

## JSON Format Technical Description (what your JSON file should look like)

```
{
  "STUDENTS": [
    {
      "DEMOGRAPHICS": {
        "gender": "Male",
        "ethnicity": "Hispanic"
      },

      "RECORDS": [
        {
          "TIME": "Fall 2014",
          "GROUPS": [
            {"status": "Enrolled"},
            {"discipline": "BIOSC"},
            {"major": "BBMI"}
          ]
        },

        {
          "TIME": "Fall 2015",
          "GROUPS": [
            {"status": "Enrolled"},
            {"discipline": "BIOSC"},
            {"major": "BBMI"}
          ]
        }
      ]
    },

    {
      "DEMOGRAPHICS": {
```

```

    "gender": "Female",
    "ethnicity": "White"
  },

  "RECORDS": [
    {
      "TIME": "Fall 2014",
      "GROUPS": [
        {"status": "Enrolled"},
        {"discipline": "BIOSC"},
        {"major": "BBMI"}
      ]
    },

    {
      "TIME": "Fall 2015",
      "GROUPS": [
        {"status": "Enrolled"},
        {"discipline": "BIOSC"},
        {"major": "BCHE"}
      ]
    }
  ]
}

```

**Notes:**

- The "DEMOGRAPHICS" object is optional. If provided, a filter menu will be generated based on the datatypes of the values of the "DEMOGRAPHICS" object attributes.
- The keys (quoted words before each colon) that are CAPITALIZED must be included (except "DEMOGRAPHICS") and the spelling and capitalization must match the example above.

- The keys that are lower-case can be customized. (i.e. the “status” key in the example above could be replaced with “loves ice cream” and it’s values could be “yes!”)
- The "RECORDS" array should be chronological order.
- The "GROUPS" array should be ordered according to the grouping heirarchy. (the example below has a student in the group "Enrolled in the discipline of BIOSC in the major of BBMI").

## JSON FAQ

For detailed information about the JSON standard: [www.json.org](http://www.json.org)

**Pay careful attention to where the example uses square brackets [ ] and where it uses curly braces { }.**

Curly brackets denote an object with information stored in KEY : VALUE pairs. The order of the keys in an object are not guaranteed to stay the same after being parsed. Browsers do funny things :(

Square brackets denote an array which is an ORDERED list of information.

This is very important because if you want to use informative values for "TIME" for your institution then I can't just sort alphabetically or use some complicated rule for sorting the "TIME" descriptors. I have to rely on the order of the information you provide.

This is also very important in the "GROUPS" array because the order denotes the hierarchy of groups.

**Do not leave trailing commas in arrays or objects.**

For example:

```
{
    "key": "object",
    "key": "number",
}
```

Will not parse correctly in some browsers because of the red comma.

**Strings cannot have line-breaks.**

Strings are text enclosed in double quotes. (i.e. "DEMOGRAPHICS"). The text within double quotes cannot have line-breaks.

For example:

```
{  
  "A JSON KEY": "A STRING ON ONE LINE THAT IS GOOD",  
  "ANOTHER JSON KEY":  
    "A STRING  
    ON MULTIPLE LINES THAT IS BAD"  
}
```

Outside of strings you can use line-breaks, spaces, and tabs wherever you want. In fact, if you want your file to be more human-readable you should use line-breaks and indentation to make the structure of your data more clear.