**Converting CSV to JSON and Loading it to Postgres with Node.js**

To convert csv to json, Node has an awesome module, [csvtojson](https://www.npmjs.com/package/csvtojson). It takes a json file and convert it to csv asynchronously. Once we convert csv to json, let’s load it to a Postgres table with jsonb data type. Postgres supports JSON data and you can query it (see the previous blog about ingesting json into Postgres [here](https://www.mydatahack.com/new-json-data-ingestion-strategy-by-using-the-power-of-postgres/)). We are going to use [pg-copy-streams](https://github.com/brianc/node-pg-copy-streams) to bulk load the json file into Postgres (see [Bulk Loading Postgres with Node.js](https://www.mydatahack.com/bulk-loading-postgres-with-node-js/)).

**Setup**

Let’s create a project folder and install all the required module.

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

mkdir node-json-load  
cd node-json-load  
npm init  
npm i pg -ES  
npm i pg-copy-streams -ES

npm i csvtojson -ES

[/cc]

We also need to create a table in Postgres with the jsonb data type.

[cc lang="sql" tab\_size="4" lines="-1" theme="mac-classic"]

Create Table usermanaged.customer (

Data jsonb

);

[/cc]

**Code**

We are getting the connection details from config.json. See [Bulk Loading Postgres with Node.js](https://www.mydatahack.com/bulk-loading-postgres-with-node-js/) for further information on how to load data into Postgres.

The key is to call executeQuery() after calling writeFileSync(). This is because writeFileSync() is a synchronous function and the query gets executed only after completing the json file creation. The json file creating happens as a callback function. JSON conversion happens asynchronously. After it is completed with the ‘done’ event, we can start writing the converted results into a file.

<pre class="theme:vs2012-black font:verdana font-size:14px toolbar:2 lang:js decode:true ">

</pre>

Fun!