Creating an Org Hierarchy Example in Python

ddas.tech/org-hierarchy-example-in-python/

October 1, 2022

Created By: Debasis Das (Oct 2022)

In this sample we will create a simple org hierarchy in Python using sample Departments and Employees and providing a employee search functionality for the departments

Lets begin with the import statements

```
import json
from json2html import *
import webbrowser
import os
```

Now lets define the Department and Employee Classes respectively

```
class Department(dict):
    def __init__(self,deptId,name):
        super().__init__()
        self.__dict__ = self
        self.deptId = deptId
        self.name = name
        self.employees = []

    def addEmployee(self,employee):
        self.employees.append(employee)

    def searchEmployee(self,employeeName):
        found = False
        for item in self.employees:
            if ((item.firstName == employeeName) | (item.lastName == employeeName)):
                found = True
                return f"Record for {employeeName} found in {self.name} Department"
        return f"Record for {employeeName} was not found in {self.name} Department"


class Employee(dict):
    def __init__(self,firstName, lastName,empId,email):
        super().__init__()
        self.__dict__ = self
        self.firstName = firstName
        self.lastName = lastName
        self.empId = empId
        self.email = email
```

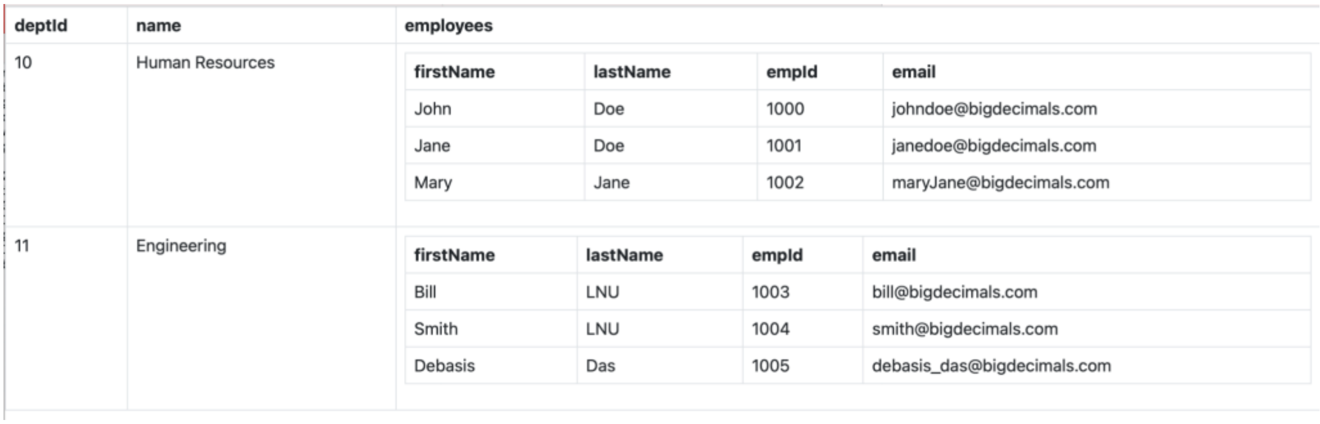
In the above block you can see that the search only searches the first name and last name against the search criteria, this can be enhanced to do partial text search against first name, last name and email as well.

Lets create some demo data

```
emp1 = Employee("John","Doe","1000","johndoe@bigdecimals.com")
emp2 = Employee("Jane","Doe","1001","janedoe@bigdecimals.com")
emp3 = Employee("Mary","Jane","1002","maryJane@bigdecimals.com")
emp4 = Employee("Bill","LNU","1003","bill@bigdecimals.com")
emp5 = Employee("Smith","LNU","1004","smith@bigdecimals.com")
emp6 = Employee("Debasis","Das","1005","debasis_das@bigdecimals.com")


dept1 = Department("10","Human Resources")
dept2 = Department("11","Engineering")
dept1.addEmployee(emp1)
dept1.addEmployee(emp2)
dept1.addEmployee(emp3)

dept2.addEmployee(emp4)
dept2.addEmployee(emp5)
dept2.addEmployee(emp6)

allDepts = [dept1,dept2]
```

Lets view the hierarchy in a nested table structure in HTML

```
formatted_table = json2html.convert(json = allDepts_JSON,table_attributes="id=\"info-
table\" class=\"table table-bordered\"")
fileName = 'dept.html'
html_string = '''
<html>
<head>
<title> HTML </title>
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
</head>
<body>
{tableData}
</body>
</html>
'''
fileIndex = open(fileName,"w")
fileIndex.write(html_string.format(tableData=formatted_table))
fileIndex.close()

fileFullPath = 'file:///'+os.getcwd()+'/'+fileName
webbrowser.open_new_tab(fileFullPath)
```



<table>
<tr><th>deptId</th><th>name</th><th>employees</th></tr>
<tr><td>10</td><td>Human Resources</td><td><table>
<tr><th>firstName</th><th>lastName</th><th>empId</th><th>email</th></tr>
<tr><td>John</td><td>Doe</td><td>1000</td><td>johndoe@bigdecimals.com</td></tr>
<tr><td>Jane</td><td>Doe</td><td>1001</td><td>janedoe@bigdecimals.com</td></tr>
<tr><td>Mary</td><td>Jane</td><td>1002</td><td>maryJane@bigdecimals.com</td></tr>
</table></td></tr>
<tr><td>11</td><td>Engineering</td><td><table>
<tr><th>firstName</th><th>lastName</th><th>empId</th><th>email</th></tr>
<tr><td>Bill</td><td>LNU</td><td>1003</td><td>bill@bigdecimals.com</td></tr>
<tr><td>Smith</td><td>LNU</td><td>1004</td><td>smith@bigdecimals.com</td></tr>
<tr><td>Debasis</td><td>Das</td><td>1005</td><td>debasis_das@bigdecimals.com</td></tr>
</table></td></tr>
</table>

Org Hier data in Python

Now lets search an employee name in a department

```
dept1.searchEmployee("John")
'Record for John found in Human Resources Department'
dept1.searchEmployee("John 1")
'Record for John 1 was not found in Human Resources Department'
```

As a next step we can save this data (normalized form) in a database and write a python program that joins the department and employee table to provide a denormalized view of the data and additionally can build a UI to search employees or view department to employee relationships.