

Metodika testovania



Metodika testovania

Tím č. 03 FireAnt

Vedúci tímu: Branislav Pecher, Ivan Srba

Autor metodiky: [Denis Mitana](#)

Dátum vytvorenia: 09.11.2019

Poslednú zmenu vykonal: [Denis Mitana](#)

Dátum poslednej zmeny: 09.11.2019

Obsah

- [Úvod](#)
- [Všeobecné pokyny k testovaniu](#)
- [Testovanie backendu](#)
 - [Umiestňovanie testovacích modulov](#)
 - [Pomenovanie testovacích modulov](#)
 - [Pomenovanie testovacích funkcií](#)
 - [Druhy testov, ktoré používame](#)
 - [Jednotkové testovanie \(unit testing\)](#)
 - [Integračné testovanie \(integration testing\)](#)
 - [Testovanie funkcií využívajúce externú databázu](#)
- [Testovanie frontendu](#)

Úvod

Testovanie pomáha overiť správnosť nášho riešenia, odhaliť chyby v skorom štádiu ich vzniku a minimalizovať počet chýb na produkcii. Testy sú taktiež súčasťou našej [Metodiky Definition of Done](#), takže je nutné ich písať.

Náš systém sa skladá z *backendu* a *frontendu*, ktoré sú oddelené. Je preto nutné testovať každú časť systému samostatne.

Všeobecné pokyny k testovaniu

- Každá funkcionálna, ktorá sa podieľa na naplnení požiadaviek na náš systém **musí** byť otestovaná.

- Funkcie a metódy je nutné písať tak, aby boli otestovateľné.
- Najlepšie je písať **jednotkové testy** pre menšie funkcie ešte pred tým, ako tieto funkcie vzniknú. Tento prístup sa nazýva *Test Driven Development (TDD)*.
- Testy sa automaticky spúšťajú pri vytvorení, úprave a mergnutí *Pull Requestu*.

Testovanie backendu

Na testovanie backendu využívame knižnicu [pytest](#).

Aby mali všetky testy jednotný formát budeme sa riadiť pravidlami uvedenými v nasledujúcich sekciách.

Umiestňovanie testovacích modulov

- Každý testovací modul je umiestnení v hierarchii balíkov, ktorá je rovnaká ako pri testovanej funkcionalite.
- Najvyššie balíky sú ešte umiestnené v balíku, ktorý špecifikuje druh testov (napr. integračné testy, jednotkové testy, ...).

Nasledujúci príklad ukazuje takúto štruktúru:

```

Štruktúra funkcionalít

api
  helpers
    suggestions.py
    url.py
    widgets.py
  views
    v1
      checker.py

```

```

Štruktúra testov

tests
  conftest.py
  integration_tests
    api
      views
        v1
          test_checker.py
  pytest.ini
  unit_tests
    api
      helpers
        test_suggestions.py
        test_url.py
        test_widgets.py

```

Pomenovanie testovacích modulov

- Každý testovací modul začína prefixom **test_** a pokračuje názvom modulu, ktorý testuje.

Príklad č 1: Ak testujem funkcionalitu z modulu *checker.py* tak názov testovacieho modulu bude *test_checker.py*.

Viac príkladov je možné vidieť v príklade vyššie.

Pomenovanie testovacích funkcií

- Každá testovacia funkcia v testovacom module začína prefixom **test_** a pokračuje názvom funkcie, ktorú testuje.
- Ak test nejakej funkcie je rozdelený do viacerých funkcií, vtedy musí za názvom funkcie nasledovať krátky a jasný popis, čo presne sa z funkcionality danej funkcie testuje (toto sa týka hlavne integračných testov, ale môže to nastať aj pri jednotkových testoch).

Príklad č. 1: Chcem otestovať funkciu `min_claim_rating(...)` a všetky možnosti má zmysel dať do jednej testovacej funkcie, tak testovaciu funkciu nazvem `test_min_claim_rating()`.

Príklad č. 2: Chcem otestovať endpoint handler `analyze(...)` keď je na vstupe článok alebo zdroj. Ak by som dal tieto testy do jednej funkcie, tak by bola príliš dlhá. Preto vytvorím dve testovacie funkcie a nazvem ich `test_analyze_article()` a analogicky `test_analyze_source()`.

Druhy testov, ktoré používame

Momentálne používame dva druhy testovania, jedná sa o jednotkové a integračné testovanie.

Jednotkové testovanie (unit testing)

Je určené pre malé funkcie, ktoré vykonávajú spravidla jednu funkciu.

Integračné testovanie (integration testing)

Je určené pre funkcie, ktoré spájajú viacero funkcionalít (predpokladá sa, že každá samotná funkcionalita je jednotkovo otestovaná).

Testovanie funkcií využívajúce externú databázu

Táto časť metodiky bude doplnená v najbližšom šprinte, hneď ako analyzujeme dostupné metódy a vyberieme pre nás tu najlepšiu.

Testovanie frontendu

Testovanie frontendu zatiaľ nie je nutné, ale čoskoro sa to zmení. **Táto časť metodiky bude doplnená ihneď, ako začneme frontend testovať.**