

Node.js

Sarah Coufal

What is Node.js?

- Open source runtime environment for executing javascript code outside of a browser
- Initially released 2009
- Express: Node.js web application framework



app.listen()

- Starts the server on specified port and listens for connections

```
app.listen(port, function (error) {  
  if (error) throw error  
  console.log("Server created Successfully")  
});
```

app.use()

- mounts the middleware function (the function that has access to the request and response objects) at the specified path

```
app.use("/hello", function (req, res) {  
  console.log("hello");  
  res.end('hello world');  
});
```

- Hello World demo
 - In terminal, `node hello.js` to start server
 - In browser, <http://165.106.10.170:31009/hello> (or whatever host/port)

Short and Long Polling Recap

- **Polling:** process by which client asks the server for new data
- **Short polling:** client sends request every couple seconds, server response is either empty or contains data object
- **Long polling:** client sends request, server keeps connection open until it receives new data, and when it does sends response with updated data
- Short/long polling demo

Short polling

```
app.use("/short", function (req, res) {  
  console.log("short request");  
  let aa = { 'hits': counter }  
  res.end(JSON.stringify(aa));  
});
```

- Connection opened
- Sends counter value response back to client
- Connection closed
- Repeats every 2 seconds

Long polling

- Connection opened
- Saves current counter value
- Checks if counter has been updated
 - If it has, sends updated counter value to client and closes connection
 - If it hasn't, checks again after a second

```
function lct(res, currentC) {
  if (currentC !== counter) {
    //console.log("counter changed!");
    let aa = { 'hits': counter }
    res.end(JSON.stringify(aa));
  } else {
    //console.log("Counter unchanged");
    setTimeout(function () { lct(res, currentC) }, 1000);
  }
}

app.use("/long", function (req, res) {
  console.log("long request");
  let currentC = counter;
  setTimeout(function () { lct(res, currentC) }, 1000);
});
```

References

<https://nodejs.org/en/about/>

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

<https://www.geeksforgeeks.org/express-js-app-listen-function/>

<https://www.geeksforgeeks.org/difference-between-app-use-and-app-get-in-express-js/>