ultima prova avengers angular

RICORDA DI IMPORTARE NELL APP.MODULE.TS

import { HttpClientModule } from '@angular/common/http'

imports: [

BrowserModule,

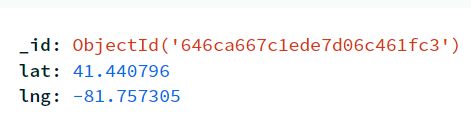
AppRoutingModule,

HttpClientModule

**COMMIT A**

**Commit A - Json**

Collegati con python a mongodb con la seguente stringa:

mongodb+srv://root:pass@cluster0.olw9u.mongodb.net/Avengers

La collection si chiama **locations (formato dei dati ->)**

pip install pymongo

pip install flask

fare un app.py

Collegati con python a mongodb con la seguente stringa:

mongodb+srv://root:pass@cluster0.olw9u.mongodb.net/Avengers

from pymongo import MongoClient

# Connessione a MongoDB

client = MongoClient("mongodb+srv://root:pass@cluster0.olw9u.mongodb.net/Avengers")

db = client.Avengers

fare un file.json e inserire le coordinate

{

"capitan-america": {

"latitude": 40.7128,

"longitude": -74.0060

},

"thor": {

"latitude": 51.5074,

"longitude": -0.1278

},

"spiderman": {

"latitude": 34.0522,

"longitude": -118.2437

},

"iron-man": {

"latitude": 37.7749,

"longitude": -122.4194

}

}

implementa il py con il **coomit b**

**Commit B - Trova avenger(lato server)**

Sul server Python crea 4 route: capitan-america, thor, spiderman, iron-man

Ciascuna delle seguenti route sceglie una coordinata a caso dal file json, la estrae e la ritorna in un formato json appropriato

[Come leggere una riga a caso da un json : <https://gist.github.com/malizia-g/024404b5f37cef339e1c8efcccdf392e>]

from pymongo import MongoClient

from flask import Flask, jsonify

import json

import random

app = Flask(\_\_name\_\_)

# Connessione a MongoDB

client = MongoClient("mongodb+srv://root:pass@cluster0.olw9u.mongodb.net/Avengers")

db = client.Avengers

# Carica i dati da un file JSON

with open("avengers.json") as f:

avengers\_data = json.load(f)

@app.route('/capitan-america')

def capitan\_america():

avenger = random.choice(avengers\_data["capitan-america"])

return jsonify(avenger)

@app.route('/thor')

def thor():

avenger = random.choice(avengers\_data["thor"])

return jsonify(avenger)

@app.route('/spiderman')

def spiderman():

avenger = random.choice(avengers\_data["spiderman"])

return jsonify(avenger)

@app.route('/iron-man')

def iron\_man():

avenger = random.choice(avengers\_data["iron-man"])

return jsonify(avenger)

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

**COMMIT C**

**Commit C - Trova avenger(lato client)**

Crea un client Angular in cui importare google-maps.

Sul client Angular crea 4 bottoni (capitan-america, thor, spiderman, iron-man) . Premendo un bottone viene chiamata la route corrispondente che ritorna la posizione dell’avenger.

Alla pressione di un bottone fai comparire l’[avenger corrispondente](https://drive.google.com/drive/folders/1R5VBIaV1hZKFmmb8RZnOB-N5pk69qvX6) nella posizione ricevuta dal server.

Implementata la parte client dell'applicazione Angular per visualizzare le posizioni degli Avengers.

Aggiunti i pulsanti (Capitan America, Thor, Spiderman, Iron Man) nell'interfaccia utente.

Ogni pulsante chiama la route corrispondente sul server per ottenere la posizione dell'Avenger selezionato.

Le coordinate ricevute dal server vengono visualizzate nella pagina dell'applicazione.

File modificati:

- app.component.html (Aggiunta struttura HTML per i pulsanti e le informazioni degli Avengers)

- app.component.ts (Implementata la logica per chiamare le route del server e visualizzare le coordinate)

app.compoent.ts

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

selectedAvenger: any;

constructor(private http: HttpClient) {}

getAvenger(avengerName: string) {

this.http.get<any>(`http://localhost:5000/${avengerName}`).subscribe(

data => {

this.selectedAvenger = data;

},

error => {

console.error('Errore durante il recupero dell\'Avenger:', error);

}

);

}

}

app.component.html

<button (click)="getAvenger('capitan-america')">Capitan America</button>

<button (click)="getAvenger('thor')">Thor</button>

<button (click)="getAvenger('spiderman')">Spiderman</button>

<button (click)="getAvenger('iron-man')">Iron Man</button>

<div \*ngIf="selectedAvenger">

<h2>{{ selectedAvenger.name }}</h2>

<p>Latitudine: {{ selectedAvenger.latitude }}</p>

<p>Longitudine: {{ selectedAvenger.longitude }}</p>

</div>

**COMMIT D**

**Commit D - Accessi segreti(lato server)**

Crea un server python con una route ‘all’ che ritorni tutte le coordinate in mongodb

**from pymongo import MongoClient**

**from flask import Flask, jsonify**

**import json**

**import random**

**app = Flask(\_\_name\_\_)**

**# Connessione a MongoDB**

**client = MongoClient("mongodb+srv://root:pass@cluster0.olw9u.mongodb.net/Avengers")**

**db = client.Avengers**

**# Carica i dati da un file JSON**

**with open("avengers.json") as f:**

**avengers\_data = json.load(f)**

**@app.route('/capitan-america')**

**def capitan\_america():**

**avenger = random.choice(avengers\_data["capitan-america"])**

**return jsonify(avenger)**

**@app.route('/thor')**

**def thor():**

**avenger = random.choice(avengers\_data["thor"])**

**return jsonify(avenger)**

**@app.route('/spiderman')**

**def spiderman():**

**avenger = random.choice(avengers\_data["spiderman"])**

**return jsonify(avenger)**

**@app.route('/iron-man')**

**def iron\_man():**

**avenger = random.choice(avengers\_data["iron-man"])**

**return jsonify(avenger)**

**AGGIUNGO QUESTO**

**@app.route('/all')**

**def get\_all\_coordinates():**

**all\_coordinates = db.locations.find({}, {"\_id": 0})**

**return jsonify(list(all\_coordinates))**

**# ... (Resto del codice)**

**if \_\_name\_\_ == '\_\_main\_\_':**

**app.run()**

**commit e**

**Commit E - Accessi segreti(lato client)**

Il Client deve avere un bottone che visualizzi gli accessi segreti al quartier generale degli avengers identificandoli con l’icona [stargate](https://drive.google.com/file/d/1zUaeruN372uD36t6DnP2Hrc6M5-GezUa/view?usp=sharing)

**import { Component } from '@angular/core';**

**import { HttpClient } from '@angular/common/http';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']**

**})**

**export class AppComponent {**

**selectedAvenger: any;**

**showAccessIcon = false; // Aggiunta della variabile per mostrare/nascondere l'icona**

**constructor(private http: HttpClient) {}**

**getAvenger(avengerName: string) {**

**this.http.get<any>(`http://localhost:5000/${avengerName}`).subscribe(**

**data => {**

**this.selectedAvenger = data;**

**},**

**error => {**

**console.error('Errore durante il recupero dell\'Avenger:', error);**

**}**

**);**

**}**

**showSecretAccess() {**

**this.showAccessIcon = true;**

**}**

**}**