

# Pokemony 4

## Tematyka

Pokemony to stworzenia, z którymi trenerzy pokemonów podróżują, pracują, bawią się i za pomocą których rozgrywają walki z innymi trenerami. Niektórzy trenerzy obierają ścieżkę naukową i prowadzą badania nad tymi stworzeniami. Profesor Samuel Oak jest jednym z takich naukowców. Obecnie prowadzi badania nad miejscami występowania bardzo rzadkiego pokemona, Mew.



Twoim zadaniem jest dokończenie prototypu aplikacji WWW do rejestracji i wizualizacji współrzędnych geograficznych w których widziano Mew oraz prototypu aplikacji mobilnej do szukania koordynatów pobliskich wystąpień tego pokemona. Niedokończoną wersję aplikacji WWW oraz funkcjonalny prototyp klienta aplikacji mobilnej znajdziesz pod adresem <http://www.mimuw.edu.pl/~ciebie/wwwwegz2015Lpopr-start.zip>.

## Dane

Otrzymujesz niedokończoną aplikację WWW napisaną w Pythonie 3 i Django 1.9. Zawiera ona wypełnioną bazę danych z następującym modelem:

### Model Sighting

x	y
42	42
1	12
0	0
899	643
...	...

Poprawne są wyłącznie dane w prostokącie o wierzchołkach (0, 0) i (899, 643). Dane powinny być unikalne.

Początkowo aplikacja WWW posiada stronę główną wyświetlającą mapę z zaznaczonymi dwoma punktami oraz rozpoczętą implementacją serwera aplikacji RESTowej do znajdowania pobliskich wystąpień Mew. Dołączony jest również funkcjonalny prototyp klienta tej aplikacji RESTowej w formie statycznej aplikacji WWW uruchamianej bezpośrednio z dysku przez przeglądarkę (np. `chrome pokemon4-mobile/client.html`).

### Zadanie na 3



Przerób stronę główną tak, by na mapie były zaznaczone wystąpienia Mew z modelu Sighting. Z punktu widzenia użytkownika mapą ma być prostokątny obrazek `kanto.png`, a zaznaczonymi punktami mają być koliste obrazki `mew.png` z kołem wycelowanym (z dokładnością do 3px) na zaznaczonym punkcie. Zaznaczone punkty mogą wystawać poza mapę i nakładać się na siebie w arbitralnej kolejności. Obrazki już są dołączone do początkowej aplikacji.

Do bazy danych dodaj nowego administratora o nazwie "oak", e-mailu "oak@oaklab.com" i hasło "gary1234".

Dodaj do klasy modelu Sighting sprawdzanie poprawności danych, tj. sprawdzanie czy współrzędne spełniają  $0 \leq x \leq 899$  i  $0 \leq y \leq 643$  oraz że są unikalne (jako para liczb). Sprawdzanie powinno być używane wszędzie, gdzie dodawany lub modyfikowany jest rekord tego modelu, włączając panel administracyjny Django.

Zadbaj, by każda strona walidowała jako poprawny HTML 5 w <https://validator.w3.org/>. Nie wyłączaj zabezpieczenia przed CSRF.

### Dodatki na wyższą ocenę

(+0.5) Dodaj do strony formularz AJAXowy (którego wysłanie nie przeładuje strony) do dodawania nowego punktu wystąpienia Mew. Formularz powinien używać metody HTTP GET. W przypadku pomyślnego zapisania punktu do bazy danych, punkt ten powinien zostać

zaznaczony na mapie. W przypadku jakiegokolwiek błędu powinien zostać wypisany komunikat "Error". Pamiętaj o użyciu napisanego wcześniej sprawdzania poprawności modelu.

(+0.5) Zmodyfikuj stronę tak, by kliknięcie na obrazek reprezentujący punkt wystąpienia Mew powodowało wyświetlenie się okienka modalnego z wypisanymi koordynatami tego punktu, guzikiem "Usuń" oraz guzikiem/linkiem do zamknięcia tego okienka. Kliknięcie guzika "Usuń" powinno spowodować wysłanie zapytania AJAXowego metodą HTTP GET usuwającego punkt z bazy danych. Po poprawnym usunięciu punktu z bazy danych okienko powinno się zamknąć, a punkt powinien zostać usunięty z mapy. W przypadku jakiegokolwiek błędu powinien zostać wypisany komunikat "Error". Za kliknięty punkt możesz uznać ten, którego obrazka opisany kwadrat znajduje się pod kursorem (tj. zwykle `onclick` na `<img>` każdego punktu jest dostatecznie precyzyjne).

(+0.5) Zaimplementuj część po stronie serwera aplikacji mobilnej do znajdowania pobliskich pokemonów. Część po stronie serwera powinna odpowiadać na zapytania HTTP GET skierowane na adres `/nearby/` z parametrami `x` i `y` będącymi poprawnymi współrzędnymi wystąpienia Mew oraz parametrem `dist` będącym liczbą naturalną (np. `/nearby/?x=300&y=200&dist=10`). Powinna odpowiedzieć na zapytanie tablicą dwuelementowych tablic z współrzędnymi punktów z modelu Sighting znajdujących się w prostokącie (włącznie z brzegami) o wierzchołkach  $(x-dist, y-dist)$  i  $(x+dist, y+dist)$ . Zadbaj, aby nie pobierać zbędnych punktów z bazy danych (tj. nie implementuj sam filtru po stronie Pythona, użyj odpowiedniego zapytania). Odpowiedź powinna być zakodowana w formacie JSON. Wyślij poprawną odpowiedź nawet, gdy nie znaleziono żadnych punktów. W przypadku dowolnego błędu po stronie serwera, nic nie musi zostać wysłane. Część kodu tej funkcji jest już zaimplementowana w `mew/rest.py` i zakłada, że serwer jest dostępny na <http://localhost:8000/>. Przetestuj, że twoje rozwiązanie działa z klientem aplikacji mobilnej. Załącz klienta do wysyłanego rozwiązania. Możesz przetestować działanie swojej strony ręcznie np. przez wchodzenie na <http://localhost:8000/nearby/?x=300&y=200&dist=10>. Możesz również edytować klienta.

(+0.5) Napisz test Djangoowy (uruchamiany przez `./manage.py test`) sprawdzający, że strona poprawnie się otwiera (HTTP 200 OK) przy pustej bazie danych. Napisz również test sprawdzający, że gdy w bazie danych są dwa punkty, to ich współrzędne (albo przerobione współrzędne) są przekazane do template strony (przyda się `response.context`).

## Reguły

Niedozwolone są jakiekolwiek formy komunikacji.

Można korzystać wyłącznie z następujących stron internetowych:

- <https://www.djangoproject.com/>
- <http://jquery.com/>
- <http://getbootstrap.com/>
- <http://www.w3schools.com/>, <http://www.w3.org/>
- <https://virtualenv.pypa.io/en/latest/>
- <https://docs.python.org/3/>
- <http://bulbapedia.bulbagarden.net/>

- <https://www.google.pl/> (w sensownym zakresie, jeśli ktoś nie wie, co to jest sensowny zakres, to dla niego sensowny zakres ogranicza się do poprzednich stron)

W szczególności nie jest dozwolone korzystanie z StackOverflow albo tutoriali innych niż znajdujące się powyżej. Oficjalne dokumentacje użytych bibliotek są ok.

Można serwować jQuery bądź Bootstrap z zewnętrznych serwerów. Do rozwiązania warto dodać README.txt opisujący co się zrobiło i dlaczego. Oprócz funkcjonalności będzie oceniany dobór metody i jakoś realizacji.

## Oddawanie

```
zip -r egz_www_2016_2.zip katalog_z_egzaminem
```

Jeśli egz\_www\_2016\_2.zip ma więcej niż 1MB, to coś za dużo zzipowałeś. Umieść spakowany plik tak, aby był dostępny pod adresem:

[http://students.mimuw.edu.pl/~twoj\\_login/egz\\_www\\_2016\\_2.zip](http://students.mimuw.edu.pl/~twoj_login/egz_www_2016_2.zip).

```
cp egz_www_2016_2.zip ~/public_html/  
chmod a+r ~/public_html/egz_www_2016_2.zip  
chmod a+x ~ ~/public_html
```

O godzinie 18:15 skrypt pobierze Twoje rozwiązanie z katalogu domowego i wyśle Ci maila. Dopiero po odebraniu maila wyjdź z sali.