


Open Opened 2 weeks ago by  **Rubén Montero**

Listas en JSON

Resumen

- Veremos cómo se representa una lista en JSON
- Crearemos una clase sencilla con un método que sólo lee e imprime una lista de *strings* JSON

Descripción

En [JavaScript Object Notation](#) (JSON) se pueden representar *listas* de elementos.

Se hace usando **corchetes** (`[]`) y separando los valores con **comas** (`,`).

Por ejemplo, esto es un documento JSON válido:

```
[5, 4, -6, 4]
```

Las listas pueden ser de cualquier tipo:

```
["Benzema", "Messi"]
```

Incluso objetos:

```
[
  { "jugador": "Benzema", "equipo": "Real Madrid" },
  { "jugador": "Messi", "equipo": "FC Barcelona" }
]
```

Anidamiento

No es que una lista *tenga que ser* el elemento raíz de un JSON.

Como ya sabes, los objetos JSON pueden anidarse, y las listas también.

Por ejemplo, esta sintaxis es válida:

```
{
  "jugador": "Benzema",
  "equipos": [
    "Olympique Lyonnais",
    "Real Madrid"
  ]
}
```

¿Y cómo se lee una lista JSON de un fichero?

De la misma forma que ya sabemos, pero empleando un `JSONArray` en lugar de un `JSONObject`:

```
FileReader reader = new FileReader("assets\\wrestlers.json");
JSONTokener tokenizer = new JSONTokener(reader);
JSONArray jsonArray = new JSONArray(tokenizer);
```

¿Y cómo se usa un `JSONArray`?

Acceder a sus elementos

Es parecido a un `ArrayList` de los que hemos usado con anterioridad.

Tiene métodos:

- `.getString`

- `.getInt`
- ...

...para acceder a su contenido. Reciben un **índice**:

```
["Undertaker", "John Cena", "David Batista"]
```

```
// Imprime "Undertaker"
System.out.println(jsonArray.getString(0));

// Imprime "David Batista"
System.out.println(jsonArray.getString(2))
```

Así que es común acceder a ellos con un **bucle** `for` :

```
for (int i = 0; i < jsonArray.length(); i++) {
    String wrestlerName = jsonArray.getString(i);
    System.out.println(wrestlerName);
}
```

Añadir elementos

La operación inversa, es decir, **introducir** elementos en un `JSONArray`, es trivial. Basta con usar `.put` y pasar como parámetro la variable que deseemos añadir:

```
JSONArray array = new JSONArray();
array.put(/* Aquí iría un elemento */);
array.put(/* Podemos añadir tantos como queramos */);
```

...algo muy similar al `.add` de `ArrayList` que ya conocemos.

La tarea

Se pide que:

- **Creas** una nueva clase `WrestlersReader`
- **No** definas un constructor
- Únicamente, **añade un** método `readWrestlers` que:
 - Sea `public void`
 - **No** reciba parámetros
 - Lea el fichero `wrestlers.json` e imprima, línea a línea, los nombres de los luchadores que se hallan en formato JSON

Por último

Verifica que el `test` funciona correctamente.

Haz `commit` y `push` para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 1](#) 2 weeks ago



Ania Blanco @ania.blanco mentioned in commit [92674d4f](#) 1 week ago