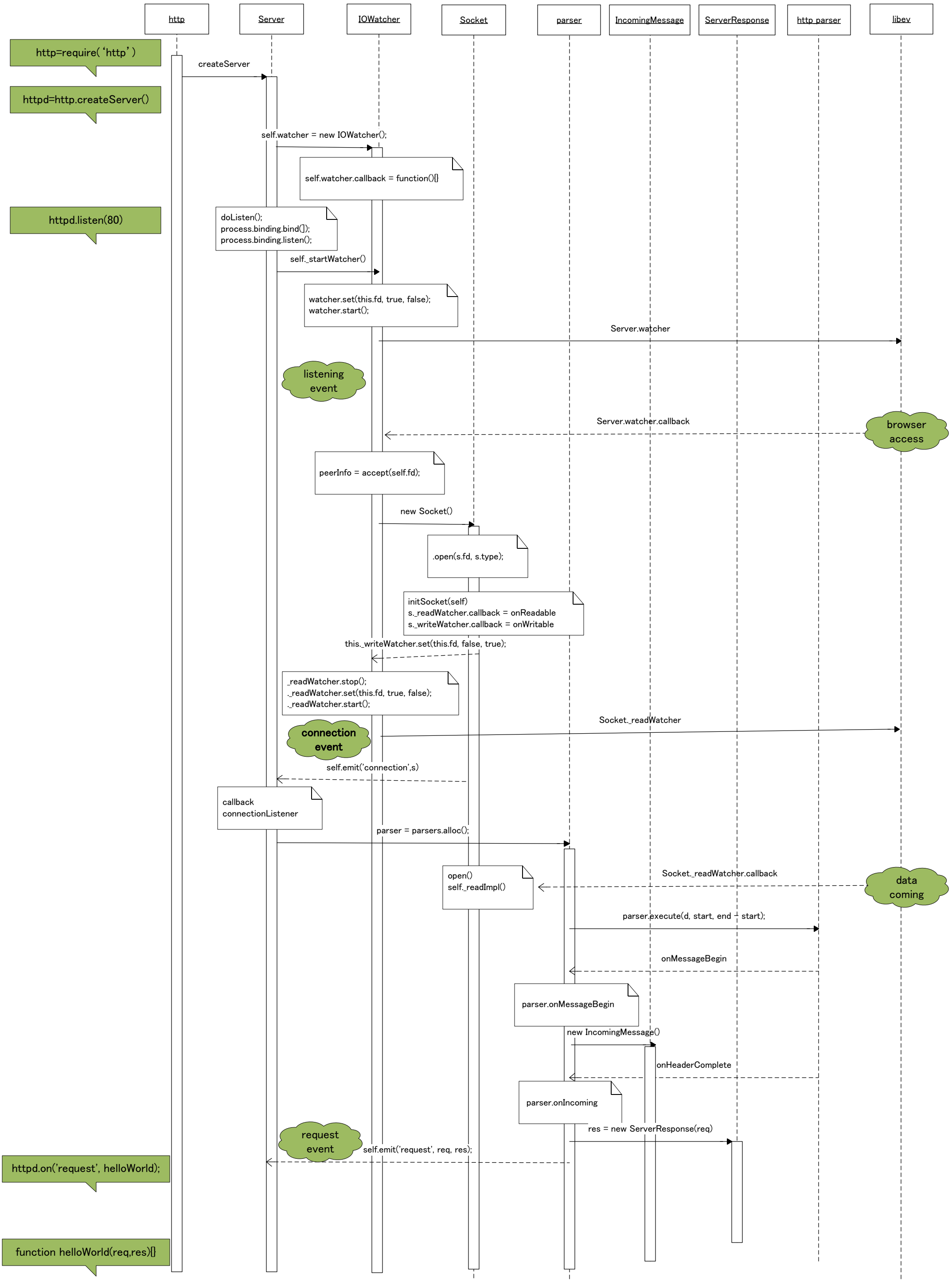
http
Server
IOWatcher
Socket
parser
IncomingMessage
ServerResponse
http parser
libev
http=require('http')
createServer
httpd=http.createServer()
self.watcher = new IOWatcher();
self.watcher.callback = function(){}
doListen();
process.binding.bind([]);
process.binding.listen();
httpd.listen(80)
self._startWatcher()
watcher.set(this.fd, true, false);
watcher.start();
Server.watcher
listening event
Server.watcher.callback
browser access
peerInfo = accept(self.fd);
new Socket()
.open(s.fd, s.type);
initSocket(self)
s._readWatcher.callback = onReadable
s._writeWatcher.callback = onWritable
this._writeWatcher.set(this.fd, false, true);
_readWatcher.stop();
._readWatcher.set(this.fd, true, false);
._readWatcher.start();
Socket._readWatcher
connection event
self.emit('connection',s)
callback connectionListener
parser = parsers.alloc();
open()
self._readImpl()
Socket._readWatcher.callback
data coming
parser.execute(d, start, end - start);
onMessageBegin
parser.onMessageBegin
new IncomingMessage()
onHeaderComplete
parser.onIncoming
res = new ServerResponse(req)
request event
self.emit('request', req, res);
httpd.on('request', helloWorld);
function helloWorld(req,res){}