Excelon

GET Test

```
http://localhost:3000/api/test
```

testing the woring of api at initial phase it should return {"message":"working"}

POST Create City

```
http://localhost:3000/api/cities
```

API Request Description

This endpoint allows the client to create a new city entry.

Request Body

- **name** (string) : The name of the city.
- **population** (number) : The population of the city.
- **country** (string) : The country in which the city is located.
- **latitude** (number) : The latitude coordinate of the city

Response

The response will be in JSON format with the following schema:

json

```json
{
    "type": "object",
    "properties": {
        "message": {
            "type": "string"
        },
        "city": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string"
```

Body raw (json)

json

```json
 {
   "name": "Rio de Janeiro",
   "population": 6748000,
   "country": "Brazil",
   "latitude": -22.9068,
   "longitude": -43.1729
  }
```

PUT Update City by Id

```
http://localhost:3000/api/cities/674ec10cd1891f13dc66aceb
```

Update City Details

This endpoint allows the user to update the details of a specific city identified by the provided ID.

Request Body

- `name` (string): The updated name of the city.
- `population` (number): The updated population of the city.
- `country` (string): The updated country of the city.
- `latitude` (number): The updated latitude coordinates of the city.
- `longitude` (number): The updated longitude coordinates of the city.

Response

Upon successful update, the endpoint returns a status code of 200 along with the updated details of the city in JSON format. The response includes the updated `name`, `population`, `country`, `latitude`, `longitude`, `createdAt`, `updatedAt`, and other metadata fields.

Body raw (json)

json

```json
{
  "name": "Updated City Name",
  "population": 1000000,
  "country": "Updated Country",
  "latitude": 12.3456,
  "longitude": 78.9101
}
```

DELETE Delete City by Id

http://localhost:3000/api/cities/674ec10cd1891f13dc66aceb

Delete City

This endpoint is used to delete a specific city.

Request

- Method: DELETE
- URL: `http://localhost:3000/api/cities/674ec10cd1891f13dc66aceb`

Response

The response is a JSON object with the following schema:

json

```json
{
    "type": "object",
    "properties": {
        "message": {
            "type": "string"
        },
        "deletedCity": {
            "type": "object",
            "properties": {
                "id": {
                    "type": "string"
```

The response contains a `message` field and a `deletedCity` object with details of the deleted city including `id`, `name`, `population`, `country`, `latitude`, and `longitude`.

GET Get City by filter

http://localhost:3000/api/cities?page=1&limit=5&sort=population:desc&search=London&fields=name,population

Get Cities

This endpoint retrieves a list of cities based on the provided parameters.

Request

- Method: GET
- URL: `http://localhost:3000/api/cities`
- Query Parameters:
    - page (integer, optional): The page number for pagination.
    - limit (integer, optional): The limit of cities to be returned per page.
    - sort (string, optional): Sort the cities based on a specific field and order (e.g. population:desc).
    - search (string, optional): Search for cities based on a specific keyword.
    - fields (string, optional): Specify the fields to be included in the response.

Response

- Status: 200
- Content-Type: application/json
- { "total": 0, "page": 0, "limit": 0, "cities": [{ "_id": "", "name": "", "population": 0, "country": "", "latitude": 0, "longitude": 0, "createdAt": "", "updatedAt": "", "__v": 0 }]}

PARAMS

page	1
limit	5
sort	population:desc
search	London
fields	name,population