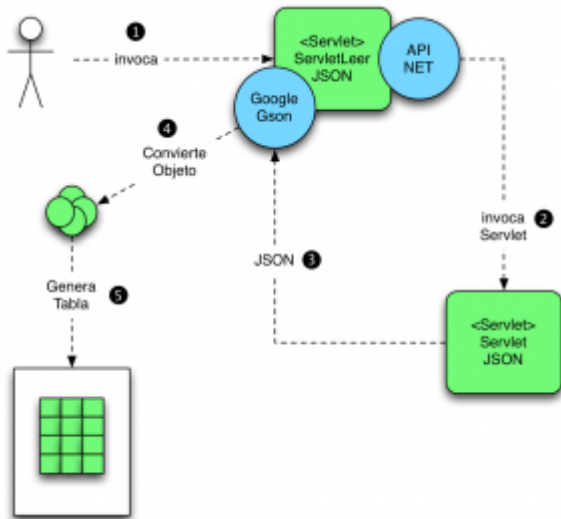


En el Post anterior de java y json hemos utilizado la librería de google para generar una estructura JSON a partir de una lista de personas .En este Post realizaremos la operación contraria y utilizaremos un Servlet para leer datos de nos llegan en formato JSON y convertirlos de forma sencilla en objetos para imprimirlos en HTML.



Java y JSON con un Servlet

Vamos a ver el código del Servlet que lee datos JSON .

```
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.URL;
import java.net.URLConnection;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```

import com.arquitecturajava.negocio.Persona;
import com.google.gson.Gson;
import com.google.gson.JsonArray;
import com.google.gson.JsonElement;
import com.google.gson.JsonParser;
import com.google.gson.stream.JsonReader;

public class ServletLeerJSON extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        URLConnection conexion = new
        URL("http://localhost:8080/JSONServlet/ServletJJson").openConnection();
        conexion.connect();

        JsonReader lector = new JsonReader(new InputStreamReader(
            conexion.getInputStream()));

        JsonParser parseador = new JsonParser();
        JsonElement raiz = parseador.parse(lector);

        Gson json = new Gson();
        JsonArray lista = raiz.getAsJsonArray();
        for (JsonElement elemento : lista) {
            Persona persona = json.fromJson(elemento, Persona.class);
            System.out.println(persona.getNombre());
        }
        PrintWriter pw=response.getWriter();
        pw.write("funciona");
        pw.close();
    }
}

```

```
}  
}
```

El código se encarga en primer lugar de abrir una conexión a la URL que almacena los datos en JSON:

```
URLConnection conexion = new  
URL("http://localhost:8080/JSONServlet/ServletJSon").openConnection();  
conexion.connect();
```

Una vez realizada esta operación usamos un `JsonReader` para leer la información a través de un `InputStreamReader`. Hecho esto usamos la clase `JsonParser` para parsear la estructura y convertirla en un `JsonElement`.

```
JsonReader lector = new JsonReader(new InputStreamReader(  
    conexion.getInputStream()));
```

```
JsonParser parseador = new JsonParser();  
JsonElement raiz = parseador.parse(lector);
```

Por último recorreremos la estructura y la convertimos en objetos java que serán impresos en un documento HTML clásico.

```
Gson gson = new Gson();  
JsonArray lista = raiz.getAsJsonArray();  
for (JsonElement elemento : lista) {  
    Persona persona = gson.fromJson(elemento, Persona.class);  
    System.out.println(persona.getNombre());  
}
```

```
}  
PrintWriter pw=response.getWriter();  
pw.write("funciona");  
pw.close();
```

Realizadas estas operaciones el servlet nos imprime los datos convertidos

