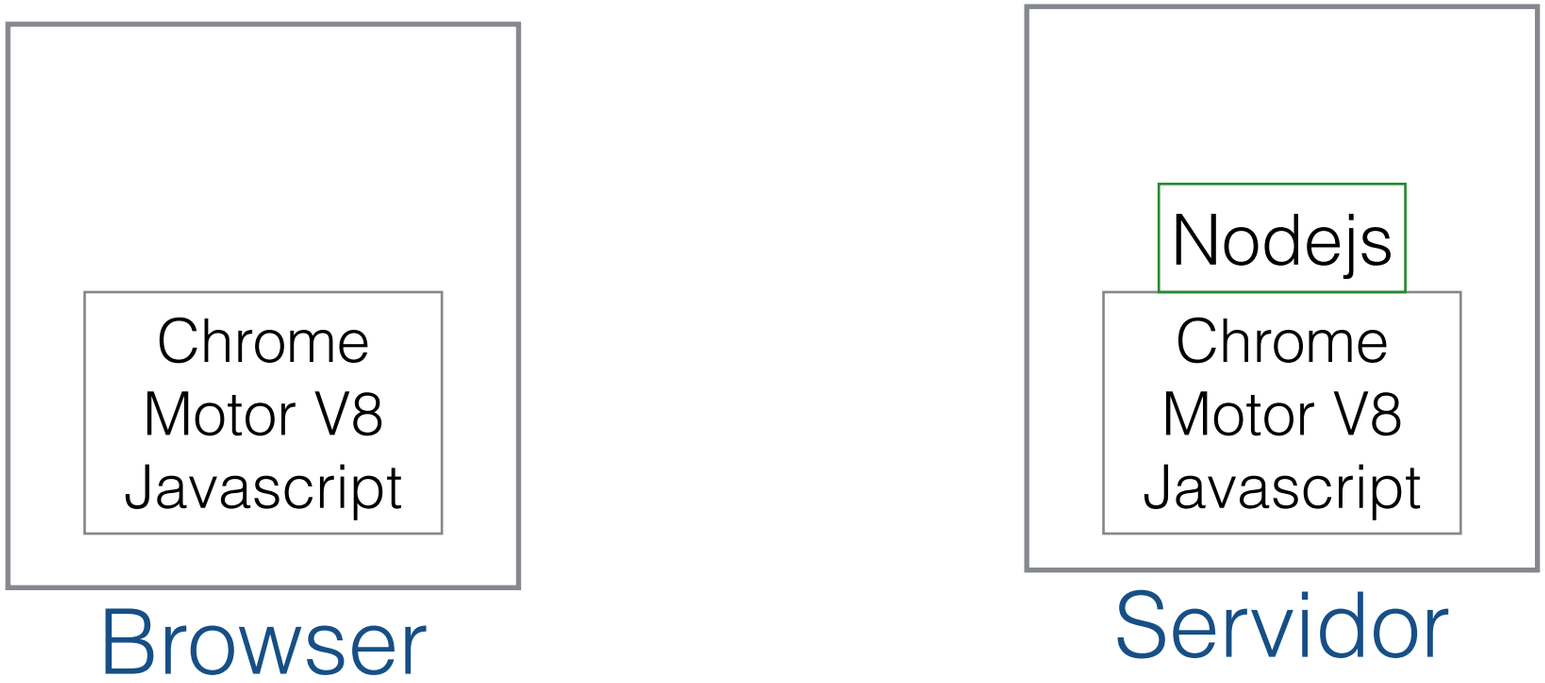
Node.js

- Plataforma construida sobre el Javascript runtime de Chrome (https://code.google.com/p/v8/) para fácilmente construir aplicaciones rápidas y escalables.

- Usa un modelo dirigido por eventos, Non-blocking I/O que lo hace liviano y eficiente, perfecto para aplicaciones de datos intensivos en real-time a través de dispositivos distribuidos.


Chrome
Motor V8
Javascript
Browser
Nodejs
Chrome
Motor V8
Javascript
Servidor

Cómo funciona?

- Trabaja con un único hilo de ejecución. Este hilo organiza todo el trabajo que le mandamos a ejecutar.

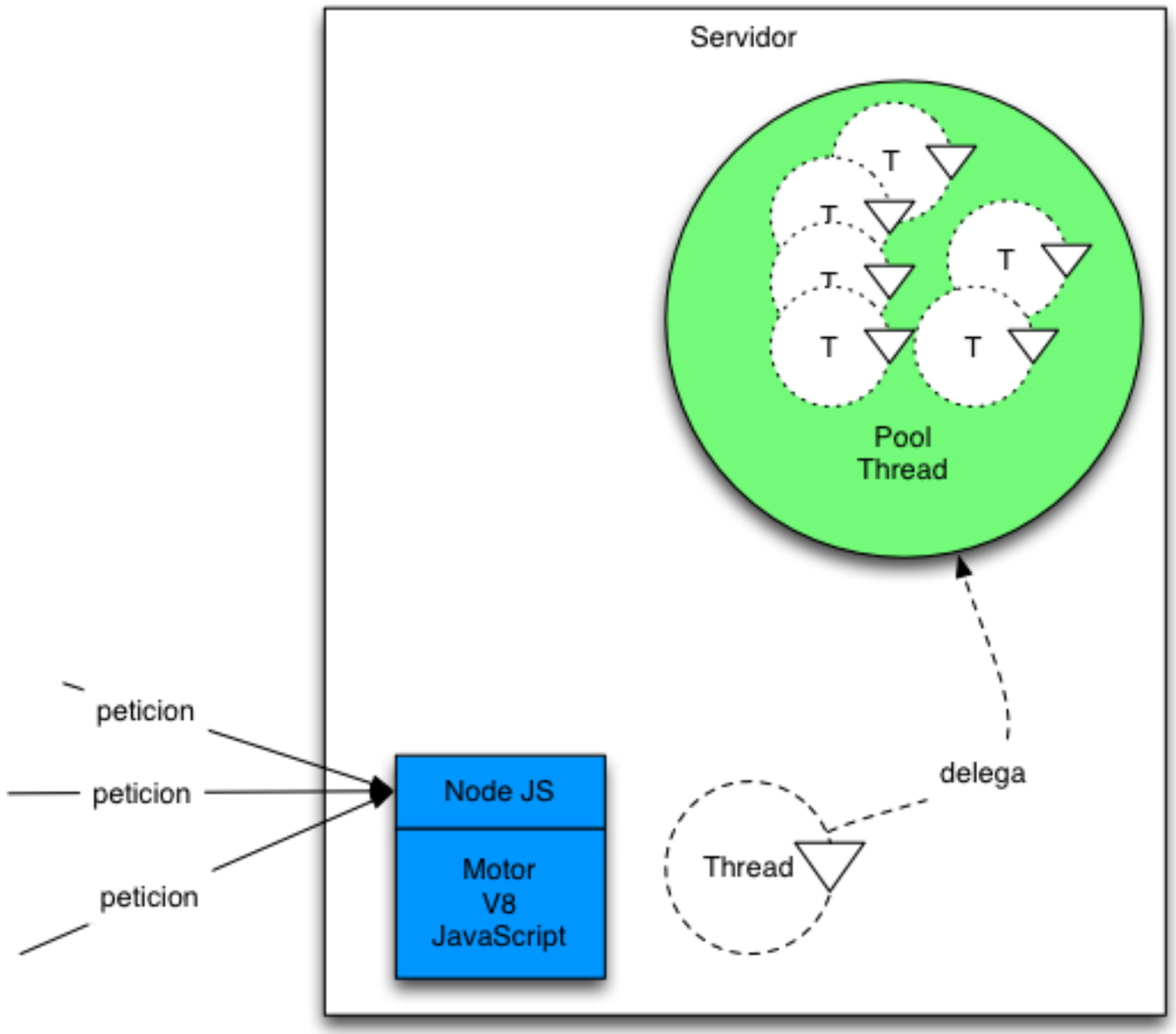
Un solo hilo?

Si una de las tareas que
e mandamos a realizar
toma mucho tiempo?

Respuesta:

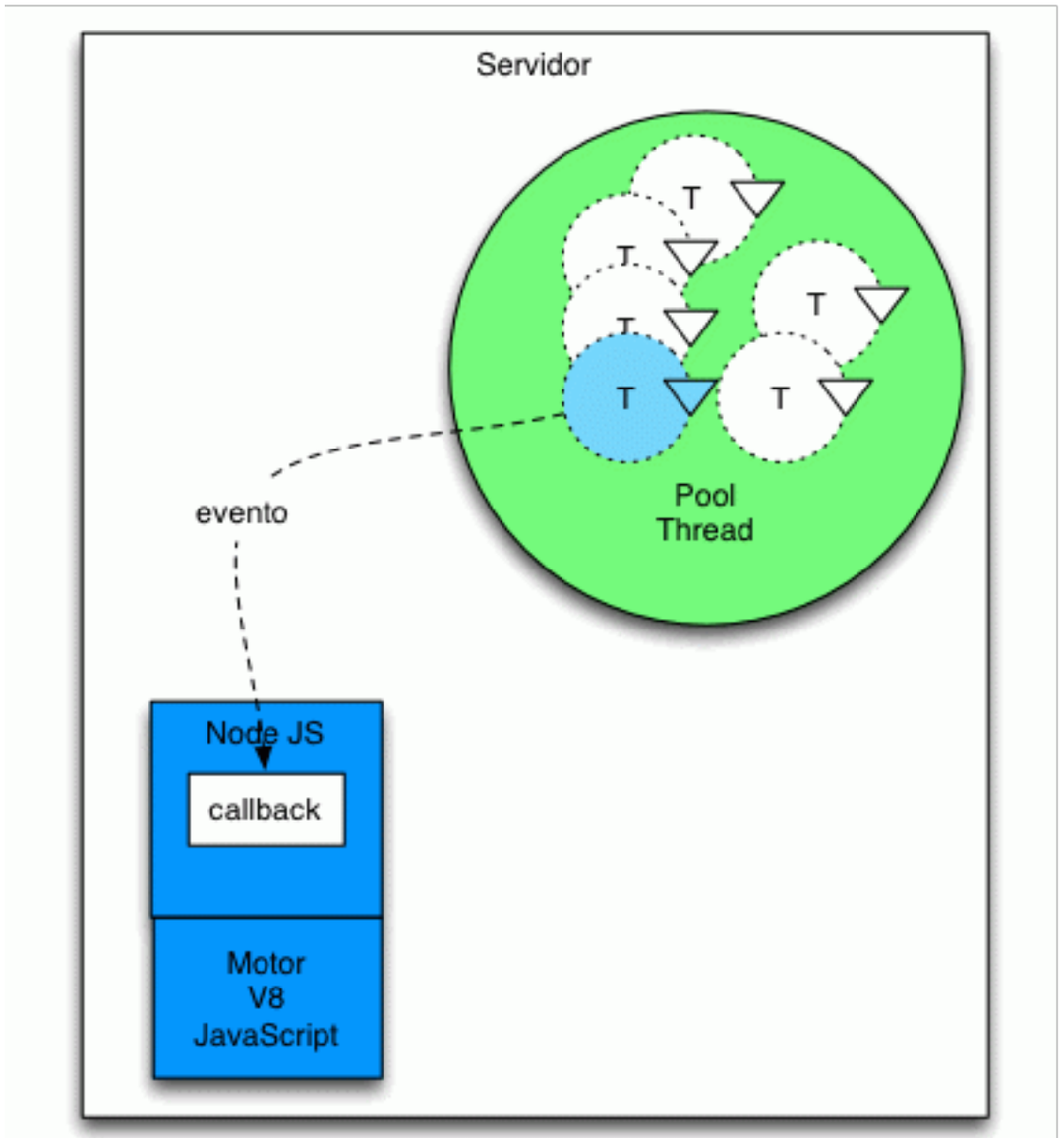
Ejecuciones Asíncronas

- Nodejs delega el trabajo a un pool de threads


Servidor
T
T
T
T
T
T
Pool
Thread
peticion
peticion
peticion
Node JS
Motor
V8
JavaScript
delega
Thread

libuv

- Dispone de su propio entorno de multi thread asíncrono.
- Nodejs envía el trabajo a realizar al pool.
- Libuv emitirá un evento que será recibido por Nodejs, una vez el trabajo haya sido terminado.


Servidor
T
T
T
T
T
T
Pool
Thread
evento
Node JS
callback
Motor
V8
JavaScript

Ejemplo:

node server.js

```
1   var http = require("http");
```

- Requiere al módulo http.
- Este viene incluído en Nodejs.
- Es accesible a través de la variable “http”.

```
3 ▼ http.createServer(function(request, response) {
```

- http crea un servidor a través de *createServer*.
- Esta función retorna un objeto.
- Este objeto tiene una función llamada *listen*.

```
7   }).listen(8888);
```

- *listen* recibe un valor numérico que representa al puerto que http va a escuchar.

Otra forma:

```
var http = require("http");
var server = http.createServer();
server.listen(8888);
```

Orientado al evento

Callbacks!

Usando módulos de Nodejs

Ejercicio

Realizar un chat con una sala global

Pasos

```
node --version
```

crear un archivo llamado package.json

```
{
  "name": “ejemplo-socket-chat",
  "version": "0.0.1",
  "description": “mi primera socket.io app",
  "dependencies": {}
}
```

Popular las dependencias para que npm las instale por nosotros

npm install --save express@4.10.2

crearse un archivo

index.js

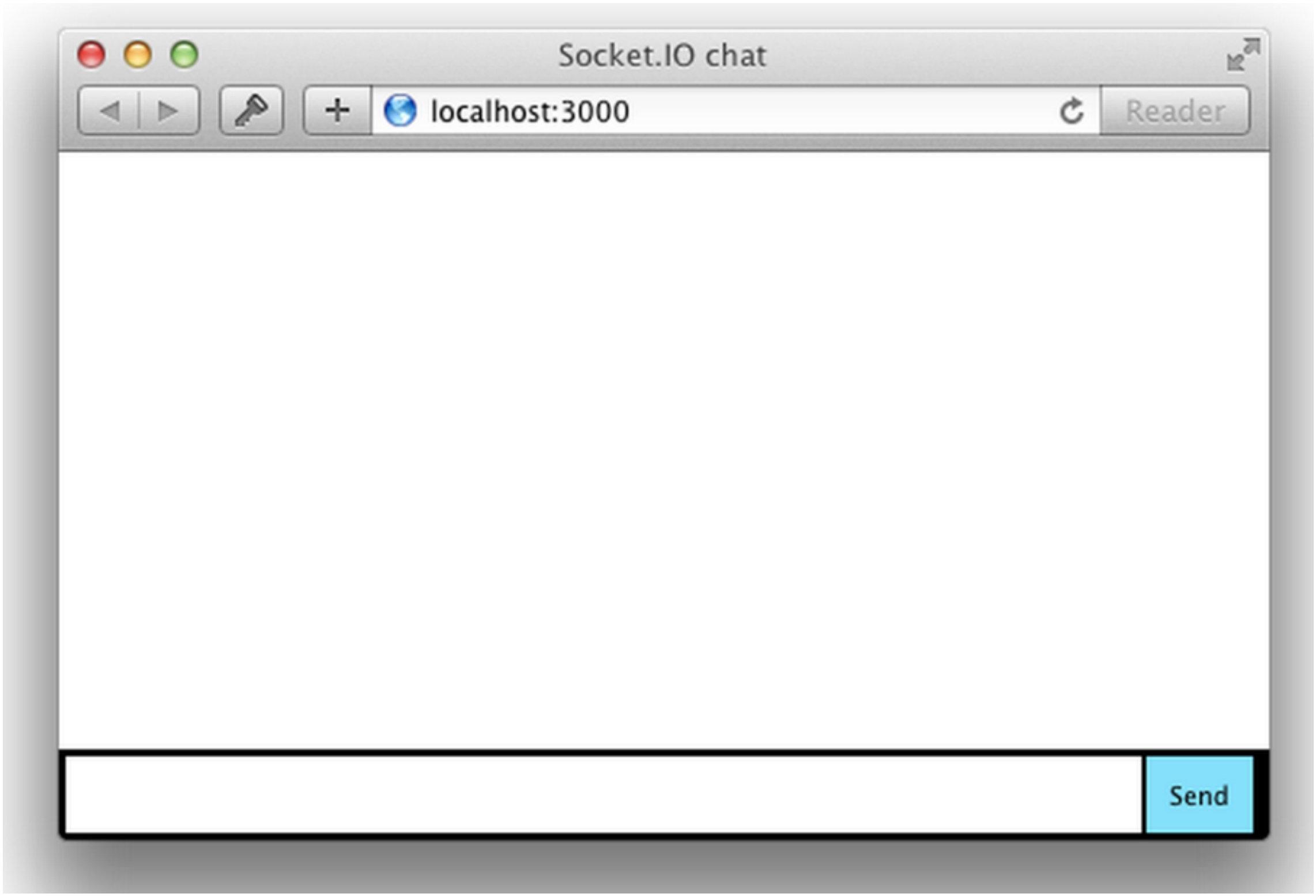
```
var app = require('express')();
var http = require('http').Server(app);

app.get('/', function(req, res){
  res.send(‘<h1>Hola Mundo!</h1>’);
});

http.listen(3000, function(){
  console.log('listening on *:3000');
});
```

node index.js

Codificando el chat


Socket.IO chat
localhost:3000
Reader
Send

npm install --save socket.io

- Añadir soporte de nicknames.
- Mostrar que un usuario está escribiendo un mensaje.
- Mostrar que un usuario se conecta o desconecta por IP (socket.handshake.address).

Gracias

Comentarios, preguntas, sugerencias, críticas…?