

BDD & Python

Workshop, Dublin, Oct 5th 2017

Colm Harrington

Before we begin ...

- Questions:
 - Who has written BDD features before?
 - Who has experience in python?
 - Who has experience in pytest?

Workshop setup

- Python installed (2 or 3 will do, but 3 is better)
- Pip installed: <https://pip.pypa.io/en/stable/installing/>
 - pip is already installed if you're using **Python 2 >=2.7.9 or Python 3 >=3.4**
 - For brevity, Install pip, don't upgrade your python
- Pipenv: to manage our dependencies in isolation
 - Or virtualenv if you're already using that
- Help your neighbour!

What is BDD?

- Dan North (Father of BDD)
- Focus on user behaviour
- “Non-technical” people can write the features
- Scenarios are written in a ‘language’ called ‘Gherkin’
- Gherkin is the language, Cucumber is a popular implementation of BDD
- Workshop is not about ‘Cucumber’
 - Ruby centric
- We will be using a pytest plugin called pytest-bdd
 - No extra test runner required IF you already use pytest

BDD

- Suitable for:
 - Higher-level
 - Functional
 - black box tests
- Works by:
 - Not magic! Unfortunately :-(
 - Limiting english to a machine parsable set
 - Gherkin
 - Tooling converts this 'english' to machine executable code

Advantages of BDD

- The missing link between product and engineering
- Inclusion * (THANKS TO AUTOMATION PANDA, LINK INCLUDED AT END OF DECK)
- Clarity
- Streamlining
- Shift-left
- Artifacts
- Automation
- Test-driven
- Code-reuse
- Parameterization
- Variation
- Momentum
- Adaptability

What is pytest?

- Common testing framework in python community
- Consists of a plugin based architecture
- Pytest-bdd is one such plugin
- No need to learn another testing framework

Pytest-bdd

- Pytest plugin
- Workshop API examples are based on open source Star Wars API
 - <http://swapi.co/>
 - Easily transferable to browser based tests as well

Feature Template

As a [role]

I want [feature]

So that [benefit]

As a real life person

I want to learn about Star Wars

So that I can impress random people
at parties

Scenario Template

Given an initial state

When an action is taken

Then verify an outcome

Given a logged in user

When the user logs out

Then the user is taken to the home page

Behaviour != Procedure

- Useful for decoupling the tests from the software
- Implementation changes often, user outcomes change less often
- Product could be completely rewritten in another language and/or framework and the features and scenarios would still be valid

Common Errors

- Coupling to the implementation
- 1:1 translation from old procedural tests
- Mixing first & third person points of view
- Creating steps that cannot be reused
- Externalising test data unnecessarily
- Inconsistency in style

Useful Links

- <http://pytest-bdd.readthedocs.io/en/stable/index.html>
- <https://docs.pytest.org/en/latest/>
- <https://dannorth.net/introducing-bdd/>
- <https://martinfowler.com/bliki/GivenWhenThen.html>
- <https://github.com/cucumber/cucumber/wiki/Gherkin>
- <https://automationpanda.com/2017/01/25/bdd-101-introducing-bdd/>
- <https://automationpanda.com/2017/01/30/bdd-101-writing-good-gherkin/>