

CSV Parser

The CSV parser located at `parser/csv_parser.py` converts the data files that comply with the CSV format guide in to cytoscape data at build time so that the web application only has to deal with the cytoscape data when running it. Below will document that process on how this is done.

To run the parser navigate in to `parser/` and run `python csv_parser.py`. Please ensure you are using Python 3.6.

Contents

- File Reading
- Conversion to Objects
- Formatting for Cytoscape
- Output Generation

File Reading

As outlined in the format guide, you will need to ensure the data is stored inside `parser/` as follows:

```
-- parser
  \-- data
    |-- 2016
    \-- 2017
      |-- nodes
        |-- person.csv
        \-- roles.csv
      \-- specialNodes
        |-- project.csv
        \-- school.csv
```

The parser will read the data folder and look for all of the *year* directories it can find. There must be at least one year folder inside the data. It reads the strings of each of these directories stores it for use when creating the cytoscape data and outputting a file of all the years included.

Inside a year folder, there must be a nodes and a specialNodes directory. Nodes must contain a person.csv and roles.csv file. The will read the file names of the specialNodes files in order to obtain the "definition" of their type. In the example above, "project", and "school" are the types. These names will be used when making connections with person.csv data.

Once the file data is read, we can create the keys which will be displayed on the site.

Conversion to Objects

The purpose of turning the data in to objects first is so that they are easier to deal with and can be formatted to cytoscape data easily. The objects are Node, Edge, and Key. A global ID is maintained to ensure each object has a unique ID. The ID of nodes will change if new data is added. This is important to know if you depend on certain nodes maintaining an ID.

For every year we need to read, we first create the special nodes. Special nodes are the simplest to create as their files do not contain connection information. After, normal nodes are created. Whilst doing this, we pass the special nodes that we just created for that year to the method so that special edges can be made their. Each of these functions pass in mutable lists of allNodes and allEdges to simplify the methods.

Once all the special and person nodes are read, and connections to special nodes are established, we create the normal edges. These are considered the "collab" edges in which person nodes connect to other person nodes.

Formatting for Cytoscape

This is the simplest part to follow. All of the objects created are done so that they can be printed to cytoscape format simply. Read up on element data here if you wish to know more about the output structure.

Output Generation

The output of the parser ends up in `parser/output/`. `output.json` and `years.json` will be generated. In order to use this in your application you need to move it inside `src/assets/` if you want to test it locally.