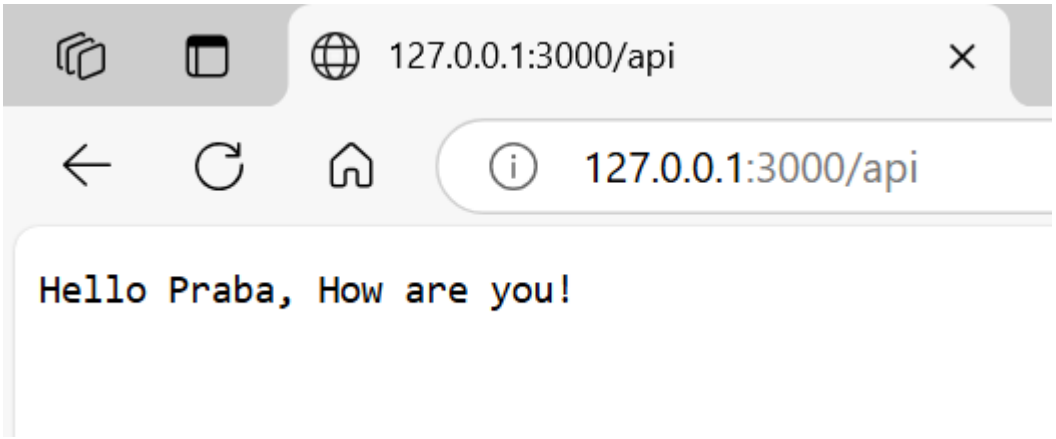
NodeJS HTTP Server

What are Modules in NodeJS?

- NodeJS HTTP module
  - Allows you to create web servers and handle HTTP requests and responses, making it a fundamental part of building web applications in NodeJS.
  - It plays a crucial role in handling server-side HTTP requests and responses, allowing for seamless communication between clients and servers.
- Uses the require() method to include the HTTP module.
- It provides utilities to create both client and server applications.
- Supports various HTTP methods like GET, POST, PUT, DELETE, etc.
- Allows ease in handling request headers, query parameters, and response bodies.

Creation of HTTP Server – Example 1

```
const http = require('http');
http.createServer((request, response) => {
    response.write('Hello Praba, How are you!');
    response.end();
}).listen(3000,'127.0.0.1');
console.log("Server started on port 3000");
```



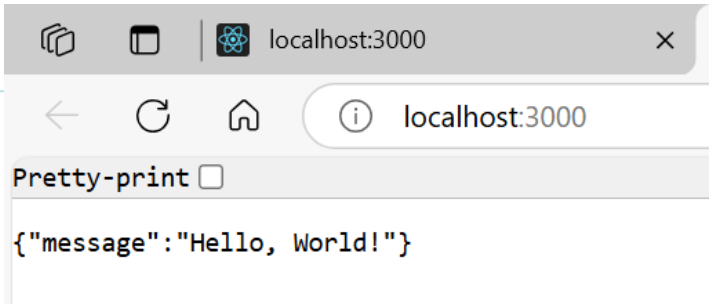
Creation of HTTP Server – Example 1

Example 1 - Description

- http.createServer() is used to create a new HTTP server.
- The (request) object represents the incoming request from the client.
- The (response) object is used to send the HTTP response to the client.
- response.write() sends data as part of the response.
- response.end() signals that the response is complete.
- server.listen() makes the server listen on the specified port (3000 in this case).

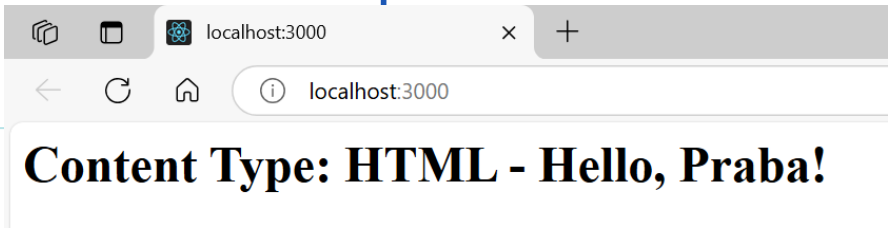
Request & Response – Example 2

```
var http = require('http');
var url = require('url');
http.createServer(function (req, res) {
    var contentType = 'application/json';
    res.writeHead(200, { 'Content-Type': contentType });
    if (contentType === 'text/html') {
        res.write('<html><body><h1>Content Type: HTML - Hello, World!</h1></body></html>');
    } else if (contentType === 'application/json') {
        res.write(JSON.stringify({message: "Hello, World!"}));
    }
    res.end();
}).listen(3000, () => {
    console.log('Server running at http://localhost:3000/');
});
```



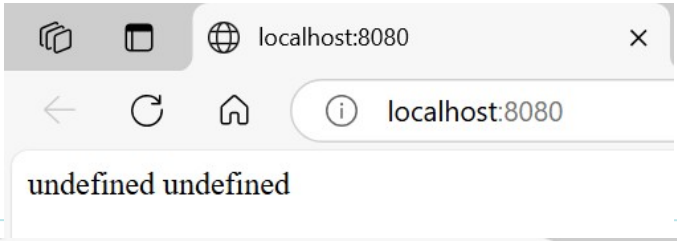
JSON Stringify plays a significant role in the JSON concept. It's a function in JavaScript that converts a JavaScript object into a JSON string. This method is commonly used to send data from the client side to a web server in a string format.

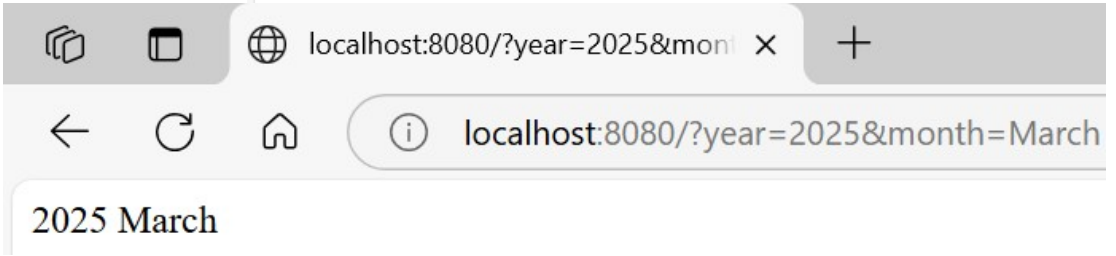
Request & Response – Example 2




```
var http = require('http');
var url = require('url');
http.createServer(function (req, res) {
    var contentType = ‘text/html';
    res.writeHead(200, { 'Content-Type': contentType });
    if (contentType === 'text/html') {
        res.write('<html><body><h1>Content Type: HTML - Hello, Praba!</h1></body></html>');
    } else if (contentType === 'application/json') {
        res.write(JSON.stringify({message: "Hello, World!"}));
    }
    res.end();
}).listen(3000, () => {
    console.log('Server running at http://localhost:3000/');
});
```

Query String – Example 3






```
var http = require('http');
var url = require('url');
http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    var q = url.parse(req.url, true).query; // Parse the query
string
    var txt = q.year + " " + q.month; // Get year and month from
                                          query string
    res.end(txt); // Display the result
}).listen(8080, () => {
    console.log('Server running at http://localhost:8080/');
});
```