

## DAY 4

- `return json.dumps(reviews[res_name][:limit])`
- `[:limit]` → exactly like a substring in java. This goes from index 0 to the limit(exclusive)
- PUT
  - `"/restaurants/<res_name>/items/<item_name>"`
  - THIS URL HAS A MEANING
  - You are going to `restaurants` then you are going to `<res_name>(Wendys)`, then you are going to that `restaurant's items(/items)` and finding that `<item_name>(burger)` to update
  - Simple Put Call  
`http://127.0.0.1:5000/restaurants/<res_name>/`

```
{  
    "address": "124 awesome drive"  
}
```
  - Complex Put Call  
`http://127.0.0.1:5000/restaurants/<res_name>/items/<item_name>`

```
{  
    "item_description": body["item_description"],  
    "item_price": body["item_price"]  
}
```
- DELETE
  - Identify what to delete
  - Specify the LOCATION of item
    - URL params
  - DELETE using
    - `.remove`
    - `.del`
  - Simple Delete call
    - `http://127.0.0.1:5000/restaurants/sushi`
  - Complex Delete Call
    - `http://127.0.0.1:5000/restaurants/sushi/items/cake`

## MongoDB

- A **collection** is a list of **documents**
- A **document** is a JSON object

### Terminal One

- `mongod --dbpath ~/Documents/data` (for mac)
- `sudo mongod` (for ubuntu)
- `Control + C` -- to exit

### Terminal Two

- `mongo`

- use testdb
- `db.numbers.insert({"num": 5})` → we are inserting a document into a collection
- **Numbers** is the **collection** that will hold multiple documents in it
- `db.numbers.find()` → listing all the documents in the collection(numbers)
- Every single object has a randomized unique object id in mongo db
  - `{ "_id" : ObjectId("5dafb940a20906a8ebba5599"), "num" : 5 }`
  - `{ "_id" : ObjectId("5dafb9f7a20906a8ebba559a"), "num" : 9 }`
- **Control + C** -- to exit

**Big Idea: Mongo DB is a way to store data permanently**

Making New terminals

- **Cmd + N** → makes whole new terminal
- **Cmd + T** → makes new terminal tab

Extra

- %20 is how spaces are encoded in URLs