Introduction :

Un fichier JSON (JavaScript Object Notation) est un format de fichier qui utilise du texte lisible par l'homme pour transmettre des données sous forme de paires clé-valeur ou d'arrays. Il est principalement utilisé pour transmettre des données entre un serveur et une application web, comme un moyen de formatage des données.

Voici un exemple simple de ce à quoi ressemble un fichier JSON :

```
{
    "nom": "Dupont",
    "prenom": "Jean",
    "age": 30,
    "adresse": {
        "rue": "123 rue du Paradis",
        "ville": "Paris",
        "codePostal": "75000"
    },
    "hobbies": ["lecture", "cinéma", "voyage"]
}
```

Dans cet exemple, nous avons un objet qui contient des informations sur une personne. Les données sont organisées en paires clé-valeur : "nom" est une clé et "Dupont" est la valeur associée à cette clé. L'objet "adresse" est un autre objet JSON imbriqué, et "hobbies" est un tableau JSON.

Le format JSON est facile à lire et à écrire pour les humains, et facile à analyser et à générer pour les machines, ce qui en fait un choix populaire pour le stockage et l'échange de données dans de nombreuses applications.

Prenons un deuxième exemple :

Soit le fichier JSON suivant :

```
{
    "chaine 1": {
        "chaine": "TF1",
        "multicast": "239.1.1.1",
        "port": 1234
    },
    "chaine 2": {
        "chaine": "BFMTV",
        "multicast": "239.1.1.2",
        "port": 1234
    },
    "chaine 3": {
        "chaine": "France2",
        "multicast": "239.1.1.3",
        "port": 1234
    },
    "chaine 4": {
```

```
        "chaine": "France 4",
        "multicast": "239.1.1.4",
        "port": 1234
    },
    "chaine 5": {
        "chaine": "France 5",
        "multicast": "239.1.1.5",
        "port": 1234
    },
    "chaine 6": {
        "chaine": "M6",
        "multicast": "239.1.1.6",
        "port": 1234
    },
    "chaine 7": {
        "chaine": "ARTE",
        "multicast": "239.1.1.7",
        "port": 1234
    },
    "chaine 8": {
        "chaine": "",
        "multicast": "",
        "port": 0
    },
    "chaine 9": {
        "chaine": "C8",
        "multicast": "239.1.1.19",
        "port": 1234
    },
    "chaine 10": {
        "chaine": "",
        "multicast": "",
        "port": 0
    },
    "chaine 11": {
        "chaine": "",
        "multicast": "",
        "port": 0
    },
    "chaine 12": {
        "chaine": "",
        "multicast": "",
        "port": 0
    }
}
```


On nomme ce fichier settings.json. Ce fichier contient les paramètres de 12 chaînes IPTV (nom de la chaîne TV, IP multicast et port.

L'objet que vous avez fourni est un objet JSON. Il est constitué de plusieurs paires clé-valeur, où chaque clé est une chaîne de caractères (par exemple, "chaine 1", "chaine 2", etc.) et chaque valeur est un autre objet JSON.

Chaque sous-objet JSON (la valeur associée à chaque clé comme "chaine 1", "chaine 2", etc.) contient trois paires clé-valeur :
"chaine" : une chaîne de caractères représentant le nom d'une chaîne de télévision.
"multicast" : une chaîne de caractères représentant une adresse multicast.
"port" : un nombre représentant un port.

Par exemple, pour la clé "chaine 1", la valeur est un objet qui représente la chaîne de télévision "TF1", avec une adresse multicast de "239.1.1.1" et un port de "1234".
Cet objet JSON pourrait être utilisé pour stocker et transmettre des informations sur différentes chaînes de télévision, y compris leur nom, leur adresse multicast et leur port.

Créons un premier script NodeJs dans un fichier pour lire notre document JSON:

Lecture d'un fichier JSON :

```
const fs = require('fs');

fs.readFile('settings.json', 'utf8', (err, jsonString) => {
    if (err) {
        console.log('Error reading file from disk:', err);
        return;
    }
    console.log(jsonString);
    try {
        const data = JSON.parse(jsonString);
        console.log(data);
    } catch(err) {
        console.log('Error parsing JSON string:', err);
    }
});
```

Le résultat qui s'affiche et qui nous présente la structure de notre fichier JSON est le suivant :

```
{
  "chaine 1": {
"chaine": "TF1", "multicast": "239.1.1.1", "port": 1234
  },
  "chaine 2": {
"chaine": "BFMTV", "multicast": "239.1.1.2", "port": 1234
  },
  "chaine 3": {
"chaine": "France2", "multicast": "239.1.1.3", "port": 1234
  },
  "chaine 4": {
"chaine": "France 4", "multicast": "239.1.1.4", "port": 1234
  },
  "chaine 5": {
"chaine": "France 5", "multicast": "239.1.1.5", "port": 1234
  },
  "chaine 6": {
"chaine": "M6", "multicast": "239.1.1.6", "port": 1234
  },
  "chaine 7": {
"chaine": "ARTE", "multicast": "239.1.1.7", "port": 1234
```

```
  },
    "chaine 8": {
"chaine": "", "multicast": "", "port": 0
},
"chaine 9": {
  "chaine": "C8","multicast": "239.1.1.19","port": 1234
},
"chaine 10": {
  "chaine": "",
"multicast": "",
"port": 0 },
"chaine 11": {
  "chaine": "",
"multicast": "",
"port": 0 },
"chaine 12": {
  "chaine": "",
"multicast": "",
"port": 0 }
}
```

```
{
  'chaine 1': { chaine: 'TF1', multicast: '239.1.1.1', port: 1234 },
  'chaine 2': { chaine: 'BFMTV', multicast: '239.1.1.2', port: 1234 },
  'chaine 3': { chaine: 'France2', multicast: '239.1.1.3', port: 1234 },
  'chaine 4': { chaine: 'France 4', multicast: '239.1.1.4', port: 1234 },
  'chaine 5': { chaine: 'France 5', multicast: '239.1.1.5', port: 1234 },
  'chaine 6': { chaine: 'M6', multicast: '239.1.1.6', port: 1234 },
  'chaine 7': { chaine: 'ARTE', multicast: '239.1.1.7', port: 1234 },
  'chaine 8': { chaine: '', multicast: '', port: 0 },
  'chaine 9': { chaine: 'C8', multicast: '239.1.1.19', port: 1234 },
  'chaine 10': { chaine: '', multicast: '', port: 0 },
  'chaine 11': { chaine: '', multicast: '', port: 0 },
  'chaine 12': { chaine: '', multicast: '', port: 0 }
                                                    }
```

Extraction d'une valeur d'un fichier JSON :

Ce code lit le fichier settings.json, le parse en objet JavaScript, puis extrait la valeur de la propriété 'chaine' de l'objet 'chaine 1' et l'affiche dans la console.

```
const fs = require('fs');

fs.readFile('settings.json', 'utf8', (err, jsonString) => {
    if (err) {
        console.log('Error reading file from disk:', err);
        return;
    }
    try {
        const obj = JSON.parse(jsonString);
        const chaine = obj['chaine 1']['chaine'];
        console.log(chaine);
    } catch(err) {
        console.log('Error parsing JSON string:', err);
    }
});
```

Ce code lit le fichier settings.json, le parse en objet JavaScript, puis extrait les valeurs des propriétés 'chaine', 'multicast' et 'port' de l'objet 'chaine 7'. Ensuite, il affiche ces valeurs dans une phrase.

```
const fs = require('fs');

fs.readFile('settings.json', 'utf8', (err, jsonString) => {
    if (err) {
        console.log('Error reading file from disk:', err);
        return;
    }
    try {
        const obj = JSON.parse(jsonString);
        const chaine = obj['chaine 7']['chaine'];
        const multicast = obj['chaine 7']['multicast'];
        const port = obj['chaine 7']['port'];

        console.log(`La chaîne 7 est la chaine ${chaine}, elle a
l'adresse multicast ${multicast}, son port est ${port}`);
    } catch(err) {
        console.log('Error parsing JSON string:', err);
    }
});
```

Dans l'exemple précédent, le résultat est :
La chaîne 7 est la chaine ARTE, elle a l'adresse multicast 239.1.1.7, son port est 1234.

Création d'un objet JSON :

Ce code crée un objet JavaScript, le convertit en chaîne JSON avec JSON.stringify(), puis reconvertit la chaîne JSON en objet JavaScript avec JSON.parse(), et affiche l'objet résultant dans la console.

```
const tableau = {
    'chaine': 'TF1',
    'multicast': '239.1.1.1',
    'port': 1234
};

const objet_json = JSON.stringify(tableau);
console.log(JSON.parse(objet_json));
```

Le résultat obtenu est :

```
{ chaine: 'TF1', multicast: '239.1.1.1', port: 1234 }
```