

The Basics of TDD and Django

Buy the Book!

Table of Contents

[Praise for *Test-Driven Development with Python*](#)

[Preface](#)

[Preface to the Third Edition: TDD in the Age of AI](#)

[Prerequisites and Assumptions](#)

[Acknowledgments](#)

[The Basics of TDD and Django](#)

[1. Getting Django Set Up Using a Functional Test](#)

[2. Extending Our Functional Test Using the unittest Module](#)

[3. Testing a Simple Home Page with Unit Tests](#)

[4. What Are We Doing with All These Tests? \(And, Refactoring\)](#)

[5. Saving User Input: Testing the Database](#)

[6. Improving Functional Tests: Ensuring Isolation and Removing Magic Sleeps](#)

[7. Working Incrementally](#)

[8. Prettification: Layout and Styling, and What to Test About It](#)

[Going to Production](#)

[9. Containerization aka Docker](#)

[10. Making Our App Production-Ready](#)

[11. Getting a Server Ready for Deployment](#)

[12. Infrastructure as Code: Automated Deployments with Ansible](#)

[Forms and Validation](#)

[13. Splitting Our Tests into Multiple Files, and a Generic Wait Helper](#)

[14. Validation at the Database Layer](#)

[15. A Simple Form](#)

[16. More Advanced Forms](#)

[More Advanced Topics in Testing](#)

[17. A Gentle Excursion into JavaScript](#)

[18. Deploying Our New Code](#)

[19. User Authentication, Spiking, and De-Spiking](#)

[20. Using Mocks to Test External Dependencies](#)

[21. Using Mocks for Test Isolation](#)

[22. Test Fixtures and a Decorator for Explicit Waits](#)

- [23. Debugging and Testing Server Issues](#)
 - [24. Finishing "My Lists": Outside-In TDD](#)
 - [25. CI: Continuous Integration](#)
 - [26. The Token Social Bit, the Page Pattern, and an Exercise for the Reader](#)
 - [27. Fast Tests, Slow Tests, and Hot Lava](#)
 - [Appendix A: Obey the Testing Goat!](#)
 - [Appendix B: The Subtleties of Functionally Testing External Dependencies](#)
 - [Appendix C: Continuous Deployment \(CD\)](#)
 - [Appendix D: Behaviour-Driven Development \(BDD\) Tools](#)
 - [Appendix E: Test Isolation, and "Listening to Your Tests"](#)
 - [Appendix F: Building a REST API: JSON, Ajax, and Mocking with JavaScript](#)
 - [Appendix G: Cheat Sheet](#)
 - [Appendix H: What to Do Next](#)
 - [Appendix I: Source Code Examples](#)
 - [Appendix J: Bibliography](#)
-

In this first part, I'm going to introduce the basics of *test-driven development* (TDD). We'll build a real web application from scratch, writing tests first at every stage.

We'll cover functional testing with Selenium, as well as unit testing, and see the difference between the two. I'll introduce the TDD workflow, red/green/refactor.

I'll also be using a version control system (Git). We'll discuss how and when to do commits and integrate them with the TDD and web development workflow.

We'll be using Django, the Python world's most popular web framework (probably). I've tried to introduce the Django concepts slowly and one at a time, and provide lots of links to further reading. If you're a total beginner to Django, I thoroughly recommend taking the time to read them. If you find yourself feeling a bit lost, take a couple of hours to go through the [official Django tutorial](#) (<https://docs.djangoproject.com/en/5.2/intro>) and then come back to the book.

In The Basics of TDD and Django, you'll also get to meet the Testing Goat...



Be careful with copy and paste. If you're working from a digital version of the book, it's natural to want to copy and paste code listings from the book as you're working through it. It's much better if you don't: typing things in by hand gets them into your muscle memory, and just feels much more real. You also inevitably make the occasional typo, and debugging is an important thing to learn.

Quite apart from that, you'll find that the quirks of the PDF format mean that weird stuff often happens when you try to copy/paste from it...

License: Creative Commons [CC-BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode) (<https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>). Last updated: 2025-10-27 16:49:22 UTC

Comments

ALSO ON OBEY THE TESTING GOAT!

Getting to a Production-Ready Deployment

9 years ago • 99 comments

Our deployment is working fine but it's not production-ready. Let's try to get it ...

Getting A Server Ready For Deployment

a year ago • 1 comment

With a PaaS, you don't get your own server, instead you're renting a "service" at a ...

Working Incrementally

2 years ago • 3 comments

Enough housekeeping with our URLs. It's time to bite the bullet and change our ...

Prettification and Styling

2 years ago • 2 comments

We're starting to release the our site, but

2 Comments

 Kgots Koete ▼



Join the discussion...

1

Share

Best

Newest

Oldest

J

junghanChoi

8 years ago

Let's Go!

1

0

Reply 



YANiC

8 years ago

So true, it makes me feel a huge difference

0

0

Reply 

[Subscribe](#)

[Privacy](#)

[Do Not Sell My Data](#)