Meeting 6

Updates

Wrote SOC API Parser to follow JSON mapping, allowing for easier changes and additions.

Changes to the DB Schema shouldn't require changes to the code base
And if they do they should be very small.

Instead changes or new collections can be added to the JSON Mapping and will automatically be picked up by the parser.

Mappings

```
{
    "<collection_name_1>" : {
        "parent_keys" : [
            "<parent_key_1>",
            "<parent_key_2>",
            "<parent_key_3>",
        ],
        "keys" : {
            "<key_name_1>" : {
                "new_key" : "<new_key>",
                "key_mod_method" : "<key_mod_method_name_1>",
                "value_mappings" : {
                    "<SOC_val_1>" : "<NEW_API_val_1>",
                    "<SOC_val_2>" : "<NEW_API_val_2>",
                },
                "augmented_keys" : ["<augmented_key_1>", "<augmented_key_2>"],
            },
            "<key_name_2>" : {
                "new_key" : "<new_key>",
                "key_mod_method" : "<key_mod_method_name_1>",
                "value_mappings" : {
                    "<SOC_val_1>" : "<NEW_API_val_1>",
                    "<SOC_val_2>" : "<NEW_API_val_2>",
                },
                "augmented_keys" : ["<augmented_key_1>", "<augmented_key_2>"],
            },
        },
    },
}
```

- Collection Name *Required*

- Parent Keys (Location of key inside nested SOC JSON)
  Key is only present if they are required
  Absent Key is equivalent to 'parent_keys : None'
  In order they will be counter in tree/dict
- Keys (SOC -> New API) *SHOULD NEVER BE NONE Value is only present if they are required Absent Value is equivalent to '"new_key" : None,"value_mappings" : None'*
  - Name (Data Name inside SOC JSON)
  - Key Mod Method (Method name that properly formats key data)
  - Mapping (New Mapped Name of Data)
    Key is only present if they are required
    Absent Key is equivalent to 'value_mappings : None'
  - Augmented Keys (Additional names for this key)

Issues
Key Levels

- Nesting Issues that share references could become an issue.
  - ex: Times and instructors
    - both nested at the section level
    - But not all instructors have the same times.

Lots of data to handle later on

- Hopefully the schema and mod method can account for this.
- Time modifications
- Sub campuses
- Majors
- Minors
- etc.
  Need to have a better way of handling mod methods

DB

DB-PoC population is successful.
Reference Links are successful.

Connection & Parser

Needs to be modularized
Almost all of the course parsing, DB connection and pushing.
Need to be modularized so that testing is easier and there is less mess with DB connections

Query

Expansions written

Below is Output from a DB Query via Python:

Courses:

```
{
    "number": "112",
    "notes": null,
    "description": null,
    "synopsisUrl": "http://www.cs.rutgers.edu/undergraduate/courses/",
    "title": "DATA STRUCTURES",
    "preReqNotes": "((01:198:111  or 14:332:252 ) and (01:640:135 ))<em> OR </em> ((01:198:111  or 14:332:252 ) and (01:640:151 ))<em> OR </em> ((01:198:111  or 14:332:252 ) and (01:640:153 ))<em
> OR </em> ((01:198:111  or 14:332:252 ) and (01:640:191 ))",
    "credits": 4,
    "expandedTitle": null,
    "__subject__": {
        "number": "198"
    },
    "__professor__": [
        {
            "name": "VENUGOPAL"
        },
        {
            "name": "CENTENO"
        }
    ],
    "__campus__": {
        "name": "New Brunswick",
        "code": "NB"
    }
}
```

Professors:

```
{
    "name": "CENTENO",
    "__course__": [
        {
            "number": "112",
            "notes": null,
            "description": null,
            "synopsisUrl": "http://www.cs.rutgers.edu/undergraduate/courses/",
            "title": "DATA STRUCTURES",
            "preReqNotes": "((01:198:111  or 14:332:252 ) and (01:640:135 ))<em> OR </em> ((01:198:111  or 14:332:252 ) and (01:640:151 ))<em> OR </em> ((01:198:111  or 14:332:252 ) and (01:640:1
53 ))<em> OR </em> ((01:198:111  or 14:332:252 ) and (01:640:191 ))",
            "credits": 4,
            "expandedTitle": null
        },
        {
            "number": "494",
            "notes": null,
            "description": null,
            "synopsisUrl": "http://www.cs.rutgers.edu/undergraduate/courses/",
            "title": "INDEP STUDY COMP SCI",
            "preReqNotes": null,
            "credits": null,
            "expandedTitle": null
        }
    ],
    "__subject__": {
        "number": "198"
    },
    "__campus__": {
        "name": "New Brunswick",
        "code": "NB"
    }
}
```

Need to write more complex query handlers and define structure.

Determine endpoint handling (Replicate DB Project?)