

Creating model managers	25
<b>Building list and detail views</b>	<b>26</b>
Creating list and detail views	26
Adding URL patterns for your views	28
Canonical URLs for models	29
<b>Creating templates for your views</b>	<b>30</b>
<b>Adding pagination</b>	<b>34</b>
<b>Using class-based views</b>	<b>36</b>
<b>Summary</b>	<b>38</b>
<b>Chapter 2: Enhancing Your Blog with Advanced Features</b>	<b>39</b>
<b>Sharing posts by email</b>	<b>40</b>
Creating forms with Django	40
Handling forms in views	41
Sending emails with Django	43
Rendering forms in templates	45
<b>Creating a comment system</b>	<b>50</b>
Building a model	50
Creating forms from models	52
Handling ModelForms in views	53
Adding comments to the post detail template	54
<b>Adding the tagging functionality</b>	<b>58</b>
<b>Retrieving posts by similarity</b>	<b>64</b>
<b>Summary</b>	<b>66</b>
<b>Chapter 3: Extending Your Blog Application</b>	<b>67</b>
<b>Creating custom template tags and filters</b>	<b>68</b>
Custom template tags	68
Custom template filters	73
<b>Adding a sitemap to your site</b>	<b>76</b>
<b>Creating feeds for your blog posts</b>	<b>80</b>
<b>Adding full-text search to your blog</b>	<b>82</b>
Installing PostgreSQL	83
Simple search lookups	84
Searching against multiple fields	84
Building a search view	85
Stemming and ranking results	88
Weighting queries	89
Searching with trigram similarity	90
Other full-text search engines	91
<b>Summary</b>	<b>91</b>

---

<b>Chapter 4: Building a Social Website</b>	<b>93</b>
<b>Creating a social website project</b>	<b>94</b>
Starting your social website project	94
<b>Using the Django authentication framework</b>	<b>95</b>
Creating a login view	96
Using Django authentication views	101
Login and logout views	102
Changing password views	108
Resetting password views	110
<b>User registration and user profiles</b>	<b>115</b>
User registration	115
Extending the user model	119
Using a custom user model	125
Using the messages framework	125
<b>Building a custom authentication backend</b>	<b>128</b>
<b>Adding social authentication to your site</b>	<b>130</b>
Running the development server through HTTPS	132
Authentication using Facebook	134
Authentication using Twitter	140
Authentication using Google	142
<b>Summary</b>	<b>147</b>
<b>Chapter 5: Sharing Content on Your Website</b>	<b>149</b>
<b>Creating an image bookmarking website</b>	<b>150</b>
Building the image model	150
Creating many-to-many relationships	152
Registering the image model in the administration site	153
<b>Posting content from other websites</b>	<b>153</b>
Cleaning form fields	154
Overriding the save() method of a ModelForm	155
Building a bookmarklet with jQuery	160
<b>Creating a detail view for images</b>	<b>168</b>
<b>Creating image thumbnails using easy-thumbnails</b>	<b>170</b>
<b>Adding AJAX actions with jQuery</b>	<b>172</b>
Loading jQuery	173
Cross-site request forgery in AJAX requests	174
Performing AJAX requests with jQuery	176
<b>Creating custom decorators for your views</b>	<b>179</b>
<b>Adding AJAX pagination to your list views</b>	<b>181</b>
<b>Summary</b>	<b>186</b>

---

<b>Chapter 6: Tracking User Actions</b>	<b>187</b>
<b>Building a follow system</b>	<b>187</b>
Creating many-to-many relationships with an intermediary model	188
Creating list and detail views for user profiles	191
Building an AJAX view to follow users	196
<b>Building a generic activity stream application</b>	<b>198</b>
Using the contenttypes framework	200
Adding generic relations to your models	201
Avoiding duplicate actions in the activity stream	204
Adding user actions to the activity stream	205
Displaying the activity stream	206
Optimizing QuerySets that involve related objects	207
Using select_related()	207
Using prefetch_related()	208
Creating templates for actions	208
<b>Using signals for denormalizing counts</b>	<b>210</b>
Working with signals	211
Application configuration classes	213
<b>Using Redis for storing item views</b>	<b>215</b>
Installing Redis	215
Using Redis with Python	217
Storing item views in Redis	218
Storing a ranking in Redis	220
Next steps with Redis	223
<b>Summary</b>	<b>223</b>
<b>Chapter 7: Building an Online Shop</b>	<b>225</b>
<b>Creating an online shop project</b>	<b>226</b>
Creating product catalog models	227
Registering catalog models on the administration site	229
Building catalog views	230
Creating catalog templates	233
<b>Building a shopping cart</b>	<b>237</b>
Using Django sessions	238
Session settings	239
Session expiration	240
Storing shopping carts in sessions	240
Creating shopping cart views	245
Adding items to the cart	245
Building a template to display the cart	247
Adding products to the cart	249
Updating product quantities in the cart	251

---

Creating a context processor for the current cart	252
Context processors	252
Setting the cart into the request context	253
<b>Registering customer orders</b>	<b>255</b>
Creating order models	255
Including order models in the administration site	257
Creating customer orders	258
<b>Launching asynchronous tasks with Celery</b>	<b>263</b>
Installing Celery	263
Installing RabbitMQ	264
Adding Celery to your project	264
Adding asynchronous tasks to your application	265
Monitoring Celery	267
<b>Summary</b>	<b>268</b>
<b>Chapter 8: Managing Payments and Orders</b>	<b>269</b>
<hr/>	
<b>Integrating a payment gateway</b>	<b>269</b>
Creating a Braintree sandbox account	270
Installing the Braintree Python module	271
Integrating the payment gateway	272
Integrating Braintree using Hosted Fields	274
Testing payments	280
Going live	283
<b>Exporting orders to CSV files</b>	<b>284</b>
Adding custom actions to the administration site	284
<b>Extending the administration site with custom views</b>	<b>287</b>
<b>Generating PDF invoices dynamically</b>	<b>292</b>
Installing WeasyPrint	292
Creating a PDF template	292
Rendering PDF files	294
Sending PDF files by email	297
<b>Summary</b>	<b>300</b>
<b>Chapter 9: Extending Your Shop</b>	<b>301</b>
<hr/>	
<b>Creating a coupon system</b>	<b>301</b>
Building the coupon model	302
Applying a coupon to the shopping cart	304
Applying coupons to orders	312
<b>Adding internationalization and localization</b>	<b>314</b>
Internationalization with Django	315
Internationalization and localization settings	315
Internationalization management commands	316
How to add translations to a Django project	316

How Django determines the current language	316
Preparing your project for internationalization	317
Translating Python code	318
Standard translations	319
Lazy translations	319
Translations including variables	319
Plural forms in translations	319
Translating your own code	320
Translating templates	324
The {% trans %} template tag	324
The {% blocktrans %} template tag	324
Translating the shop templates	325
Using the Rosetta translation interface	328
Fuzzy translations	331
URL patterns for internationalization	332
Adding a language prefix to URL patterns	332
Translating URL patterns	333
Allowing users to switch language	334
Translating models with django-parler	336
Installing django-parler	336
Translating model fields	337
Integrating translations into the administration site	339
Creating migrations for model translations	340
Adapting views for translations	341
Format localization	344
Using django-localflavor to validate form fields	345
<b>Building a recommendation engine</b>	<b>347</b>
Recommending products based on previous purchases	347
<b>Summary</b>	<b>356</b>
<b>Chapter 10: Building an E-Learning Platform</b>	<b>357</b>
<b>Setting up the e-learning project</b>	<b>358</b>
<b>Building the course models</b>	<b>359</b>
Registering the models in the administration site	361
Using fixtures to provide initial data for models	362
<b>Creating models for diverse content</b>	<b>365</b>
Using model inheritance	366
Abstract models	366
Multi-table model inheritance	367
Proxy models	367
Creating the content models	368
Creating custom model fields	370
Adding ordering to module and content objects	372
<b>Creating a CMS</b>	<b>377</b>
Adding an authentication system	377

---

Creating the authentication templates	378
Creating class-based views	381
Using mixins for class-based views	381
Working with groups and permissions	383
Restricting access to class-based views	385
<b>Managing course modules and their contents</b>	<b>391</b>
Using formsets for course modules	391
Adding content to course modules	396
Managing modules and their contents	402
Reordering modules and their contents	407
Using mixins from django-braces	407
<b>Summary</b>	<b>411</b>
<b>Chapter 11: Rendering and Caching Content</b>	<b>413</b>
<b>Displaying courses</b>	<b>414</b>
<b>Adding student registration</b>	<b>419</b>
Creating a student registration view	419
Enrolling on courses	422
<b>Accessing the course contents</b>	<b>425</b>
Rendering different types of content	429
<b>Using the cache framework</b>	<b>432</b>
Available cache backends	432
Installing Memcached	433
Cache settings	434
Adding Memcached to your project	434
Monitoring Memcached	435
Cache levels	436
Using the low-level cache API	436
Caching based on dynamic data	438
Caching template fragments	440
Caching views	441
Using the per-site cache	441
<b>Summary</b>	<b>442</b>
<b>Chapter 12: Building an API</b>	<b>443</b>
<b>Building a RESTful API</b>	<b>444</b>
Installing Django REST framework	444
Defining serializers	445
Understanding parsers and renderers	446
Building list and detail views	447
Creating nested serializers	450
Building custom API views	452
Handling authentication	453

---

Adding permissions to views	454
Creating viewsets and routers	456
Adding additional actions to viewsets	457
Creating custom permissions	458
Serializing course contents	459
Consuming the REST API	461
<b>Summary</b>	<b>465</b>
<b>Chapter 13: Building a Chat Server</b>	<b>467</b>
<b>Creating a chat application</b>	<b>467</b>
Implementing the chat room view	468
Deactivating the per-site cache	471
<b>Real-time Django with Channels</b>	<b>471</b>
Asynchronous applications using ASGI	471
The request/response cycle using Channels	472
<b>Installing Channels</b>	<b>473</b>
<b>Writing a consumer</b>	<b>476</b>
<b>Routing</b>	<b>477</b>
<b>Implementing the WebSocket client</b>	<b>478</b>
<b>Enabling a channel layer</b>	<b>484</b>
Channels and groups	484
Setting up a channel layer with Redis	484
Updating the consumer to broadcast messages	486
Adding context to the messages	490
<b>Modifying the consumer to be fully asynchronous</b>	<b>494</b>
<b>Integrating the chat with existing views</b>	<b>495</b>
<b>Summary</b>	<b>496</b>
<b>Chapter 14: Going Live</b>	<b>497</b>
<b>Creating a production environment</b>	<b>497</b>
Managing settings for multiple environments	498
Using PostgreSQL	500
Checking your project	501
Serving Django through WSGI	501
Installing uWSGI	502
Configuring uWSGI	502
Installing NGINX	505
The production environment	506
Configuring NGINX	506
Serving static and media assets	509
Securing connections with SSL/TLS	511
Creating an SSL/TLS certificate	511

Configuring NGINX to use SSL/TLS	512
Configuring your Django project for SSL/TLS	514
Redirecting HTTP traffic over to HTTPS	515
Using Daphne for Django Channels	516
Using secure connections for WebSockets	517
Including Daphne in the NGINX configuration	518
Creating a custom middleware	520
Creating a subdomain middleware	522
Serving multiple subdomains with NGINX	523
<b>Implementing custom management commands</b>	<b>524</b>
<b>Summary</b>	<b>527</b>
<b>Other Books You May Enjoy</b>	<b>529</b>
<b>Index</b>	<b>533</b>

---





# Preface

Django is a powerful Python web framework that encourages rapid development and clean, pragmatic design, while offering a relatively shallow learning curve. This makes it attractive to both novice and expert programmers.

This book will guide you through the entire process of developing professional web applications with Django. The book not only covers the most relevant aspects of the framework, but it will also teach you how to integrate other popular technologies into your Django projects.

The book will walk you through the creation of real-world applications, solving common problems, and implementing best practices, using a step-by-step approach that is easy to follow.

After reading this book, you will have a good understanding of how Django works and how to build practical, advanced web applications.

## Who this book is for

This book is intended for developers with Python knowledge who wish to learn Django in a pragmatic way. Perhaps you are completely new to Django, or you already know a little but you want to get the most out of it. This book will help you to master the most relevant areas of the framework by building practical projects from scratch. You need to have familiarity with programming concepts in order to read this book. Some previous knowledge of HTML and JavaScript is assumed.