

## EJEMPLO COMO OBTENER VALORES A PARTIR DE LA RESPUESTA DE UNA API EN FORMATO JSON

```
// SE IMPORTA LA LIBRERÍA json.jar

//Declaramos nuestra variable de tipo String con un formato de Json
String responseJson = "{
    \"Account\":{
        \"Holder\":\"Gari Kaspárov\",
        \"CurrencyCode\":\"CRC\",
        \"ProductName\":\"Cuenta\",
        \"State\":\"1\"
    },
    \"IsSuccessful\":true,
    \"Errors\":[{
        \"Account\":123456789,
        \"Code\":\"asde23\"
    }],
    \"OperationId\":\"31e32f8f-6da5-4d3c-8434-203d8f53ed89\"
}";

//Convertimos el string en un objeto json
JSONObject json = new JSONObject(responseJson);

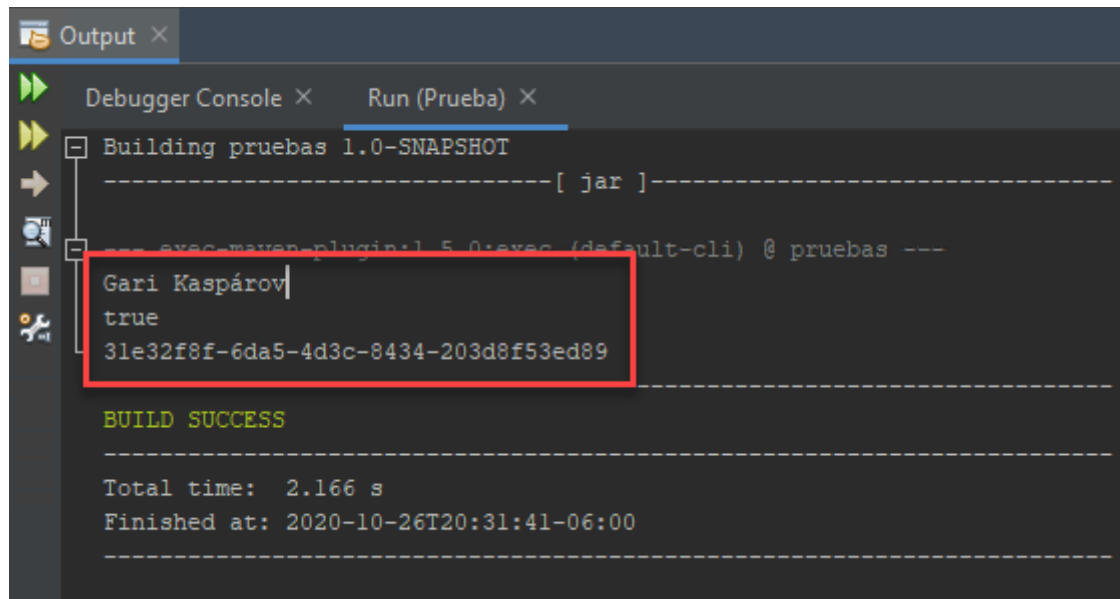
//Obtenemos la propiedad dentro de una rama del json
String holder = json.getJSONObject("Account").getString("Holder");

//Obtenemos la propiedad de tipo boolean fuera de una rama en un json
boolean isSuccessful = json.getBoolean("IsSuccessful");

//Obtenemos la propiedad de tipo string fuera de una rama en un json
String operationId = json.getString("OperationId");

//Para el caso de Errors, los [] representan que hay un arreglo en el json, para extraer los datos
debemos acceder al arreglo
Integer account = json.getJSONArray("Errors").getJSONObject(0).getInt("Account");
String code = json.getJSONArray("Errors").getJSONObject(0).getString("Code");

//Imprimimos las respuestas
System.out.println(holder);
System.out.println(isSuccessful);
System.out.println(operationId);
```



```
Output ×
Debugger Console × Run (Prueba) ×
Building pruebas 1.0-SNAPSHOT
-----[ jar ]-----
... exec-maven-plugin:1.5.0:exec (default-cli) @ pruebas ---
{
  "name": "Gari Kaspárov",
  "isHuman": true,
  "id": "31e32f8f-6da5-4d3c-8434-203d8f53ed89"
}
BUILD SUCCESS
-----
Total time: 2.166 s
Finished at: 2020-10-26T20:31:41-06:00
-----
```

Como vemos en el ejemplo, lo importante es conocer el lugar donde se encuentre el campo que necesitamos extraer del json, ya sea dentro de una rama del json o fuera de ella.

Además, es importante definir en el llamado de la función, el tipo de dato que traeremos del objeto json, en el ejemplo usamos **json.getString** y **json.getBoolean** pero podemos utilizar otros métodos como **getDouble**, **getFloat**, **getLong**, **getInt**, entre otros, esto nos ayudará a tipar automáticamente el tipo de dato que recibamos en el json y así procesarlo mejor.

En caso de que no exista el campo a consultar, pueden gestionarse excepciones en JSON. Más información en los enlaces

ENLACES:

<https://oracle-max.com/como-obtener-los-valores-de-las-propiedades-de-un-json-en-java/>

<https://stleary.github.io/JSON-java/org/json/JSONException.html>