

```
#!/usr/bin/env node

// File: ./bin/www.js

/**
 * Module dependencies.
 */

var app = require('..../app');
var debug = require('debug')('crime-viewer:server');
var http = require('http');

/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

/**
 * Create HTTP server.
 */

var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  var port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}

/**
```

```
55 * Event listener for HTTP server "error" event.
56 */
57
58 function onError(error) {
59   if (error.syscall !== 'listen') {
60     throw error;
61   }
62
63   var bind = typeof port === 'string'
64     ? 'Pipe ' + port
65     : 'Port ' + port;
66
67   // handle specific listen errors with friendly messages
68   switch (error.code) {
69     case 'EACCES':
70       console.error(bind + ' requires elevated privileges');
71       process.exit(1);
72       break;
73     case 'EADDRINUSE':
74       console.error(bind + ' is already in use');
75       process.exit(1);
76       break;
77     default:
78       throw error;
79   }
80 }
81
82 /**
83  * Event listener for HTTP server "listening" event.
84  */
85
86 function onListening() {
87   var addr = server.address();
88   var bind = typeof addr === 'string'
89     ? 'pipe ' + addr
90     : 'port ' + addr.port;
91   debug('Listening on ' + bind);
92 }
93
```

