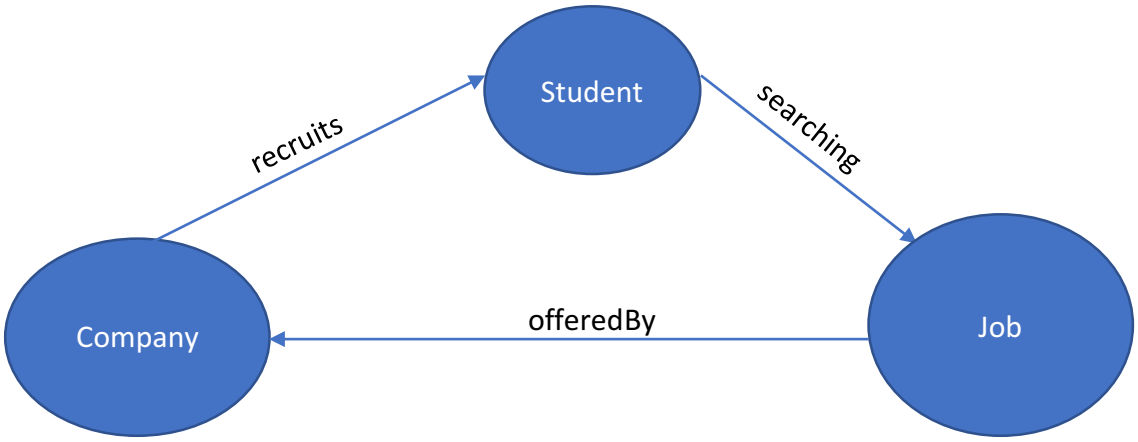
Mini Group Project – IBM Graph Database

Introduction –

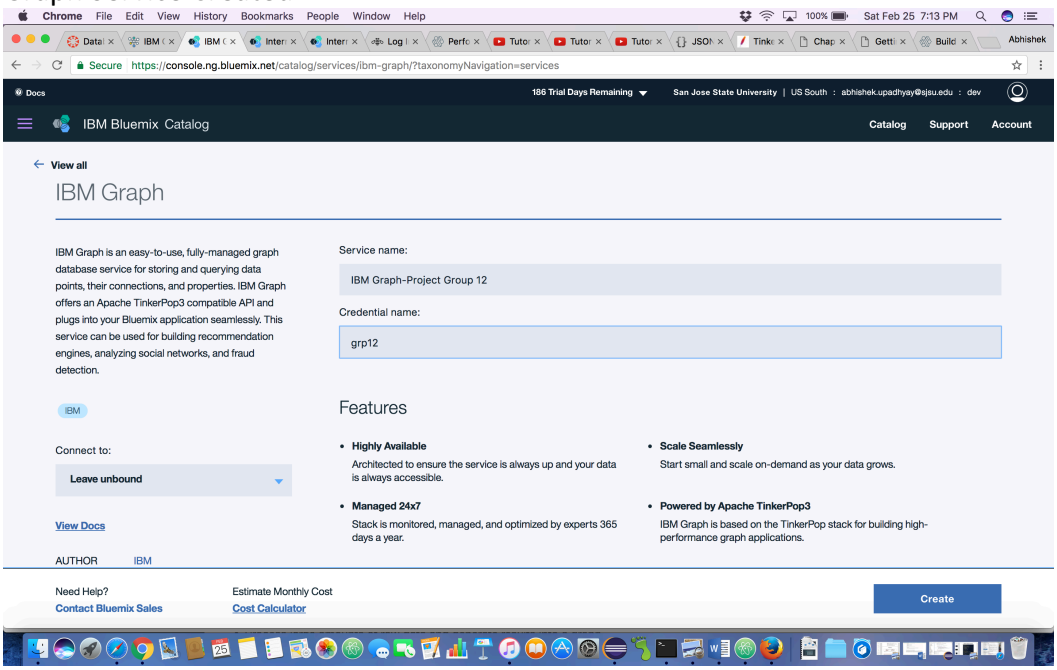
We have implemented a “Student Job Search & Recruitment” and the schema design is as below



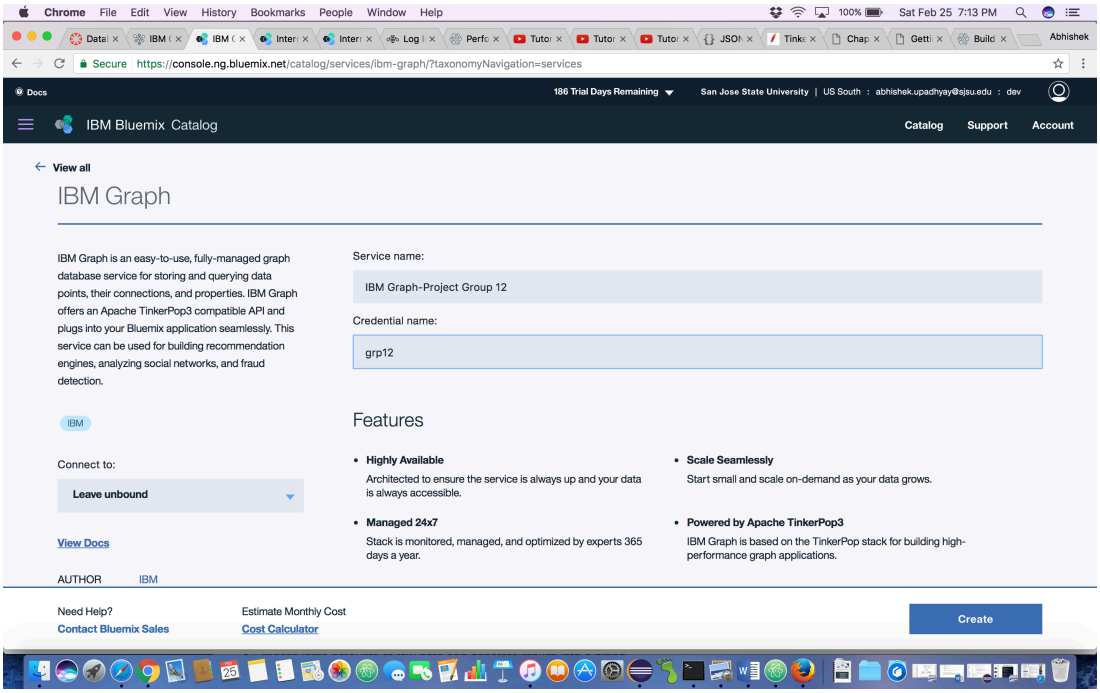
In this project we have understood and implemented the following:

- Schema, vertices, indexes, edges creation
- Traversing the graph through gremlin query interface

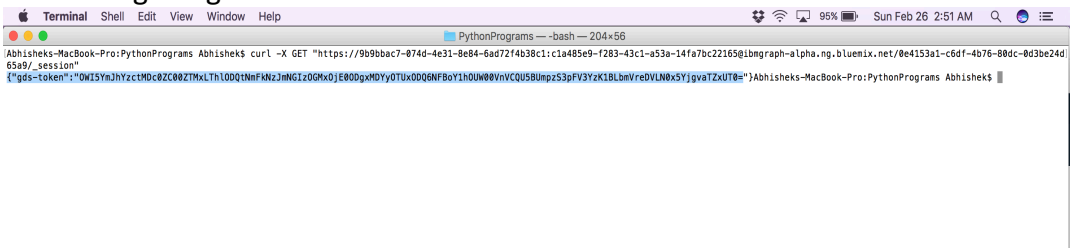
Graph Service Created



Getting the credentials



Generating the gd-token



Preparing this input schema

```
{
  "vertexLabels" : [
    {"name" : "Student"},
    {"name" : "Job"},
    {"name" : "Company"}
  ],
  "edgeLabels" : [
    {"name" : "searching" , "multiplicity" : "MULTI"},
    {"name" : "recruits"  , "multiplicity" : "MULTI"},
    {"name" : "offeredBy" , "multiplicity" : "MULTI"}
  ],
  "propertyKeys" :[
    {"name" : "student_Name"  , "dataType" : "String" , "cardinality" : "SINGLE" },
    {"name" : "student_Email"  , "dataType" : "String" , "cardinality" : "SINGLE"},
    {"name" : "job_title"  , "dataType" : "String" , "cardinality" : "SINGLE"},
    {"name" : "job_category"  , "dataType" : "String" , "cardinality" : "SINGLE"},
    {"name" : "job_location"  , "dataType" : "String" , "cardinality" : "SINGLE"},
    {"name" : "company_name"  , "dataType" : "String" , "cardinality" : "SINGLE"},
    {"name" : "company_location"  , "dataType" : "String" , "cardinality" : "SINGLE"},
  ],

  "vertexIndexes" : [
    {"name" : "studentByName" , "propertyKeys" : ["student_Name"] ,"composite" : true ,"unique" : false},
    {"name" : "companyByLocation" , "propertyKeys" : ["company_location"] ,"composite" : true , "unique" : false},
    {"name" : "jobsByLocation" , "propertyKeys" : ["job_location"] ,"composite" : true , "unique" : false}
  ]
}
```

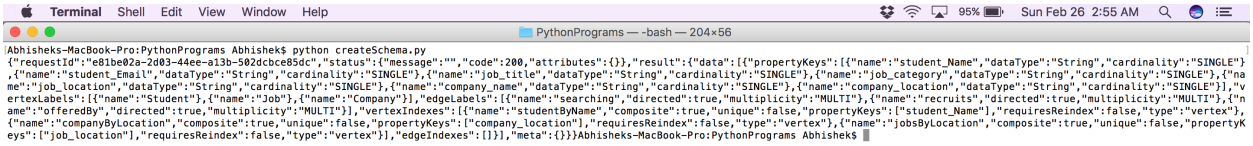
LF UTF-8 JSON

Posting and creating the input schema through curl command and schema API

```python
import subprocess

schemaFile = open("studentSchema.json" ,'rb').read()


subprocess.call([
    'curl',
    '-X',
    'POST',
    '-d',
    schemaFile,
    '-H',
    'content-type : application/json',
    '-H',
    'Authorization : gds-token N2Y2NDhiNWItN2IxZS00ODQ5LWI3MzktMTBlNmQ1NzMyMTE1OjE0ODgxMDUxODgxNjY6bkxDa0NvV2NQQWI1cWJHajlOV
    'https://ibmgraph-alpha.ng.bluemix.net/0e4153a1-c6df-4b76-80dc-0d3be24d65a9/g/schema'
])

```

LF UTF-8 Python

Schema Created

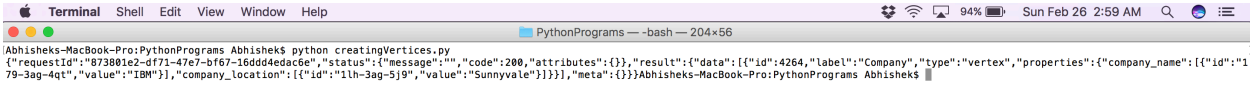


Creating vertex through vertices API

```python
import subprocess

vertexDataJsonQuery = '{ "label": "Company", "properties": {"company_name":"IBM", "company_location":"Sunnyvale"}}'

subprocess.call([
    'curl',
    '-X',
    'POST',
    '-d',
    vertexDataJsonQuery,
    '-H',
    'content-type : application/json',
    '-H',
    'Authorization : gds-token OWI5YmJhYzctMDc0ZC00ZTMxLThlODQtNmFkNzJmNGIzOGMxOjE0ODgxMDYyOTUxODQ6NFBoY1hOUW00VnVCQU5BUmpzS
    'https://ibmgraph-alpha.ng.bluemix.net/0e4153a1-c6df-4b76-80dc-0d3be24d65a9/g/vertices'
])

```

LF UTF-8 Python

Vertex Created



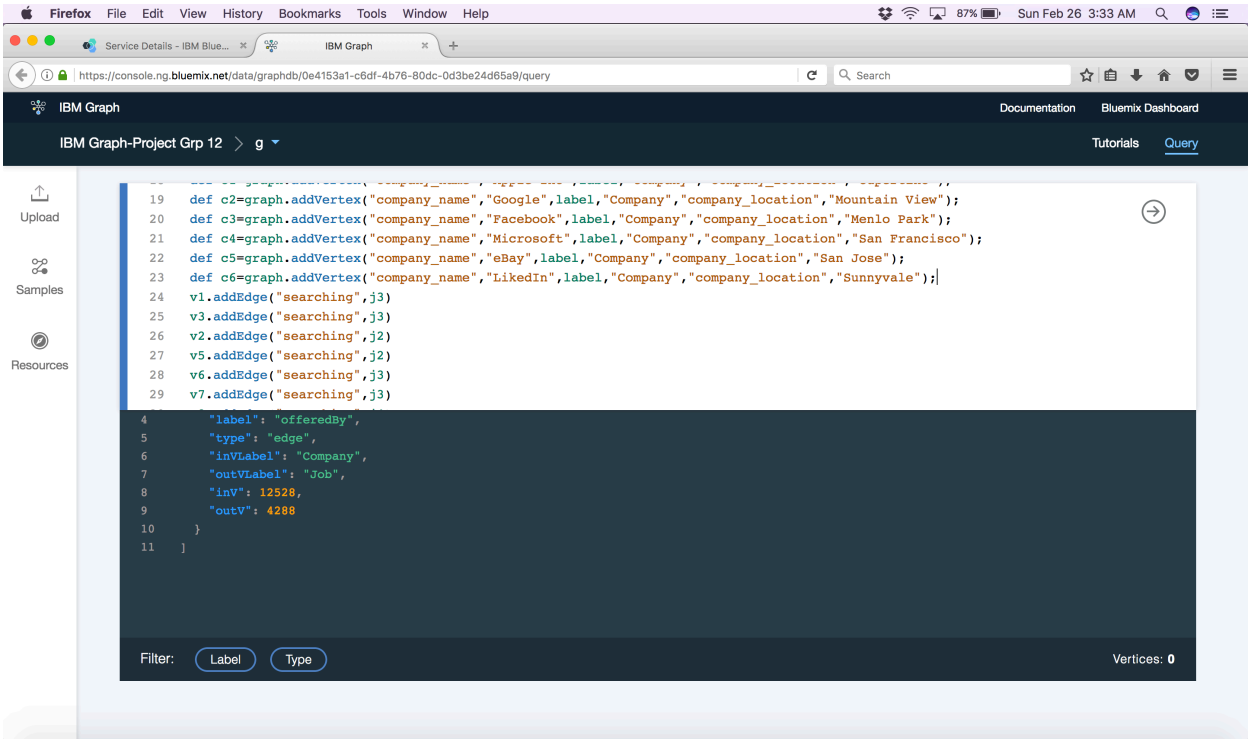
Creating Edges

```
import subprocess

edgeDataJsonQuery = '{ "outV" : 4336,"label": "recruits", "inV" : 4208 }'

subprocess.call([
    'curl',
    '-X',
    'POST',
    '-d',
    edgeDataJsonQuery,
    '-H',
    'content-type : application/json',
    '-H',
    'Authorization : gds-token OWI5YmJhYzctMDc0ZC00ZTMxLThlODQtNmFkNzJmNGIzOGMxOjE0ODgxMDYyOTUxODQ6NFBoY1hOUW00VnVCQU5BUmpzS3pFV3YzK1BLbmVreDVLN0x5YjgvaT
    'https://ibmgraph-alpha.ng.bluemix.net/0e4153a1-c6df-4b76-80dc-0d3be24d65a9/g/edges'
])
```

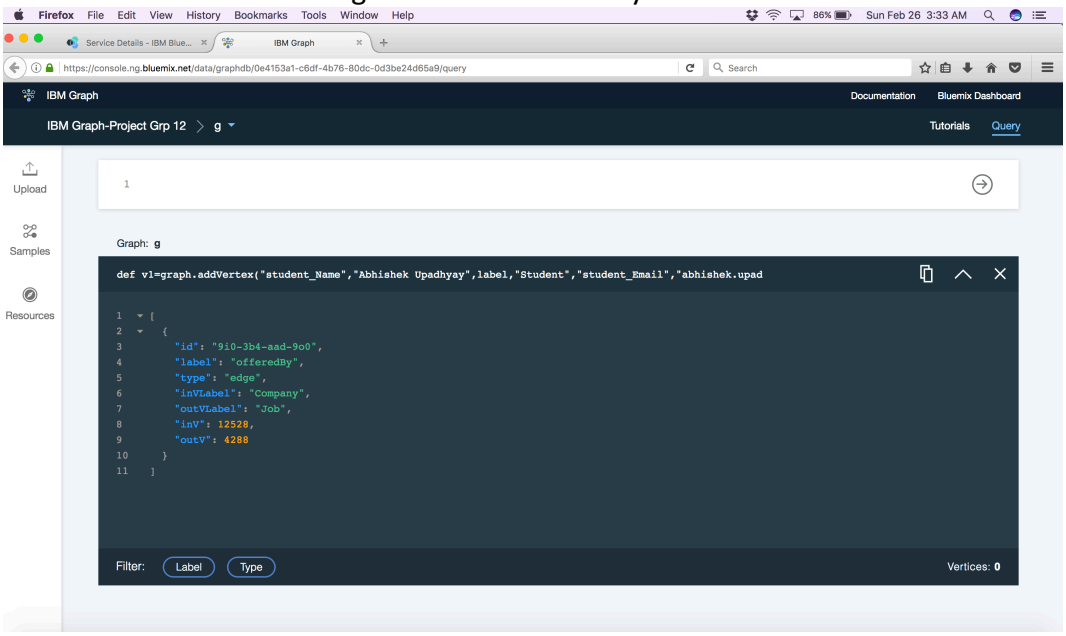
Edges Created

```
PythonPrograms — -bash — 204×56
Abhisheks-MacBook-Pro:PythonPrograms Abhishek$ python creatingEdges.py
{"requestId":"8b670b34-4878-419a-a4de-ba48ffc35e88","status":{"message":"","code":200,"attributes":{}},"result":{"data":[{"id":"odxce-38w-2l91-37k","label":"searching","type":"edge","inVLabel":"Job","outV
Label":"Student","inV":4160,"outV":4208}],"meta":{}}}Abhisheks-MacBook-Pro:PythonPrograms Abhishek$
```

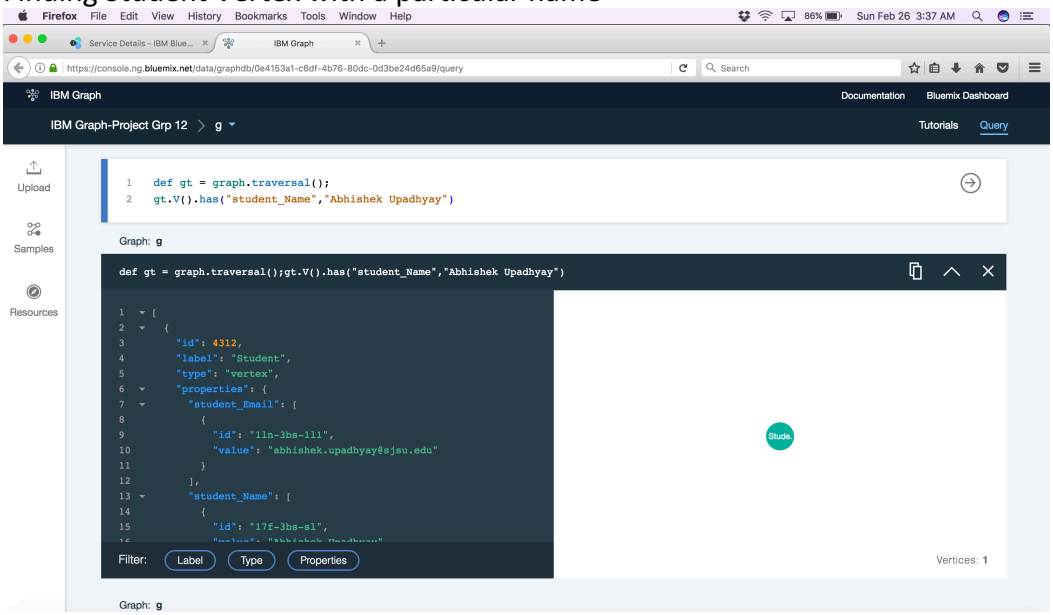
Adding some more data to vertices and edges through gremlin



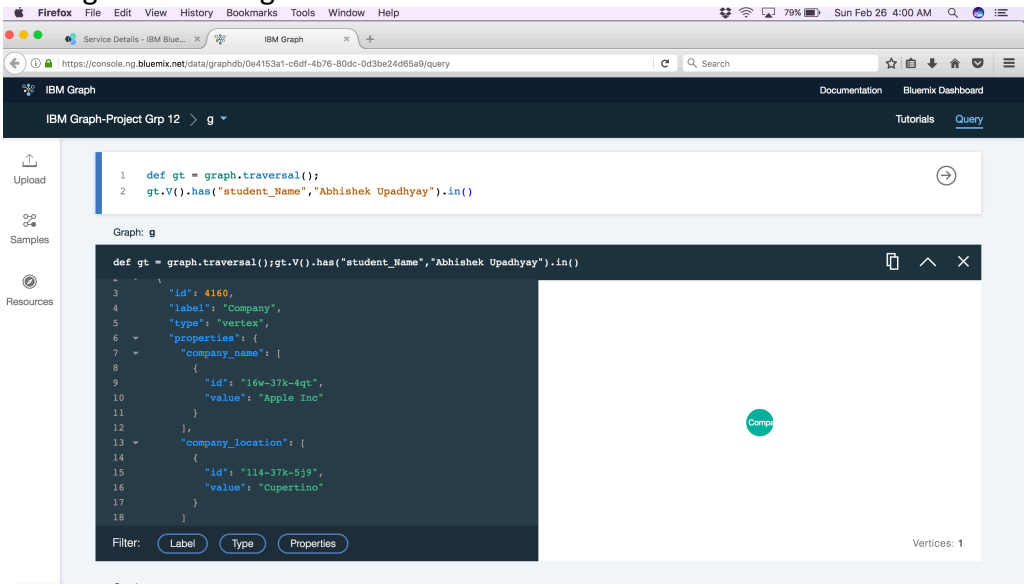
All data for vertices and edges added successfully



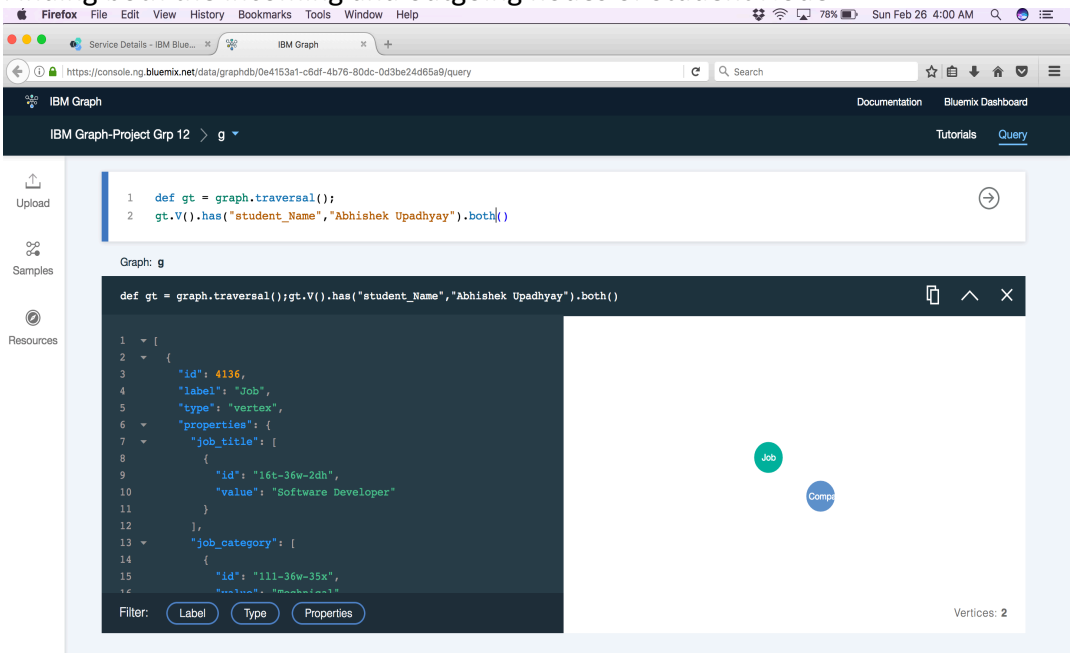
Finding Student Vertex with a particular name



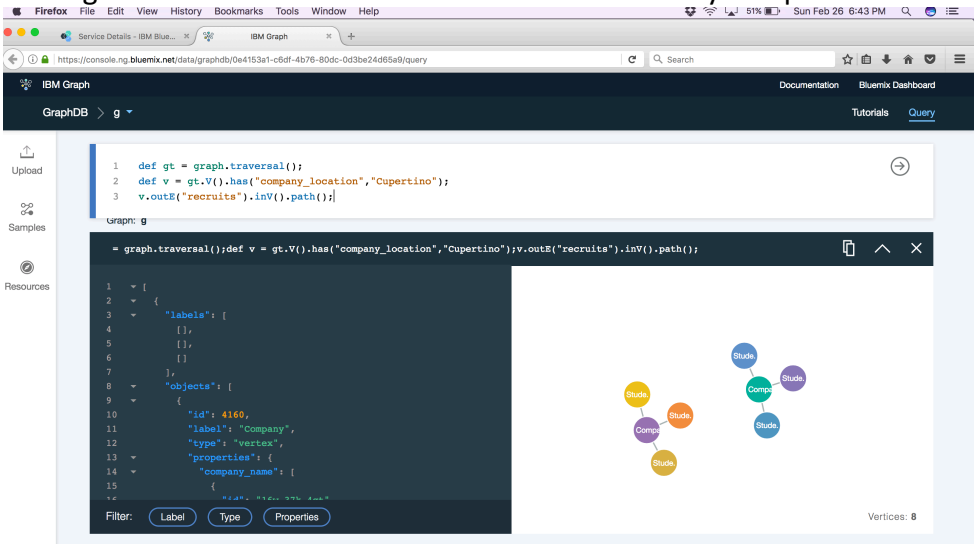
Finding the incoming node to Student node



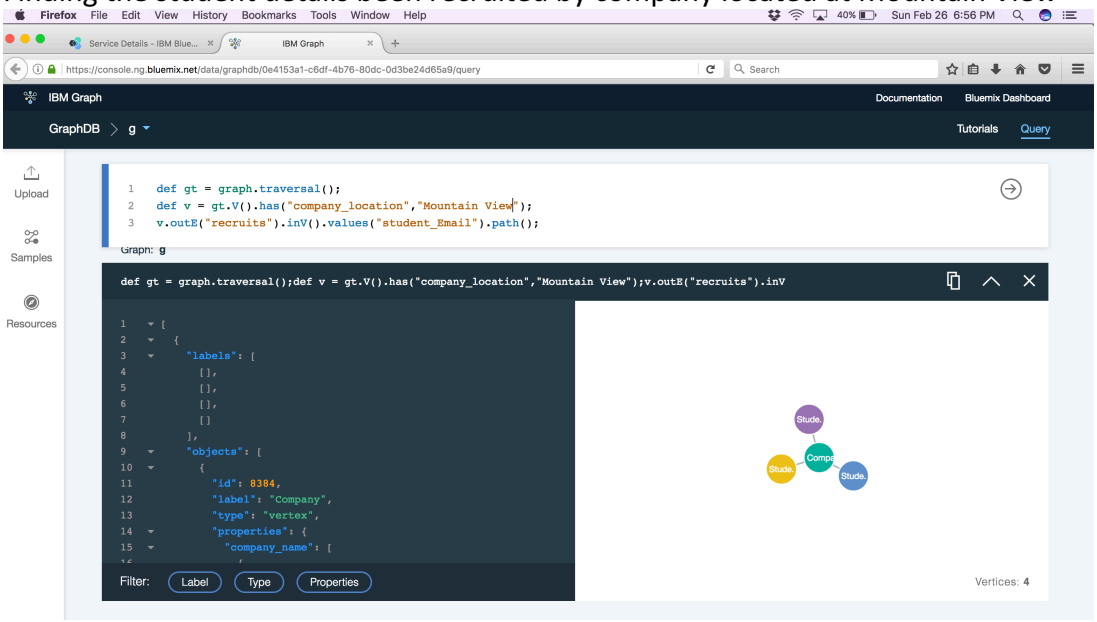
Finding both the incoming and outgoing nodes of Student Node



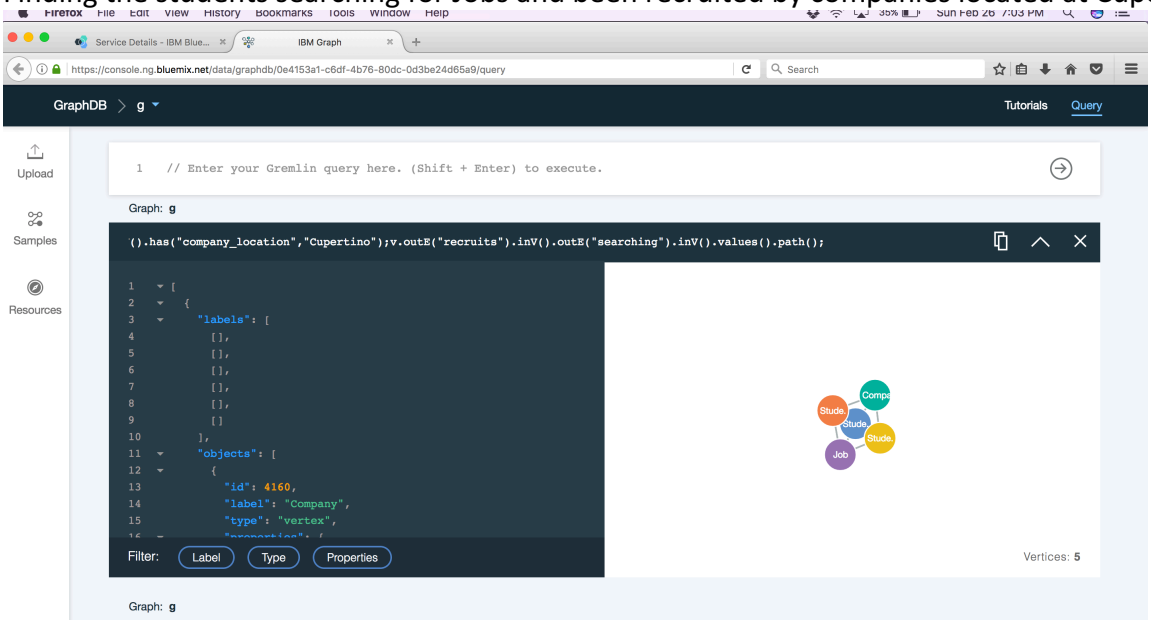
Finding the student details who have been recruited by companies based out of Cupertino



Finding the student details been recruited by company located at Mountain View



Finding the students searching for Jobs and been recruited by companies located at Cupertino



Finding the students searching for Jobs and been recruited by companies located at Mountain View