

1. Record Nr.	UNINA9910819840803321
Autore	Tracey Karen M
Titolo	Django 1.1 testing and debugging : building rigorously tested and bug-free Django applications // Karen M. Tracey
Pubbl/distr/stampa	Birmingham, U.K., : Packt Pub., 2010
ISBN	1-282-57942-8 9786612579424 1-84719-757-4
Edizione	[1st ed.]
Descrizione fisica	1 online resource (436 p.)
Collana	From technologies to solutions
Disciplina	006.76
Soggetti	Computer programs - Testing Debugging in computer science Application software - Development Python (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover; Copyright; Credits; About the Author; About the Reviewer; Table of Contents; Preface; Chapter 1: Django Testing Overview; Getting started: Creating a new application; Understanding the sample unit test; Understanding the sample doctest; Running the sample tests; Breaking things on purpose; Test errors versus test failures; Command line options for running tests; Verbosity; Settings; Pythonpath; Traceback; Noinput; Version; Summary; Chapter 2: Does This Code Work? Doctests in Depth; The Survey application models; Testing the Survey model; Testing Survey model creation Is that test useful?Developing a custom Survey save method; Deciding what to test; Some pros and cons of doctests so far; Additional doctest caveats; Beware of environmental dependence; Beware of database dependence; Beware of test interdependence; Beware of Unicode; Summary; Chapter 3: Testing 1, 2, 3: Basic Unit Testing; Unit tests for the Survey save override method; Pros of the unit test version; Cons of the unit test version; Revisiting the doctest caveats; Environmental dependence; Database dependence; Test interdependence; Unicode; Providing data for unit tests

Providing data in test fixtures
Example test that needs test data; Using the admin application to create test data; Writing the function itself; Writing a test that uses the test data; Extracting the test data from the database; Getting the test data loaded during the test run; Creating data during test set up; Summary; Chapter 4: Getting Fancier: Django Unit Test Extensions; Organizing tests; Creating the survey application home page; Defining the survey application URLs; Developing views to serve pages; Creating templates for pages; Testing the survey home page

Creating the survey detail pages
Refining the survey detail view; Templates for the survey detail pages; Basic testing of the survey detail pages; Customizing the admin add and change survey pages; Developing a custom survey form; Configuring admin to use the custom form; Testing the admin customization; Additional test support; Supporting additional HTTP methods; Maintaining persistent state; E-mail services; Providing test-specific URL configuration; Response context and template information; Testing transactional behavior

Chapter 5: Filling in the Blanks: Integrating Django and Other Test Tools
Problems of integration; Specifying an alternative test runner; Creating a new management command; How much of the code are we testing?; Using coverage standalone; Integrating coverage into a Django project; The twill web browsing and testing tool; Using the twill command line program; Using twill in a TestCase; Summary; Chapter 6: Django Debugging Overview; Django debug settings; The DEBUG and TEMPLATE_DEBUG settings; The TEMPLATE_STRING_IF_INVALID setting; Debug error pages; Database query history
Debug support in the development server

Sommario/riassunto

This book teaches by example. It walks in detail through development of a sample application, illustrating each step via complete working code and either screenshots or console snippets. The cumbersome and time consuming task of debugging will be a cake walk with this book. If you are a Django application developer who wants to create robust applications quickly that work well and are easy to maintain in the long term, this book is for you. This book is the right pick if you want to be smartly tutored to make best use of Django's rich testing and debugging support and make testing an effortles
