

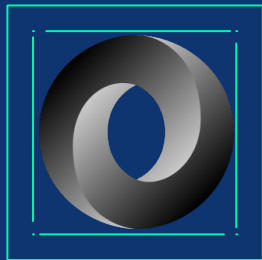


# JSON & JSON SCHEMA

David Herrero Faura  
David Pérez Barroso  
Iván Risueño Martín  
Andreu Orensanz Bargalló

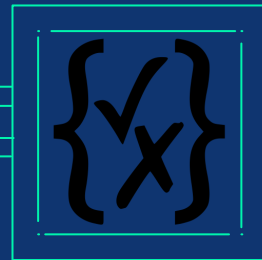


# INTRODUCCIÓ



## JSON

Format de text senzill  
per l'intercanvi de  
dades

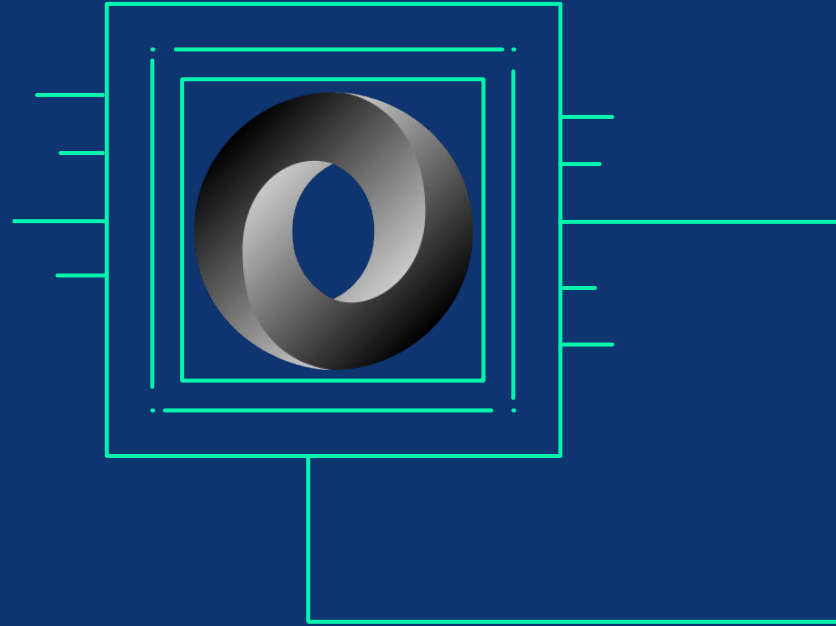


## JSON SCHEMA

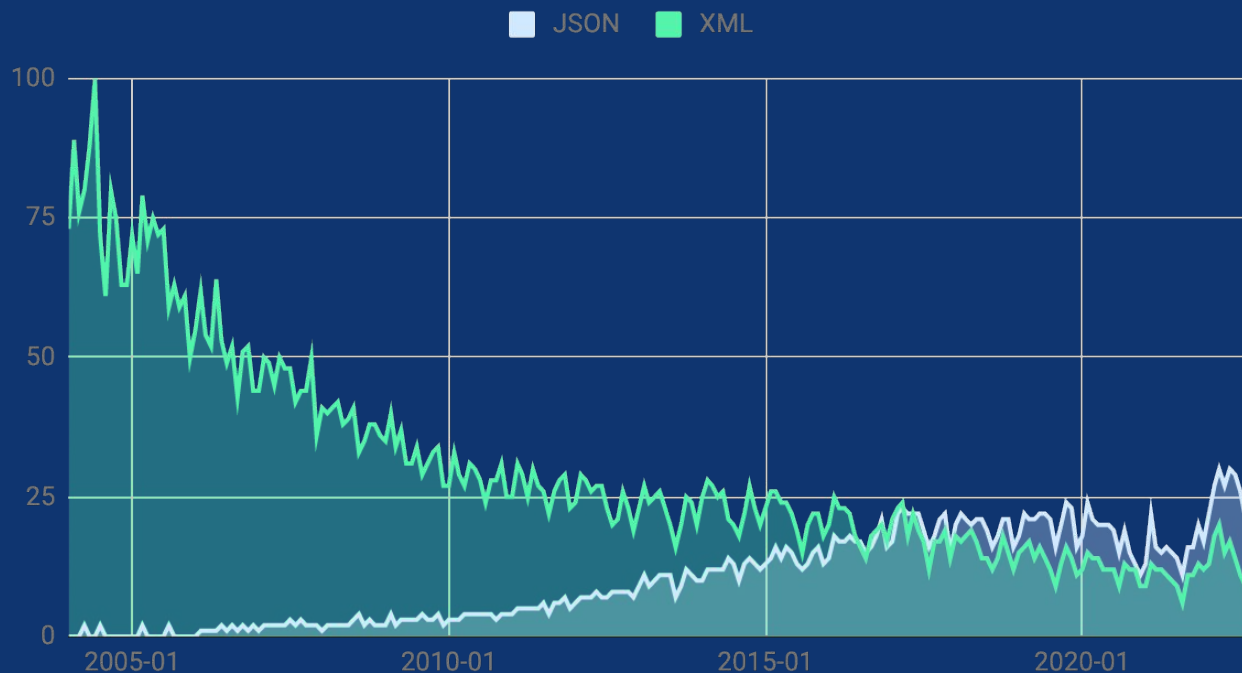
Vocabulari que permet realitzar  
anotacions i validacions de  
documents JSON

# HISTÒRIA DE JSON

- Definit per Douglas Crockford l'any 2001
- Èxit en poc temps degut a la senzillesa i facilitat d'ús



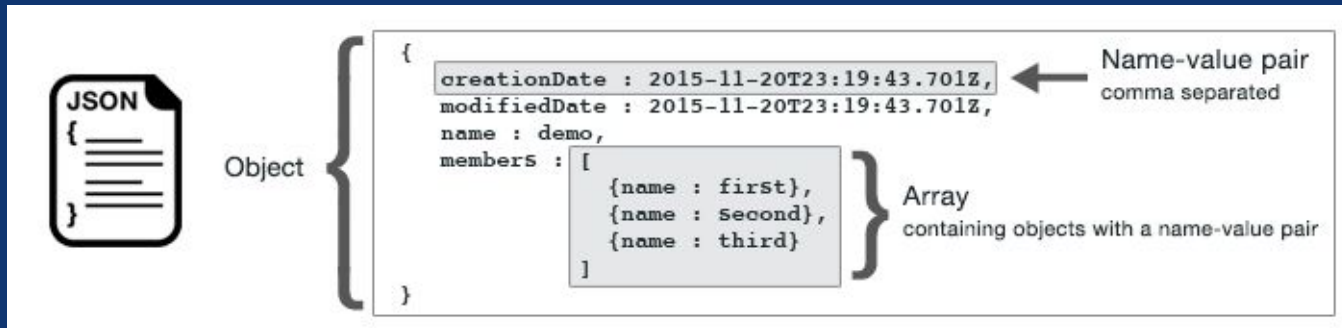
## Comparació JSON - XML al llarg dels anys



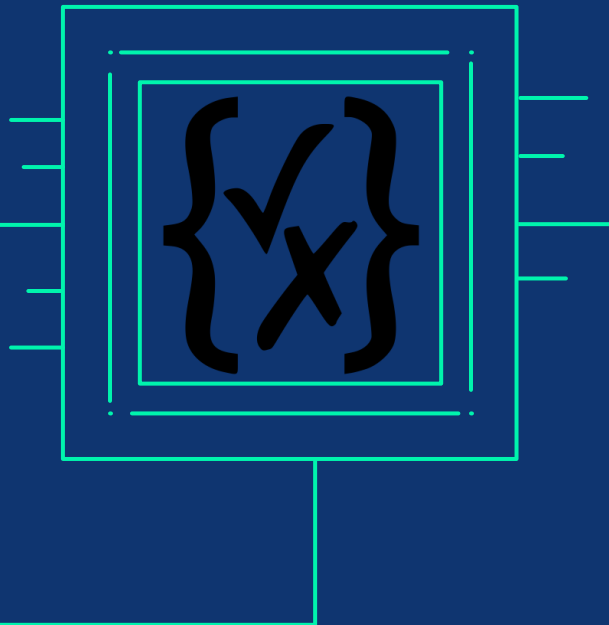
Font: <https://trends.google.es/trends/explore?date=all&q=%2Fm%2F05cmtt%2Fm%2F08745>

# CARACTERÍSTIQUES JSON

- Parells "clau - valor"
- Llistes ordenades de valors
- Els valors poden ser *Strings*, números, booleans, objectes JSON
- Flexible, lleuger i fàcilment transferible

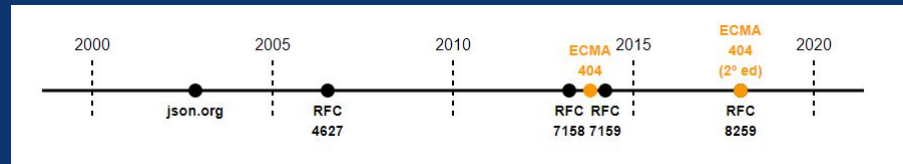


# HISTÒRIA DE JSON SCHEMA



Les darreres versions de les normes són les següents:

- **març del 2015:** RFC 7493: The I-JSON Message Format
- **desembre de 2017:** RFC 8259: (JSON) Data Interchange Format
- **desembre de 2017:** ECMA 404 (2n ed): The JSON Data Interchange Syntax



# CARACTERÍSTIQUES JSON SCHEMA

- És una llista de regles, recollides en un objecte JSON
- Funciona amb parells *keyword* - *value*
- Document JSON se l'anomena instància i document amb la descripció se l'anomena *schema*
- Els *keywords* comuns: *type*, *minLength*, *minItems*



# EXEMPLES

JSON Schema i instàncies de JSON





# EXEMPLE JSON

Exemple d'instància JSON

```
JSON Object → {  
  "company": "mycompany",  
  "companycontacts": { ← Object Inside Object  
    "phone": "123-123-1234",  
    "email": "myemail@domain.com"  
  },  
  "employees": [ ← JSON Array  
    {  
      "id": 101,  
      "name": "John",  
      "contacts": [ ← Array Inside Array  
        "email1@employee1.com",  
        "email2@employee1.com"  
      ]  
    },  
    {  
      "id": 102, ← Number Value  
      "name": "William",  
      "contacts": null ← Null Value  
    }  
  ]  
}
```



# EXEMPLE APLICATIU JSON

Moltes crides a APIs retornen les dades en format JSON

```
{
  "statuses": [
    {
      "created_at": "Mon Oct 03 07:52:32 +0000 2022",
      "id": 1576842625638092800,
      "id_str": "1576842625638092800",
      "text": "@NickFarmer3 You know why he's limited weapons to Ukraine, @EmmanuelMacron in",
      "truncated": true,
      "entities": {
        "hashtags": [],
        "symbols": [],
        "user_mentions": [
          {
            "screen_name": "NickFarmer3",
            "name": "Nick Farmer",
            "id": 890629417,
            "id_str": "890629417",
            "indices": [
              0,
              12
            ]
          },
          {
            "screen_name": "EmmanuelMacron",
            "name": "Emmanuel Macron",
            "id": 1976143068,
            "id_str": "1976143068",
            "indices": [
              59,
              74
            ]
          }
        ]
      }
    }
  ]
}
```



# EXEMPLE JSON SCHEMA 1

Conceptes bàsics sobre l'estructura

```
{
  "$id": "https://example.com/person.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Person",
  "type": "object",
  "properties": {
    "firstName": {
      "type": "string",
      "description": "The person's first name."
    },
    "lastName": {
      "type": "string",
      "description": "The person's last name."
    },
    "age": {
      "description": "Age in years which must be equal to or greater than zero.",
      "type": "integer",
      "minimum": 0
    }
  }
}
```

```
{
  "firstName": "John",
  "lastName": "Doe",
  "age": 21
}
```



# EXEMPLE JSON SCHEMA 2

Restriccions: *required*, *maximum* i *minimum*

```
{
  "$id": "https://example.com/geographical-location.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Longitude and Latitude Values",
  "description": "A geographical coordinate.",
  "required": [ "latitude", "longitude" ],
  "type": "object",
  "properties": {
    "latitude": {
      "type": "number",
      "minimum": -90,
      "maximum": 90
    },
    "longitude": {
      "type": "number",
      "minimum": -180,
      "maximum": 180
    }
  }
}
```

```
{
  "latitude": 48.858093,
  "longitude": 2.294694
}
```



# EXEMPLE JSON SCHEMA 3

## - ESTRUCTURA

Referències i definicions: *\$ref* i *\$defs*

```
{
  "$id": "https://example.com/arrays.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "description": "A representation of a person, company, organization, or place",
  "type": "object",
  "properties": {
    "fruits": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "vegetables": {
      "type": "array",
      "items": { "$ref": "#/$defs/veggie" }
    }
  },
  "$defs": {
    "veggie": {
      "type": "object",
      "required": [ "veggieName", "veggieLike" ],
      "properties": {
        "veggieName": {
          "type": "string",
          "description": "The name of the vegetable."
        },
        "veggieLike": {
          "type": "boolean",
          "description": "Do I like this vegetable?"
        }
      }
    }
  }
}
```



# EXEMPLE JSON SCHEMA 3

## - DADES

Referències i definicions: *\$ref* i *\$defs*

```
{
  "fruits": [ "apple", "orange", "pear" ],
  "vegetables": [
    {
      "veggieName": "potato",
      "veggieLike": true
    },
    {
      "veggieName": "broccoli",
      "veggieLike": false
    }
  ]
}
```



# AVANTATGES DE L'ÚS DE JSON

- Processament ràpid i eficient (fàcil de parsejar).
- Ús del parell clau:valor (fàcil de llegir).
- Facilitat per al compartiment de dades.
- Suport de la majoria de plataformes i llenguatges.

# DESAVANTATGES DE L'ÚS DE JSON

- No permet strings de més d'una línia.
- Només accepta certs tipus de dades. No pot representar ni imatges ni gràfics.
- Poc extensible. S'han de definir noves etiquetes o atributs.
- S'han d'utilitzar elements externs per tractar la seguretat.



# CONCLUSIONS

- Ideal per a tractar objectes amb molts atributs però curts.
- Fàcil de llegir per a un humà i de tractar i parsejar per a un ordinador.
- Ideal d'utilitzar si es té una sistema de seguretat configurat (OAuth, Bearer).
- Avantatges i desavantatges de ser text/plain (simple, soportable, representar estructures complexes).

# REFERÈNCIES

<https://json-schema.org/learn/miscellaneous-examples.html>

<https://www.mclibre.org/consultar/informatica/lecciones/formato-json.html>

<https://www.techquintal.com/advantages-and-disadvantages-of-json/>

<https://xdoc.mx/preview/json-schema-departamento-de-computacion-5c1feba62b02b>

<https://ca.wikipedia.org/wiki/JSON>



# REPARTIMENT DE TASQUES

## DAVID HERRERO

- Desavantatges JSON
- Conclusions
- Orador

## IVÁN RISUEÑO

- Exemples JSON Schema
- Avantatges JSON

## ANDREU ORENSANZ

- Història JSON
- Característiques JSON schema
- Exemple JSON aplicatiu
- Orador

## DAVID PÉREZ

- Introducció
- Història JSON Schema
- Característiques JSON
- Exemple JSON



# GRÀCIES PER LA VOSTRA ATENCIÓ

Si teniu cap pregunta, estarem encantats de resoldre-la

