

Exploring Testing Options



Jamie Counsell

SOFTWARE DEVELOPER

@jamiecounsell www.jamiecounsell.me



Testing Options



Our focus so far

- Python's default test features
- Django's built in test features

Other options

- Third party testing tools
- Additional testing libraries
- End-to-end testing

pytest

Extends unit tests

- Detailed failures
- Better autodiscovery
- Plugin support

Pytest in Django using pytest-django

- Write more function-based tests
- Re-use databases between tests
- File-based configuration
- Single-word command to run tests



tox

Good for building reusable Django apps

- Distributed for many users
- Regression issues a larger risk

Multi-version testing

- Multiple Python versions
- Multiple Django versions



```
def multiply(a, b):  
    """  
    >>> multiply(4, 3)  
    12  
    >>> multiply('a', 3)  
    'aaa'  
    """  
    return a * b
```

Doctests

- Tests written inside comment strings
- Great replacement for short unit tests
- Assist in documenting code



Ghost Inspector

UI testing made easy

- Record UI tests using Chrome extension
- Run tests remotely from a real browser
- Compare screenshots
- Reasonably priced with a free tier

Limitations

- Unreliable with AJAX-based or single page apps
- Tests can be difficult to change as UI changes



Demo



Ghost Inspector Testing

- Creating tests
- Running tests
- Screenshot comparison



Summary



Other testing options to explore

- pytest
- tox
- doctests
- Ghost Inspector

Course Summary



Testing in Django

- Python's standard test classes
- Django test classes
- Best practices for testing in Django

Measuring test coverage

- Configuring coverage for accuracy
- Keeping coverage up

Testing tools

- Selenium
- Ghost Inspector

