CompanyAPI Documentation

Release 0.0.1

Rich Yap

June 14, 2018

CompanyAPI App main

app.**create_company** ( )

Create company

This function allows user to create(post) a company.

Returns company information of the company added by user in json and http status code

- Example:

```
curl -X POST http://127.0.0.1:5000/ -H 'cache-control: no-cache' -H
'content-type: application/json' -H 'postman-token:
c0db7776-aa16-7702-7467-7a3bca90a5a7'             -d '{
"name": "Company_tempname",
"employees_num": 100,
"email": "info@company.com",
"location": "Philippines",
"industry": "Franchising"
}'
```

- Expected Success Response:

```
HTTP Status Code: 201

{
  "email": "info@company.com",
  "employees_num": 100,
  "industry": "Franchising",
  "location": "Philippines",
  "name": "Company_tempname"
}
```

- Expected Fail Response:

```
HTTP Status Code: 400
{'error': 'Duplicate company name'}
```

app.**delete_company** (*company_id*)

Delete company

This function allows user to delete a company.

Parameters **company_id** (*int*) – id of the company

Returns delete status in json and http status code

- Example:

```
curl -X DELETE http://127.0.0.1:5000/3 -H 'cache-control: no-cache' -H
'content-type: application/json'
```

- Expected Success Response:

```
HTTP Status Code: 200

{
    "Delete": true
}
```

- Expected Fail Response:

```
HTTP Status Code: 404

{'error': 'Not found'}
```

`app.`**`get_company`** (*company_id*)

Get information of a specific company

This function allows user to get a specific company's information through their company_id.

Parameters **`company_id`** (*`int`*) – id of the company

Returns company information of the company accessed by user in json and http status code

- Example:

```
curl -X GET http://127.0.0.1:5000/1 -H 'cache-control: no-cache' -H
'content-type: application/json'
```

- Expected Success Response:

```
HTTP Status Code: 200

{
"Company": {
        "email": "feedback@jollibee.com",
        "employees_num": 9999,
        "id": 1,
        "industry": "Food",
        "location": "Philippines",
        "name": "Jollibee Foods Corporation"
    }
}
```

- Expected Fail Response:

```
HTTP Status Code: 404

{'error': 'Not found'}
```

`app.`**`index`** ( )

Get List of Companies

This function allows user to get a list of the companies with their respective information.

Returns company information of all companies in json and http status code

- Example:

```
curl -X GET http://127.0.0.1:5000/ -H 'cache-control: no-cache' -H
'content-type: application/json'
```

- Expected Success Response:

```
HTTP Status Code: 200

{
"Companies": [
        {
            "email": "feedback@jollibee.com",
            "employees_num": 9999,
            "id": 1,
            "industry": "Food",
            "location": "Philippines",
            "name": "Jollibee Foods Corporation"
        },
        {
            "email": "storm@storm.com",
            "employees_num": 100,
            "id": 3,
            "industry": "Technology",
            "location": "Philippines",
            "name": "Storm"
        },
        {
            "email": "info@company.com",
            "employees_num": 100,
            "id": 6,
            "industry": "Franchising",
            "location": "Philippines",
            "name": "Company_tempname"
        }
    ]
}
```

`app.`**`not_found`**`()`

Error handler
This function returns a not found error in json when called.

Returns not found error in json

`app.`**`search`**`()`

Search company
This function allows user to search for companies through substring search of company names.

Returns searched companies in json and http status code

- Example:

```
curl -X POST http://127.0.0.1:5000/search -H 'cache-control: no-cache' -H
'content-type: application/json' -H 'postman-token:
f2c027be-acda-3a8e-51b6-3b64036df882'
-d '{
    "value": "Jollibee"
}'
```

- Expected Success Response:

```
HTTP Status Code: 200

{
    "Companies": [
        {
            "email": "feedback@jollibee.com",
            "employees_num": 9999,
            "id": 1,
            "industry": "Food",
            "location": "Philippines",
            "name": "Jollibee Foods Corporation"
        }
    ]
}
```

app.**update_company** (*company_id*)

Update information of a specific company

This function allows user to get a specific company's information through their company_id.

Parameters **company_id** (*int*) – id of the company

Returns company information of the company updated by user in json and http status code

- Example:

```
curl -X PUT http://127.0.0.1:5000/2 -H 'cache-control: no-cache' -H
'content-type: application/json' -H 'postman-token:
423a3668-cdaf-5581-5465-0ad4ed1a50c2'
-d '{
    "email": "google_drive@gmail.com",
    "employees_num": 9999,
    "industry": "Software",
    "location": "USA",
    "name": "Google Drive"
}'
```

- Expected Success Response:

```
HTTP Status Code: 200

{
    "email": "google_drive@gmail.com",
    "employees_num": 9999,
    "industry": "Software",
    "location": "USA",
    "name": "Google Drive"
}
```

- Expected Fail Response:

```
HTTP Status Code: 404

{'error': 'Not found'}

or
```

```
HTTP Status Code: 404

{'error': 'Duplicate company name'}
```

CHAPTER 2

CompanyAPI models

CHAPTER 2

CompanyAPI models

CHAPTER 4

Unit testing using unittest

class `unittest_companyAPI.`**`CompanyAppTest`** (*methodName='runTest'*)

`setUp` ( )

Hook method for setting up the test fixture before exercising it.

`test_delete_company` ( )

Test Deleting a Company

This function tests the companyAPI's function to delete a company in the database.

Parameters **`self`** (*`CompanyAppTest.`*) – an instance of the companyAPI app

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED

delete = {"Delete": True} != <json returned by the test_client>
```

`test_delete_nonexisting_company` ( )

Test Deleting a non-existing Company

This function tests the companyAPI's function to return an error message when trying to delete a non-existing company.

Parameters **`self`** (*`CompanyAppTest.`*) – an instance of the companyAPI app

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED
{"error": "Not found"} != <json returned by the test_client>
```

test_get_company ( )

Test Getting Company Information of a Company

This function tests the companyAPI's function to return the right information of the company with id = 1

Parameters self (*CompanyAppTest.*) – an instance of the companyAPI app.

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED

{'Company':
    {'email': 'feedback@jollibee.com',
     'employees_num': 9999,
     'id': 1,
     'industry': 'Food',
     'location': 'Philippines',
     'name': 'Jollibee Foods Corporation'}
}

!= <json returned by the test_client>
```

test_get_empty_company ( )

Test Getting Company Information of a non-existing Company

This function tests the companyAPI's function to return the error message when trying to access a non-existing company where id = 999.

Parameters self (*CompanyAppTest.*) – an instance of the companyAPI app.

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED

{"error": "Not found"} != <json returned by the test_client>
```

test_post_company ( )

Test Adding (Post) a Company with valid company information

This function tests the companyAPI's function to receive company's data and insert this information to the database.

Parameters self (*CompanyAppTest.*) – an instance of the companyAPI app.

Returns none.

- Expected Success Response:

  OK

- Expected Fail Response:

```
FAILED
{
    "name": "Company_tempname",
    "employees_num": 100,
    "email": "info@company.com",
    "location": "Philippines",
    "industry": "Franchising"
}
!= <json returned by the test_client>
```

test_post_duplicate_company_name ( )

Test Adding (Post) a Company with a Similar Company Name in the Database
This function tests the companyAPI's function to return an error when receiving a company data with a name similar to the companies listed in the database.

Parameters **self** (*CompanyAppTest.*) – an instance of the companyAPI app

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED

{"error": "Duplicate company name"} != <json returned by the test_client>
```

test_put_company ( )

Test Editing a Company Information
This function tests the companyAPI's function to edit a company's information.

Parameters **self** (*CompanyAppTest.*) – an instance of the companyAPI app

Returns none.

- **Expected Success Response::**
  OK

- Expected Fail Response:

```
FAILED

{
    "email": "feedback@jollibee.com",
    "employees_num": 9999,
    "industry": "Food",
    "location": "Philippines",
    "name": "Jollibee Foods Corporation"
} != <json returned by the test_client>
```

test_put_duplicate_company_name ( )

Test Editing a Company Information with a Similar Company Name in the Database

This function tests the companyAPI's function to return an error when editing a company name to a new company name similar to the companies listed in the database.

Parameters **`self`** (*`CompanyAppTest.`*) – an instance of the companyAPI app

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED

{"error": "Duplicate company name"} != <json returned by the test_client>
```

`test_put_non_existing_company`( )

Test Editing a Company Information with invalid company id

This function tests the companyAPI's function to return an error when editing a non-existing company's information where the id is = 888.

Parameters **`self`** (*`CompanyAppTest.`*) – an instance of the companyAPI app

Returns none.

-

  Expected Success Response::
  OK

- Expected Fail Response:

```
FAILED
{"error": "Not found"} != <json returned by the test_client>
```

`test_put_null_company_name`( )

Test Editing a Company Information with invalid Company name

This function tests the companyAPI's function to return an error when editing a company name to a new company name which is not a string.

Parameters **`self`** (*`CompanyAppTest.`*) – an instance of the companyAPI app

Returns none.

- Expected Success Response:

```
OK
```

- Expected Fail Response:

```
FAILED

{"error": "Company name not a string"} != <json returned by the
test_client>
```

CHAPTER 5

Indices and tables

- *Index*
- *Module Index*
- *Search Page*

Python Module Index