

1. Record Nr.	UNINA9910467225503321
Autore	Ravindran Arun
Titolo	Django Design Patterns and Best Practices - Second Edition [[electronic resource] /] / Ravindran, Arun
Pubbl/distr/stampa	Packt Publishing, , 2018
Edizione	[2nd edition]
Descrizione fisica	1 online resource
Soggetti	Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Chapter 1: Django and Patterns; Why Django?; The story of Django; A framework is born; Removing the magic; Django keeps getting better; How does Django work?; What is a pattern?; Gang of four patterns; Is Django MVC?; Fowler's patterns; Are there more patterns?; Patterns in this book; Criticism of patterns; How to use patterns; Python Zen and Django's design philosophy; Summary; Chapter 2: Application Design; How to gather requirements?; Are you a storyteller?; HTML mockups.; Designing the application; Dividing a project into apps; Reuse or roll-your-own?; My app sandbox; Which packages made it?; Best practices before starting a project; SuperBook -- your mission, should you choose to accept it; Why Python 3?; Which Django Version to use; Starting the project; Summary; Chapter 3: Models; M is bigger than V and C; The model hunt; Splitting models.py into multiple files; Structural patterns; Patterns -- normalized models; Problem details; Solution details; Three steps of normalization; First normal form (1NF); Second normal form (2NF); Third normal form (3NF).; Django models; Performance and denormalization; Should we always normalize?; Pattern -- model mixins; Problem details; Solution details; Model mixins; Pattern -- user profiles; Problem details; Solution details; Signals; Admin; Multiple profile types; Pattern -- service objects; Problem details; Solution details; Retrieval patterns; Pattern -- property field; Problem details; Solution details; Cached properties; Pattern -- custom model managers; Problem details; Solution details; Set operations on QuerySets; Chaining multiple QuerySets; Migrations;

Summary; Chapter 4: Views and URLs.; A view from the topViews got classier; Class-based generic views; Class-Based Views are not always Class-Based Generic Views; View mixins; Order of mixins; Decorators; View patterns; Pattern -- access controlled views; Problem details; Solution details; Pattern -- context enhancers; Problem details; Solution details; Pattern -- services; Problem details; Solution details; Designing URLs; URL anatomy; What happens in urls.py?; Simplified URL pattern syntax; Regular expression URL pattern syntax; Can the simplified syntax replace regular expressions?; Names and namespaces; Pattern order.; URL pattern stylesDepartment store URLs; RESTful URLs; React.js, Vue.js, and other view replacements; Summary; Chapter 5: Templates; Understanding Django's template language features; Variables; Attributes; Filters; Tags; Philosophy -- don't invent a programming language; Jinja2; Organizing templates; How templates work; Using Bootstrap; But they all look the same!; Lightweight alternatives; Template patterns; Pattern -- template inheritance tree; Problem details; Solution details; Pattern -- the active link; Problem details; Solution details; A template-only solution; Custom tags; Summary.

Sommario/riassunto

Build maintainable websites with elegant Django design patterns and modern best practices About This Book Explore aspects of Django from Models and Views to testing and deployment Understand the nuances of web development such as browser attack and data design Walk through various asynchronous tools such as Celery and Channels Who This Book Is For This book is for you whether you're new to Django or just want to learn its best practices. You do not have to be an expert in Django or Python. No prior knowledge of patterns is expected for reading this book but it would be helpful. What You Will Learn Make use of common design patterns to help you write better code Implement best practices and idioms in this rapidly evolving framework Deal with legacy code and debugging Use asynchronous tools such as Celery, Channels, and asyncio Use patterns while designing API interfaces with the Django REST Framework Reduce the maintenance burden with well-tested, cleaner code Host, deploy, and secure your Django projects In Detail Building secure and maintainable web applications requires comprehensive knowledge. The second edition of this book not only sheds light on Django, but also encapsulates years of experience in the form of design patterns and best practices. Rather than sticking to GoF design patterns, the book looks at higher-level patterns. Using the latest version of Django and Python, you'll learn about Channels and asyncio while building a solid conceptual background. The book compares design choices to help you make everyday decisions faster in a rapidly changing environment. You'll first learn about various architectural patterns, many of which are used to build Django. You'll start with building a fun superhero project by gathering the requirements, creating mockups, and setting up the project. Through project-guided examples, you'll explore the Model, View, templates, workflows, and code reusability techniques. In addition to this, you'll learn practical Python coding techniques in Django that'll enable you to tackle problems related to complex topics such as legacy coding, data modeling, and code reusability. You'll discover API design principles and best practices, and understand the need for asynchronous workflows. During this journey, you'll study popular Python code testing techniques in Django, various web security threats and their countermeasures, and the monitoring and performance of your application. Style and approach This comprehe...
