

## Instructions for Running the CSV Parser:

Here are the instructions for running the csv parser located in:

/GTA\_bySelfieSisters/node-csv-parser/parser.js

### Starting up:

The parser comes with the database and the packages already present in its working directory.

You shouldn't have to download anything else to run the parser.js file. All you need to do is configure the application to your database's url.

Make sure you have [MongoDB](#), and [NodeJS](#). installed. For additional details see Jill's presentation slides about Mongo DB and Robo3T (11/1 in the presentations folder).

### Using the Application:

To use the application you just need to replace the URL variable at the top of the parser.js file with the URL to your Mongo database.

```
const fs = require('fs')
var parse = require('csv-parse')
var MongoClient = require('mongodb').MongoClient;
var url = "";
```

url should be set to the url of your mongo database. Usually that would look like this:

```
"mongodb://localhost:27017/mydb"
```

```
mongodb://<hostname>:<mongo_port>/<name_for_the_database>
```

In the same directory is the “IHopeThisWorks.csv” file which contains all 23000+ systems Jeff provided to us. By running the parser.js file, it should format every line of the of system data into its own separate dictionaries and then insert that dictionary into your database at:

**“TheThingThatContainsAllOfTheImportantCSV\_Stuff”**

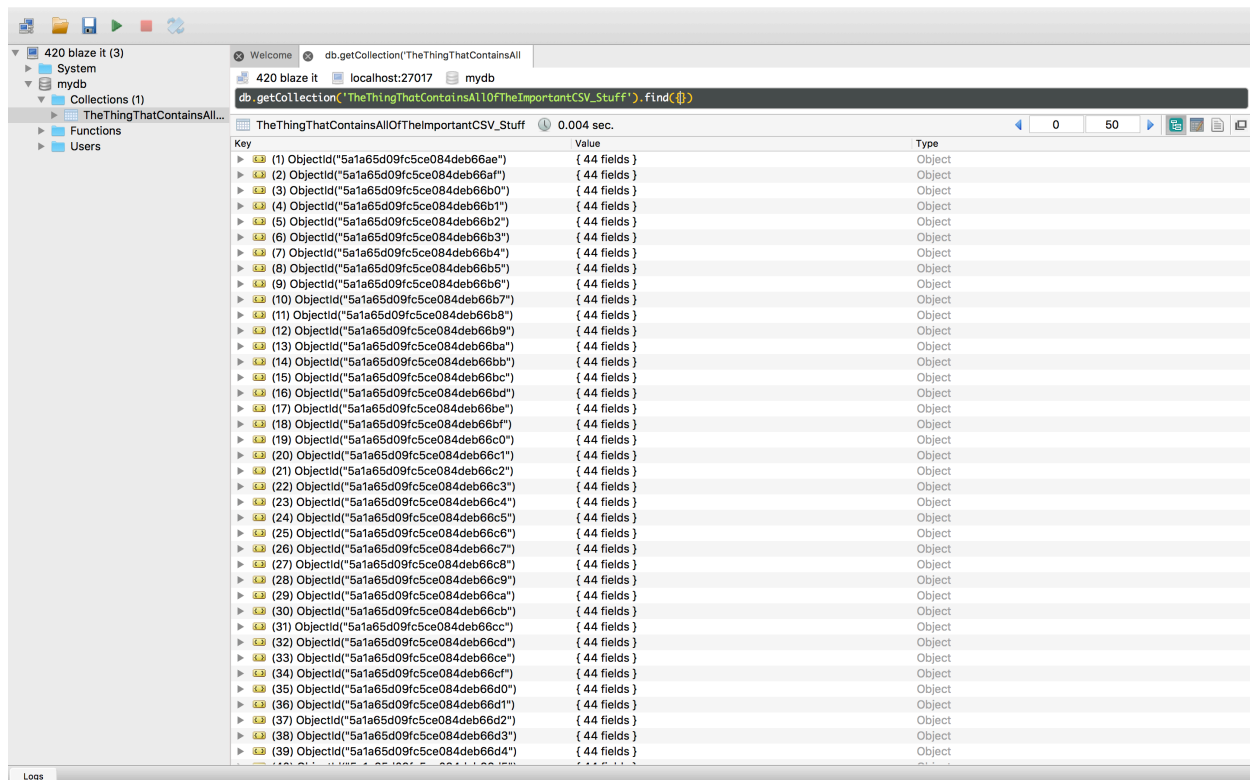
collection.

To run the application, run the parser.js file like any other Node application

```
Users-MBP:node-csv-parser user$ node parser.js
```

If the application runs successfully it should say “VAMOS. BOOMSHAKALAKA” 23000+ times

In Robo3T the end result should look like this:



Key	Value	Type
(1) ObjectId("5a1a65d09fc5ce084deb66ae")	{ 44 fields }	Object
(2) ObjectId("5a1a65d09fc5ce084deb66af")	{ 44 fields }	Object
(3) ObjectId("5a1a65d09fc5ce084deb66b0")	{ 44 fields }	Object
(4) ObjectId("5a1a65d09fc5ce084deb66b1")	{ 44 fields }	Object
(5) ObjectId("5a1a65d09fc5ce084deb66b2")	{ 44 fields }	Object
(6) ObjectId("5a1a65d09fc5ce084deb66b3")	{ 44 fields }	Object
(7) ObjectId("5a1a65d09fc5ce084deb66b4")	{ 44 fields }	Object
(8) ObjectId("5a1a65d09fc5ce084deb66b5")	{ 44 fields }	Object
(9) ObjectId("5a1a65d09fc5ce084deb66b6")	{ 44 fields }	Object
(10) ObjectId("5a1a65d09fc5ce084deb66b7")	{ 44 fields }	Object
(11) ObjectId("5a1a65d09fc5ce084deb66b8")	{ 44 fields }	Object
(12) ObjectId("5a1a65d09fc5ce084deb66b9")	{ 44 fields }	Object
(13) ObjectId("5a1a65d09fc5ce084deb66ba")	{ 44 fields }	Object
(14) ObjectId("5a1a65d09fc5ce084deb66bb")	{ 44 fields }	Object
(15) ObjectId("5a1a65d09fc5ce084deb66bc")	{ 44 fields }	Object
(16) ObjectId("5a1a65d09fc5ce084deb66bd")	{ 44 fields }	Object
(17) ObjectId("5a1a65d09fc5ce084deb66be")	{ 44 fields }	Object
(18) ObjectId("5a1a65d09fc5ce084deb66bf")	{ 44 fields }	Object
(19) ObjectId("5a1a65d09fc5ce084deb66c0")	{ 44 fields }	Object
(20) ObjectId("5a1a65d09fc5ce084deb66c1")	{ 44 fields }	Object
(21) ObjectId("5a1a65d09fc5ce084deb66c2")	{ 44 fields }	Object
(22) ObjectId("5a1a65d09fc5ce084deb66c3")	{ 44 fields }	Object
(23) ObjectId("5a1a65d09fc5ce084deb66c4")	{ 44 fields }	Object
(24) ObjectId("5a1a65d09fc5ce084deb66c5")	{ 44 fields }	Object
(25) ObjectId("5a1a65d09fc5ce084deb66c6")	{ 44 fields }	Object
(26) ObjectId("5a1a65d09fc5ce084deb66c7")	{ 44 fields }	Object
(27) ObjectId("5a1a65d09fc5ce084deb66c8")	{ 44 fields }	Object
(28) ObjectId("5a1a65d09fc5ce084deb66c9")	{ 44 fields }	Object
(29) ObjectId("5a1a65d09fc5ce084deb66ca")	{ 44 fields }	Object
(30) ObjectId("5a1a65d09fc5ce084deb66cb")	{ 44 fields }	Object
(31) ObjectId("5a1a65d09fc5ce084deb66cc")	{ 44 fields }	Object
(32) ObjectId("5a1a65d09fc5ce084deb66cd")	{ 44 fields }	Object
(33) ObjectId("5a1a65d09fc5ce084deb66ce")	{ 44 fields }	Object
(34) ObjectId("5a1a65d09fc5ce084deb66cf")	{ 44 fields }	Object
(35) ObjectId("5a1a65d09fc5ce084deb66d0")	{ 44 fields }	Object
(36) ObjectId("5a1a65d09fc5ce084deb66d1")	{ 44 fields }	Object
(37) ObjectId("5a1a65d09fc5ce084deb66d2")	{ 44 fields }	Object
(38) ObjectId("5a1a65d09fc5ce084deb66d3")	{ 44 fields }	Object
(39) ObjectId("5a1a65d09fc5ce084deb66d4")	{ 44 fields }	Object