



JSON.stringify() & JSON.parse()

JSON.stringify()

JSON.stringify() เป็น built-in JavaScript method, ที่แปลงจาก JSON(JavaScript Object Notation) String เป็น JavaScript Object

- JSON เป็น lightweight data format ที่ถูกใช้สำหรับข้อมูลที่แลกเปลี่ยนกันระหว่าง Web Servers และ Browser
- เป็น Subset ย่อยของภาษา JavaScript

เพื่อแปลงจาก JavaScript Object ไปเป็น JSON String, User สามารถใช้ JSON.stringify() method และส่ง Object ในรูปแบบของ Argument

- หาก Object สามารถถูกแปลงไปเป็น JSON String ได้ JSON.stringify() จะคืนเป็น String ที่แสดงถึงข้อมูลใน Object

```
const person = {
  name: "John",
  age: 30,
  address: {
    city: "New York",
    state: "NY"
  }
};

const jsonString = JSON.stringify(person);

console.log(jsonString); // Output: '{"name":"John","age":30,"address":{"city":"New York","state":"NY"}}'
```

ในตัวอย่าง:

- เรามี JavaScript Object ด้วย Properties ที่ถูกนำเสนอในรูปแบบชุดของ Person ที่มี 3 Properties: name, age, address

- เราส่ง Object นี้ไปให้ JSON.stringify() ซึ่งจะส่งคืนเป็นค่า JSON String ที่มีข้อมูลที่เกี่ยวข้องอยู่ข้างใน

Important Note:

JSON.stringify() จะไม่รวม Properties ใดๆ ที่มีค่าเป็น undefined, function, symbol value

MDN Web Docs:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify

JSON.parse()

JSON.parse() เป็น built-in JavaScript method, ที่แปลงจาก JSON(JavaScript Object Notation) String เป็น JavaScript Object

- JSON เป็น lightweight data format ที่ถูกใช้สำหรับข้อมูลที่แลกเปลี่ยนคุ้ยกันระหว่าง Web Servers และ Browser
- เป็น Subset ย่อยของภาษา JavaScript

เพื่อแปลงจาก JSON String ไปเป็น JavaScript Object, User สามารถใช้ JSON.parse() method และส่ง JSON String ในรูปแบบของ Argument

- หาก JSON String นั้นถูกต้อง, JSON.parse() จะส่งคืนเป็น JavaScript Object

```
const jsonString = '{"name": "John", "age": 30}';
const person = JSON.parse(jsonString);

console.log(person.name); // Output: "John"
console.log(person.age); // Output: 30
```

ในตัวอย่างเรามี JSON String ที่เป็น Object มี 2 Properties ด้วยกันคือ

- name และ age

เราจะส่ง JSON String นี้ไปยัง JSON.parse() ซึ่งจะส่งคืน JavaScript Object ที่เรากำหนดตัวแปรข้างใน จากนั้นเราสามารถเข้าถึง Properties ของข้อมูลใน Object ได้ผ่านเครื่องหมายจุด(.)

Important Note:

JSON.parse() จะ Throw Error ถ้า JSON String นั้นไม่ถูกต้อง

- เพื่อหลีกเลี่ยง Error, สามารถใช้ “try...catch” Statement เพื่อจับ Error ใดๆก็ตามที่อาจเกิดขึ้น

```
const invalidJsonString = '{name: "John", "age": 30}';

try {
  const person = JSON.parse(invalidJsonString);
  console.log(person);
} catch (error) {
```

```
console.error(error);  
}
```

ดังในตัวอย่าง :

- Missing quote ที่ name Property
- เมื่อเราพยายาม JSON String นี้โดยใช้ JSON.parse(), มันจะ Throw 'SyntaxError' เพราะ String ใน JSON นั้นไม่ถูกต้อง
- เราจึงใช้ try...catch Statement เพื่อ Log Error ออกมาใน Console

MDN Web Docs:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse