

Podręcznik użytkownika

1. Instalacja

1) Przygotowanie wirtualnego środowiska

Uruchamiamy skrypt `install_virtualenv.sh`:

```
$ ./install_virtualenv.sh
```

Należy zwrócić uwagę na to, aby alias "python" używanej powłoki (shell) odnosił się do interpretera

python w wersji 2. Wymagane jest również połączenie z internetem.

2) Konfiguracja bazy danych

Aby utworzyć nową bazę danych, należy wywołać komendę:

```
$ ./manager.py create_db
```

(należy zwrócić szczególną uwagę, aby `manager.py` NIE był uruchamiany za pomocą `python manager.py`)

Następnie należy utworzyć pierwotną wersję bazy danych oraz miejsca na migracje przyszłych wersji bazy danych:

```
./manager db init
```

Potem należy wykonać update bazy danych:

```
$ ./manager db upgrade
```

3) Uruchamianie serwera

Uruchamianie serwera odbywa się za pomocą komendy:

```
$ ./manager.py runserver
```

Serwer jest dostępny pod adresem `http://127.0.0.1:5000`

2. Konfiguracja

Aplikację możemy konfigurować za pośrednictwem pliku `config.py`.

Możemy zmienić ścieżkę bazy danych, włączyć/wyłączyć tryb debugowania, włączyć/wyłączyć ochronę przed atakami typu CRSF, zmienić klucz do generowania podpisów kryptograficznych chroniących przed atakami CRSF, itp. Dokładny opis pliku `config.py` z objaśnieniem do czego służą konkretne stałe znajduje się w dokumentacji aplikacji.

WAŻNE:

Przy pierwszym logowaniu danego użytkownika zostanie utworzone konto w bazie danych.

Będzie ono

identyfikowane adres e-mail. Ze względów bezpieczeństwa nowo dodane konta mają rolę użytkownika. Aby umożliwić użytkownikowi danego konta dodawanie oraz modyfikację postów należy zwiększyć uprawnienia danego konta z poziomu

użytkownika do poziomu administratora:

Uruchamiamy interpreter pythona:

```
$ ./manager.py shell
```

Importujemy model użytkownika:

```
>>> from app.models import User
```

Importujemy bazę danych:

```
>>> from app import db
```

Wyszukujemy użytkownika w bazie danych:

```
>>> u = User.query.filter(User.email=="jan.kowalski@example.com").first()
```

Commitujemy nasze zmiany do bazy danych:

```
>>> db.session.add(u)
```

```
>>> db.session.commit()
```