

Reference Sheet: Node.js and Network Communication

Name: _____ Section: _____

Debugging

`console.log(message)`
`message` : string

Writes a line of text to the browser console or terminal in which Node.js is running

Node.js: Server-side JavaScript

`require(module)`

`module` : string Returns object

Loads and returns the library module with the specified name

Module `http`

`createServer(handler)`

`handler` : function Returns object

Creates a *server* whose incoming requests are processed by the supplied function:

`handler(request, response)`

`request` : object `response` : object

Called each time a new connection is received; it will be held open until explicitly closed

`server.listen(port)`

`port` : number

Starts accepting connections on the specified port. Use port 80 to handle `http://localhost`, or *e.g.* port 1024 for `http://localhost:1024` (must use this or a higher port on Mac OS).

`request.url`

string

The path to the requested resource (*e.g.* file). To strip the leading slash: `request.url.substring(1)`

`response.end(message)`

`message` : string

Transmits the specified reply and closes the connection

Module `url`

`parse(url, arguments)`

`url` : string `arguments` : boolean Returns object

Returns an object that breaks the URL into pieces given by string properties (*e.g.* `pathname` for the path to the resource). If `arguments` was `true`, it also contains an object `query` with a string property for each named argument (*e.g.* `name1` and `name2` in `/filename.html?name1=value1&name2=value2`).

Module `fs`

`existsSync(path)`

`path` : string Returns boolean

Reports whether there exists a file at the specified path

`readFileSync(path)`

`path` : string Returns string

Returns a string containing the contents of the file at the specified path. Crashes if it does not exist!

XMLHttpRequest: Client-side networking in JavaScript

`new XMLHttpRequest()`

Returns object

Constructs a bidirectional network *connection*

`connection.open(method, url)`

`method` : string `url` : string

To behave as if the `url` was entered in the browser's location bar, set `method` to 'GET'.

`connection.send()`

Transmits the request to the server. Call after `open()`.

`connection.onreadystatechange`

Set to a handler function to be called each time progress is made. Once `connection.readyState` is `XMLHttpRequest.DONE`, the string `connection.responseText` contains the server's reply.