


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

Guardando preferencias en JSON

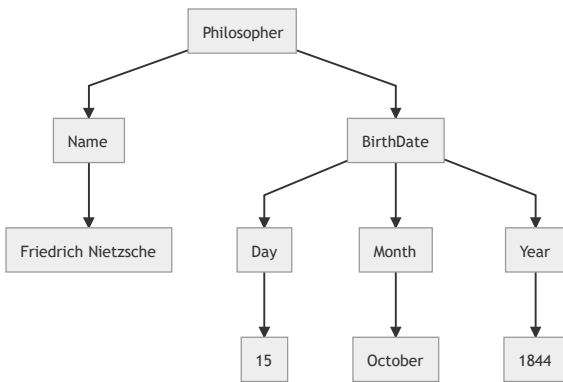
Resumen

- Echaremos un vistazo al formato JSON
- Añadiremos un método `saveExampleJSON` a nuestra clase `HomeCinemaPreferences.java` que creará y guardará un JSON de ejemplo (`assets\example.json`), con dos atributos
- Añadiremos `saveAsJSON`, que guardará un JSON (`cinemaPrefs.json`) con los valores de las preferencias

Descripción

[JavaScript Object Notation](#) (JSON) es un formato ligero de intercambio de datos. A pesar de su nombre, es independiente del lenguaje JavaScript.

Es un formato sencillo que se basa en almacenar *tuplas* **clave-valor** y encerrar *objetos* entre **llaves** (`{ }`). Por ejemplo, esta información:



...se almacenaría así:

```
{
  "philosopher": {
    "name": "Friedrich Nietzsche",
    "birthDate": {
      "day": 15,
      "month": "October",
      "year": 1844
    }
  }
}
```

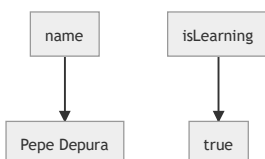
Fíjate que los miembros que se encuentran al **mismo nivel** se **separan** con **comas** (`,`), y que los valores, a excepción de los numéricos, van entre **comillas** (`" "`).

Guardando las preferencias

Comencemos

Escribamos un nuevo método `saveExampleJSON` en nuestro `HomeCinemaPreferences.java`.

El **objetivo final** de este **ejemplo** será crear *desde 0* un objeto JSON con dos atributos:



...y escribirlo a disco:

```
{
  "name": "Pepe Depura",
  "isLearning": true
}
```

Crear un `JSONObject`

La clase que *representa* un objeto JSON se llama `JSONObject`, en la librería `org.json:json-20220320` usada en el proyecto.

Crearlo es **fácil**:

```
JSONObject jsonObject = new JSONObject();
```

...y **almacenar** un par de valores, también:

```
jsonObject.put("name", "Pepe Depura");
jsonObject.put("isLearning", true);
```

Escribir a un fichero

Podemos guardarlo a un fichero... **fácilmente**:

```
FileWriter writer = new FileWriter("assets\\example.json");
jsonObject.write(writer, 2, 0); // Estos números indican la indentación del resultado,
                                // espacios en blanco que mejoran la legibilidad

writer.flush();
writer.close();
```

(Hay que considerar el `IOException`)

¡Método **terminado**!

La tarea

Se pide:

- Que `HomeCinemaPreferences.java` cuente con el código `saveExampleJSON` visto arriba
- Que añadas un nuevo método `saveAsJSON`:
 - Guardará el valor de los atributos `username` y `darkModePreferred` a disco duro en un archivo `assets\\cinemaPrefs.json`
 - Tendrá un contenido JSON como el siguiente:

```
{
  "username": "John Doe",
  "prefersDarkMode": true
}
```

Un comentario extra

Nótese que podríamos *anidar* un `JSONObject` dentro de otro si quisiéramos conseguir resultados como el primer ejemplo de la tarea. Por ejemplificarlo, sería algo así:

```
// Esto es un ejemplo aparte
JSONObject jsonBirthDate = new JSONObject();
jsonBirthDate.put("day", 15);
jsonBirthDate.put("month", "October");
jsonBirthDate.put("year", 1844);

JSONObject jsonPhilosopher = new JSONObject();
jsonPhilosopher.put("birthDate", jsonBirthDate);
```

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 [c90fe68d](#) 1 week ago