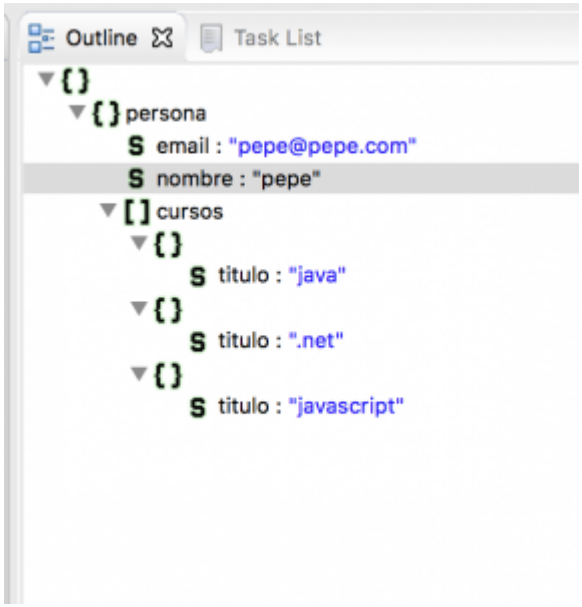
JSON-P (JSON Processing) es una de las novedades de Java Enterprise Edition 8 .¿Para que sirve JSON-P? . Esta nueva especificación sirve para procesar estructuras JSON de una forma sencilla y acceder a los datos de forma muy directa. La especificación se encarga de definir un puntero a la estructura de datos que necesitamos dentro de un documento JSON. Vamos a ver una pequeña introducción. En primer lugar vamos a definir las dependencias de Maven que necesitamos.

```
&amp;lt;project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"&amp;gt;
        &amp;lt;modelVersion&amp;gt;4.0.0&amp;lt;/modelVersion&amp;gt;
&amp;lt;groupId&amp;gt;com.arquitecturajava&amp;lt;/groupId&amp;gt;
        &amp;lt;artifactId&amp;gt;JSONP&amp;lt;/artifactId&amp;gt;
        &amp;lt;version&amp;gt;1&amp;lt;/version&amp;gt;
        &amp;lt;dependencies&amp;gt;
&amp;lt;dependency&amp;gt;
    &amp;lt;groupId&amp;gt;org.glassfish&amp;lt;/groupId&amp;gt;
    &amp;lt;artifactId&amp;gt;javax.json&amp;lt;/artifactId&amp;gt;
    &amp;lt;version&amp;gt;1.1.2&amp;lt;/version&amp;gt;
&amp;lt;/dependency&amp;gt;
        &amp;lt;/dependencies&amp;gt;
&amp;lt;/project&amp;gt;
```

Es momento de construir nuestro fichero JSON que va a tener unos datos muy sencillos.

```
{
	"persona": {
		"email": "pepe@pepe.com",
		"nombre": "pepe",
		"cursos": [
			{
				"titulo": "java"
			},
			{
				"titulo": ".net"
			},
			{
				"titulo": "javascript"
			}
		]
	}
}
```

En este caso estamos ante unos datos de una persona y los cursos que ha realizado. Eclipse nos puede mostrar una estructura a detalle del arbol:



Podemos usar JSON-P (JSON Processing) para acceder a la información de los cursos de una forma muy directa.

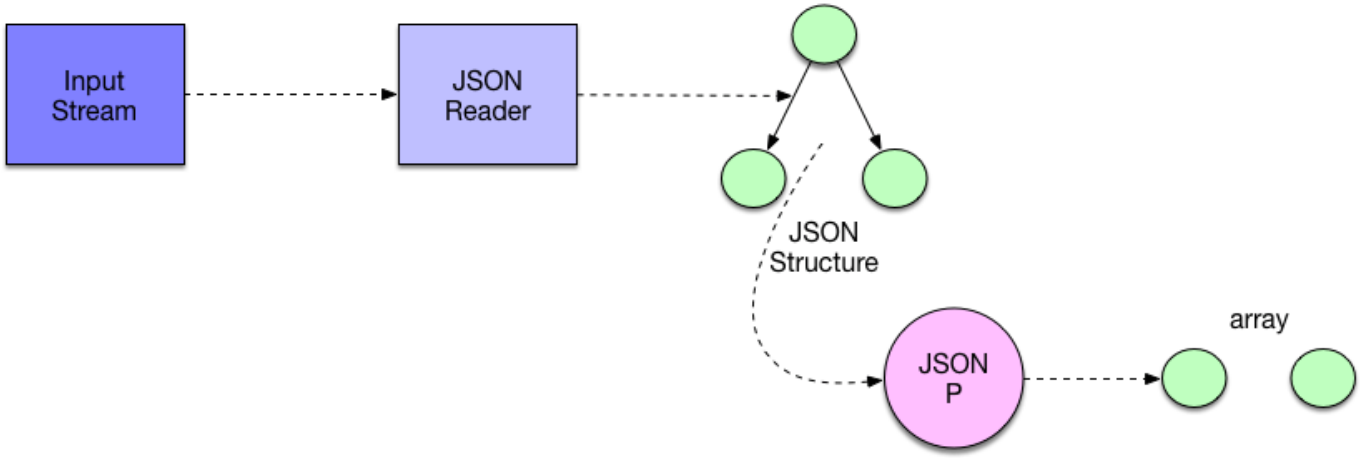
```
package com.arquitecturajava;

import java.io.IOException;
import java.io.InputStream;

import javax.json.Json;
import javax.json.JsonArray;
import javax.json.JsonPointer;
import javax.json.JsonReader;
import javax.json.JsonStructure;
import javax.json.JsonValue;

public class Principal {

        public static void main(String[] args) {
```

```
                try (InputStream is =
JsonPointer.class.getClassLoader().getResourceAsStream("datos.json");
                                    JsonReader jsonreader =
Json.createReader(is)) {

                        JsonStructure jsonEstructura =
jsonreader.read();

                        JsonPointer jsonPointer =
Json.createPointer("/persona/cursos");
                        JsonValue jv =
jsonPointer.getValue(jsonEstructura);
                        JsonArray lista=jv.asJsonArray();

                        lista.forEach(System.out::println);

                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        }
}
```

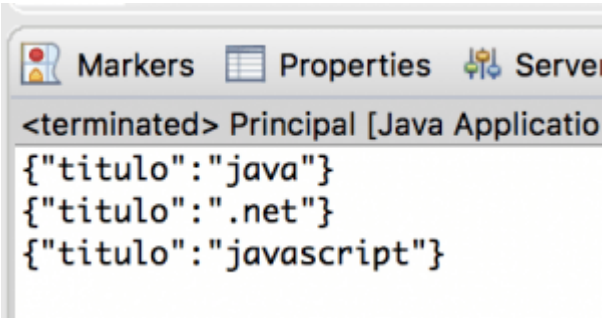
Utilizando JSON-P

En este caso el código es muy sencillo de entender. Partimos de un inputStream que se encarga de leer nuestro fichero JSON. Una vez tenemos el fichero JSON utilizamos un JSON Reader para leer el fichero. Realizado este paso leemos el fichero construimos una estructura en memoria a esa estructura la aplicamos un puntero (JSON pointer). Este

puntero indica que vamos a seleccionar unicamente los cursos de esta persona.



Realizado este paso convertimos el resultado del puntero y lo imprimimos en la consola como un sencillo array.



Acabamos de filtrar los datos de una estructura JSON utilizando JSON-P (JSON processing) una de las JSRs de Java EE 8 . El manejo de datos JSON cada día es más habitual y esta JSR nos puede aportar una solución rápida y elegante.

Otros artículos relacionados

- JAX-RS Client y JSON
- REST JSON y Java
- Objetos Javascript y JSON
- REST JSON y Java

Artículos externos

- JSON