

Metodika testovania backend funkcionality

Slavomír Mazúr

November 2017

1 Testovanie BE funkcionality

Funkcionalita jednotlivých častí sa testuje pomocou jednotkových (angl. Unit) testov, ktoré testujú malé časti aplikácie alebo jednotlivé prípady použitia. V rámci týchto testov sa testujú odpovede (angl. Responses) na jednotlivé požiadavky (angl. Requests). Testujú sa jednak také prípady, ktoré vedú k správne výsledku (napríklad správne vyplnenie registračných údajov), avšak je potrebné otestovať aj také prípady, o ktorých vieme, že musí nastať chyba a otestovať či táto chyba naozaj nastala (napríklad registrácia pomocou emailu, pod ktorým už niekto zaregistrovaný je). Výhodou takéhoto prístupu je, že pri zmene alebo pridaní novej funkcionality vieme zároveň skontrolovať či sa stará funkcionality nezmenila a či sú odpovede na jednotlivé požiadavky stále správne. Pri testoch sa vytvorí dočasná databáza, ktorá na začiatku neobsahuje žiadne dáta, preto ak je potrebné musia byť dáta do nej vložené.

- Všetky testovacie súbory a ostatné súbory, ktoré s testovaním súvisia sa nachádzajú v priečinku `/tests/`
- Testy pre jednotlivé moduly sa nachádzajú v samostatných súboroch (napr. `test_login.py` alebo `test_registration.py`)
- Každý test sa nachádza v samostatnej metóde, pričom názov metódy musí začínať slovom "test"(Django následne rozozná, že sa jedná o testovaciu metódu)
- Pri testoch sa simuluje sa požiadavka (GET, POST, DELETE, PUT), pričom sa následne otestuje hodnota odpovede, s očakávanou správnu odpoveďou
- Dáta, ktoré sa posielajú ako JSON objekty v rámci jednotlivých požiadaviek sa nachádzajú v súbore `data.json`, aby kód testov pôsobil prehľadnejšie

- Dáta, ktoré je potrebné vložiť do dočasnej databázy pred testovaním daného modulu sa vkladajú v metóde setUp(), ktorá sa volá vždy ako prvá v rámci triedy
- Každá metóda je okomentovaná pomocou documentation string (docstring), pričom komentár pár vetami opisuje funkčnosť, ktorá je v danej metóde testovaná

Testy je možné spustiť príkazom:

```
python manage.py test tests
```

čo znamená, že sa vykonajú všetky testy, ktoré sa nachádzajú v priečinku /tests.

```
def test_ping_response(self):
    """Test ping response"""
    response = self.client.get('/ping/')
    self.assertEqual(response.status_code, 200)
    self.assertEqual(response.content, b'pong')
```

Daný úsek kódu obsahuje ukážkovú metódu, ako je možné vidieť:

- Názov metódy začína slovom "test".
- Hneď za definíciou metódy sa v triple-quotes(") nachádza docstring, ktorý v krátkosti opisuje čo sa v danej metóde testuje.
- Pomocou klienta je odoslaná GET požiadavka na /ping/.
- Očakáva sa, že status code, na danej odpovede bude 200, pričom je pomocou funkcie assertEquals porovnaná skutočná hodnota s tou očakávanou.
- Rovnako je porovnaný aj obsah danej odpovede, ktorý by mal byť v tomto prípade reťazec 'pong'.
- V prípade, že sú všetky porovnania vyhovujúce (vyhodnotené ako true), test je úspešný, v opačnom prípade je vypísaná chybová hláška, v ktorej je uvedené kde došlo ku nerovnosti.