SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

Course Code and Name : 19TS601 FULL STACK DEVELOPMENT

Unit 3 : NODEJS AND EXPRESS

Topic : JSON

JSON

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a lightweight format for storing and transporting data
- JSON is often used when data is sent from a server to a web page
- JSON is "self-describing" and easy to understand

- **JSON Example**

An employees object: an array of 3 employee records (objects):

```
{
"employees":[
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName":"Jones"}
]
}
```

JSON Syntax Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JavaScript Object Notation

- The JSON format is syntactically identical to the code for creating JavaScript objects.
- Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.
- The JSON syntax is derived from JavaScript object notation syntax, but the JSON format is text only.
- Code for reading and generating JSON data can be written in any programming language.

JSON Data - A Name and a Value

- JSON data is written as name/value pairs, just like JavaScript object properties.
- A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:
- "firstName":"John"
- JSON names require double quotes.
- JavaScript names do not.

JSON Objects

- JSON objects are written inside curly braces.
- Just like in JavaScript, objects can contain multiple name/value pairs:
- {"firstName":"John", "lastName":"Doe"}

JSON Arrays

- JSON arrays are written inside square brackets.
- Just like in JavaScript, an array can contain objects:

```
"employees":[
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName":"Jones"}
]
```

- In the example above, the object "employees" is an array. It contains three objects.
- Each object is a record of a person (with a first name and a last name).

Converting a JSON Text to a JavaScript Object

- A common use of JSON is to read data from a web server, and display the data in a web page.
- For simplicity, this can be demonstrated using a string as input.
- First, create a JavaScript string containing JSON syntax:

```
var text = '{ "employees" : [' +
'{ "firstName":"John" , "lastName":"Doe" },' +
'{ "firstName":"Anna" , "lastName":"Smith" },' +
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

- Then, use the JavaScript built-in function JSON.parse() to convert the string into a JavaScript object:
- var obj = JSON.parse(text);
- Finally, use the new JavaScript object in your page:

Example

```
<!DOCTYPE html>
<html>
<body>
 <h2>Create Object from JSON String</h2>
 <p id="demo"></p>
 <script>
var text = '{"employees":[' +
'{"firstName":"John","lastName":"Doe" },' +
'{"firstName":"Anna","lastName":"Smith" },' +
'{"firstName":"Peter","lastName":"Jones" }]}';
 obj = JSON.parse(text);
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>
 </body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
<body>
 <h2>Create Object from JSON String</h2>
 <p id="demo"></p>
 <script>
var text = '{"employees":[' +
'{"firstName":"John","lastName":"Doe" },' +
'{"firstName":"Anna","lastName":"Smith" },' +
'{"firstName":"Peter","lastName":"Jones" }]}';
 obj = JSON.parse(text);
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>
 </body>
</html>
```

Create Object from JSON String

- Express.js is a popular web application framework for Node.js, offering robust features for building web applications, including single-page, multi-page, and API-driven applications.


Thank
You!