Big Corp API

Overview

Your task is to build a web app with a read-only JSON api for 3 resources, `employees` , `departments` , `offices` , you can use any framework or language for building your app.

Data Sources

The data source for `employees` is the api:

`https://rfy56yfcwk.execute-api.us-west-1.amazonaws.com/bigcorp/employees`

it supports two ways of querying, with limit in offset

`https://rfy56yfcwk.execute-api.us-west-1.amazonaws.com/bigcorp/employees?limit=10&offset=20` (gets 21th through 31st record) or querying multiple ids

`https://rfy56yfcwk.execute-api.us-west-1.amazonaws.com/bigcorp/employees?id=3&id=4&id=5` (returns records with id=3, id=4, id=5)

The data source for `offices` and `departments` are the files offices.json, departments.json. These are the entire list of offices and departments, and you can just read all the data for each and keep it in memory in your app.

API

Your app should support GET requests to two endpoints (list & detail) for each resource following a standard REST convention

detail e.g. `/employees/1`

list e.g. `/employees`

list should support query `limit` and `offset` to support pagination.

`limit` is the max number of records returned, and `offset` is the index at which to start.

for instance `/employees?limit=10&offset=17` returns the 18th through 27th employee By default, `limit` is 100 and the max `limit` is 1000

in addition, both methods should support a query parameter called `expand` that lets you expand data along paths of to-one relationships

There are four relationships that can be expanded,

- `manager` in `employees` (expands to `employees`)
- `office` in `employees` (expands to `offices`)
- `department` in `employees` (expands to `departments`)
- `superdepartment` in `departments` (expands to `departments`)

see the examples below for how this works.

Examples

```
/employees?limit=3&expand=department
```


```
[{
        "first": "Patricia",
        "last": "Diaz",
        "id": 1, "manager": null,
        "department": {
                "id": 5,
                "name": "Inbound Sales",
                "superdepartment": 1
        },
        "office": 2
}, {
        "first": "Daniel",
        "last": "Smith",
        "id": 2,
        "manager": 2,
        "department": {
                "id": 5,
                "name": "Inbound Sales",
                "superdepartment": 1
        },
        "office": 2
}, {
        "first": "Thomas",
        "last": "Parker",
        "id": 3,
        "manager": 1,
        "department": {
                "id": 4,
                "name": "Design",
                "superdepartment": 3
        },
        "office": null
}]
```


```
/employees?limit=3&expand=department.superdepartment&expand=manager.office
```


```
[{
        "first": "Patricia",
        "last": "Diaz",
        "id": 1,
        "manager": null,
        "department": {
                "id": 5,
```

```
                "name": "Inbound Sales",
                "superdepartment": {
                        "id": 1,
                        "name": "Sales",
                        "superdepartment": null
                }
        },
        "office": 2
}, {
        "first": "Daniel",
        "last": "Smith",
        "id": 2,
        "manager": {
                "first": "Patricia",
                "last": "Diaz",
                "id": 1,
                "manager": null,
                "department": 5,
                "office": {
                        "id": 2,
                        "city": "New York",
                        "country": "United States",
                        "address": "20 W 34th St"
                }
        },
        "department": {
                "id": 5,
                "name": "Inbound Sales",
                "superdepartment": {
                        "id": 1,
                        "name": "Sales",
                        "superdepartment": null
                }
        },
        "office": 2
}, {
        "first": "Thomas",
        "last": "Parker",
        "id": 3,
        "manager": null,
        "department": {
                "id": 4,
                "name": "Design",
                "superdepartment": {
                        "id": 3,
                        "name": "Product",
                        "superdepartment": null
                }
        },
        "office": null
}]
```

```
/departments/9?expand=superdepartment.superdepartment
```

```
{
        "id": 9,
        "name": "Sales Development",
        "superdepartment": {
                "id": 6,
                "name": "Outbound Sales",
                "superdepartment": {
                        "id": 1,
                        "name": "Sales",
                        "superdepartment": null
                }
        }
}
```

Notes

There is no need to build any kind of POST or other write api. Also, there's no need to use a database or caching system of any kind for this problem.

Evaluation Criteria

- Correctness
- Avoid overuse of employees api endpoint. For instance, requests like `/employees?limit=1000&offset=10000&expand=manager.manager.manager` should not make thousands or even hundreds of calls to the endpoint.
- Well designed APIs
- Well structured and reusable abstractions