans&amp;gt;
```

CURSO Java Herencia

GRATIS

APUNTATE!!