Il Formato JSON

Struttura, sintassi, esempi ed utilizzi

Francesco Gobbi

20 febbraio 2026

Cos'è JSON

Definizione

JSON (**JavaScript Object Notation**) è un formato standard aperto per:

- memorizzare informazioni in modo strutturato
- scambiare dati tra applicazioni (web e non)

- È semplice da scrivere e da analizzare
- È indipendente dalla piattaforma e dal linguaggio
- È ampiamente supportato nelle applicazioni web moderne

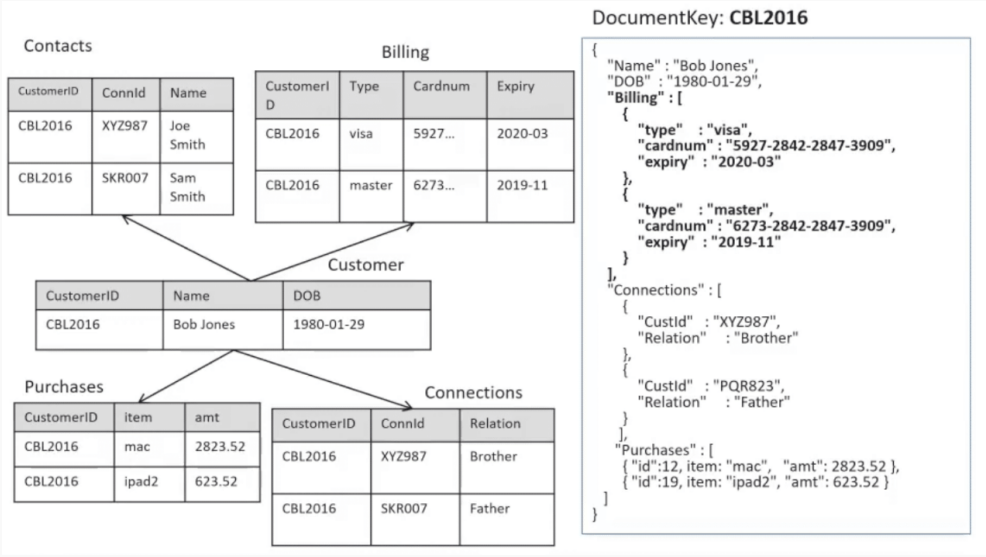
Perché JSON è così diffuso

- È compatto rispetto a XML (meno struttura ripetuta)
- Si adatta bene allo scambio dati (risposte di servizi web / API)
- Qualsiasi linguaggio che gestisce stringhe può analizzarlo

Database e JSON

- DB relazionali (es. **MySQL**, **PostgreSQL**) supportano campi JSON
- DB NoSQL (es. **MongoDB**) usano documenti in stile JSON per i record

JSON nei Database Relazionali



Differenze tra XML e JSON

XML

- Molto flessibile: tag personalizzabili
- Verboso: tag aperti/chiusi
- Utile in contesti complessi e documentali

JSON

- Più compatto e leggibile
- Struttura immediata (oggetti/array)
- Ideale per scambio dati in applicazioni web

Idea chiave

JSON tende a essere più semplice e immediato da leggere; XML più descrittivo ma spesso più complesso.

Differenze tra XML e JSON

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<endereco>
    <cep>31270901</cep>
    <city>Belo Horizonte</city>
    <neighborhood>Pampulha</neighborhood>
    <service>correios</service>
    <state>MG</state>
    <street>Av. Presidente Antônio Carlos, 6627</street>
</endereco>
```

vs.

JSON

```json
{
    "endereco": {
        "cep": "31270901",
        "city": "Belo Horizonte",
        "neighborhood": "Pampulha",
        "service": "correios",
        "state": "MG",
        "street": "Av. Presidente Antônio Carlos, 6627"
    }
}
```

Formato di JSON

JSON è costituito da **due sole strutture**:

- **insieme di coppie** (nome, valore) = oggetti
- **lista ordinata** di valori = array

Esempio (oggetto)

```
{
  "nome": "Andrea",
  "eta": 18
}
```

Oggetti letterali JSON

Un **oggetto** è un insieme di coppie `chiave: valore`:

- racchiuso tra { e }
- coppie separate da virgole
- le chiavi sono stringhe

Sintassi

```
{
  "proprieta1": "valore",
  "proprieta2": "valore"
}
```

Nota importante

JSON ammette solo valori: stringhe, numeri, booleani, array, oggetti, `null`. Non contiene funzioni.

Tipi di dato supportati

JSON supporta i seguenti tipi:

- **Number** (interi e decimali)
- **String**
- **Boolean** (`true`/`false`)
- **Array**
- **Object**
- **null**

Nota sui numeri

In JSON non sono previsti formati come ottale o esadecimale, e non esistono `NaN` o `Infinity`.

Esempi rapidi di tipi

String e Number

```
{
  "nome": "Andrea",
  "eta": 18
}
```

Boolean

```
{
  "promosso": true
}
```

Array

```
{
  "voti": [6, 7, 8]
}
```

null

```
{
  "valore": null
}
```

Array e Object

Array: lista ordinata di valori

```
{
  "books": [
    {"linguaggio": "Java",   "edizione": "seconda"},
    {"linguaggio": "Python", "edizione": "prima"}
  ]
}
```

Object: insieme di coppie chiave-valore

```
{ "id": "01669",
  "linguaggio": "JAVA",
  "prezzo": 15.50 }
```

Whitespace (spazi)

Gli spazi (whitespace) possono essere inseriti per migliorare la leggibilità:

- non cambiano il significato del JSON
- rendono il documento più facile da leggere

Creare oggetti JSON in JavaScript

In JavaScript possiamo creare oggetti (in stile JSON) in tre modi:

1. Creare un oggetto vuoto
2. Creare un oggetto con il costruttore `Object()`
3. Creare un oggetto assegnando valori direttamente

Esempi

```
var oggetto1 = {};
var oggetto2 = new Object();
var oggetto3 = { "titolo":"TPSIT vol.3", "prezzo":23.50 };
```