



Presentación

Nombre: Andy Manuel

Apellidos: Reyes Del Rosario

Matricula: 100488534

Carrera: Licenciatura en Informática

Asignatura: INF5170

Sección: 01

Maestro: Radhames Silverio González

JSON

JavaScript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web (por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o vice versa). Se enfrentará a menudo con él, así que este artículo le entrega todo lo que necesita saber para trabajar con JSON utilizando JavaScript, incluyendo el análisis JSON para acceder a los datos en su interior, y cómo crear JSON.

¿Qué es realmente JSON?

JSON es un formato de datos basado en texto que sigue la sintaxis de objeto de JavaScript, popularizado por Douglas Crockford. Aunque es muy parecido a la sintaxis de objeto literal de JavaScript, puede ser utilizado independientemente de JavaScript, y muchos entornos de programación poseen la capacidad de leer (convertir; parsear) y generar JSON.

Los JSON son cadenas - útiles cuando se quiere transmitir datos a través de una red. Debe ser convertido a un objeto nativo de JavaScript cuando se requiera acceder a sus datos. Esto no es un problema, dado que JavaScript posee un objeto global JSON que tiene los métodos disponibles para convertir entre ellos.

Un objeto JSON puede ser almacenado en su propio archivo, que es básicamente sólo un archivo de texto con una extensión `.json`, y una MIME type (en-US) de `application/json`.

Estructura del JSON

Como se describió previamente, un JSON es una cadena cuyo formato recuerda al de los objetos literales JavaScript. Es posible incluir los mismos tipos de datos básicos dentro de un JSON que en un objeto estándar de JavaScript - cadenas, números, arreglos, booleanos, y otros literales de objeto. Esto permite construir una jerarquía de datos, como ésta:

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "name": "Madame Uppercut",
    "age": 39,
    "secretIdentity": "Jane Wilson",
    "powers": [
      "Million tonne punch",
      "Damage resistance",
      "Superhuman reflexes"
    ]
  },
  {
    "name": "Eternal Flame",
    "age": 1000000,
    "secretIdentity": "Unknown",
    "powers": [
      "Immortality",
      "Heat Immunity",
      "Inferno",
      "Teleportation",
      "Interdimensional travel"
    ]
  }
]
}

```

Si se carga este objeto en un programa de JavaScript, convertido (parseado) en una variable llamada `superHeroes` por ejemplo, se podría acceder a los datos que contiene utilizando la misma notación de punto/corchete que se revisó en el artículo JavaScript object basics. Por ejemplo:

```

superHeroes.homeTown
superHeroes['active']

```

Para acceder a los datos que se encuentran más abajo en la jerarquía, simplemente se debe concatenar los nombres de las propiedades y los índices de arreglo requeridos. Por ejemplo, para acceder al tercer superpoder del segundo héroe registrado en la lista de miembros, se debería hacer esto:

```

superHeroes['members'][1]['powers'][2]

```

1. Primero el nombre de la variable — `superHeroes`.
2. Dentro de esta variable para acceder a la propiedad `members` utilizamos `["members"]`.
3. `members` contiene un arreglo poblado por objetos. Para acceder al segundo objeto dentro de este arreglo se utiliza `[1]`.
4. Dentro de este objeto, para acceder a la propiedad `powers` utilizamos `["powers"]`.
5. Dentro de la propiedad `powers` existe un arreglo que contiene los superpoderes del héroe seleccionado. Para acceder al tercer superpoder se utiliza `[2]`.

Arreglos como JSON

Anteriormente se mencionó que el texto JSON básicamente se parece a un objeto JavaScript, y esto es en gran parte cierto. La razón de esto es que un arreglo es también un JSON válido, por ejemplo:

```
[
  {
    "name": "Molecule Man",
    "age": 29,
    "secretIdentity": "Dan Jukes",
    "powers": [
      "Radiation resistance",
      "Turning tiny",
      "Radiation blast"
    ]
  },
  {
    "name": "Madame Uppercut",
    "age": 39,
    "secretIdentity": "Jane Wilson",
    "powers": [
      "Million tonne punch",
      "Damage resistance",
      "Superhuman reflexes"
    ]
  }
]
```

Éste es un JSON perfectamente válido. Para acceder a esta version convertida se debe comenzar con un índice de arreglo, por ejemplo `[0]["powers"][0]`.