

目次

はじめに	iii
本書の対象読者と必要な事前知識、および構成	v
本書のサンプルの動作環境と付属データ・会員特典データについて	vii

Chapter

1

コードスニペット共有サイトの開発 1

1.1 Django プロジェクトのはじめかた 2

1.1.1 本章で作成するアプリケーション	2
1.1.2 Django プロジェクトの作成	5
1.1.3 プロジェクトの設定を変更する	7
1.1.4 まとめ	10

1.2 Django アプリケーションの作成と Hello World 10

1.2.1 アプリケーションの作成	10
1.2.2 Hello World の表示（ビューの開発）	12
1.2.3 Hello World の表示（ルーティングの設定）	14
1.2.4 ルーティングの設定	17
1.2.5 まとめ	20

1.3 データベースとの連携 20

1.3.1 モデルの定義	20
1.3.2 データベースマイグレーション	22
1.3.3 管理画面の使い方	23
1.3.4 データベースの基本操作	26
1.3.5 まとめ	29

1.4	テンプレートエンジンによるHTMLの生成	29
1.4.1	HTMLの生成	29
1.4.2	トップページにスニペットの一覧を表示	32
1.4.3	スニペットの詳細ページ	37
1.4.4	スニペットの登録・編集ページ	40
1.4.5	まとめ	46
1.5	テンプレートの部品化とCSSの適用	47
1.5.1	Bootstrap5を使って見た目を整える	47
1.5.2	自分で定義したCSSを適用する	49
1.5.3	テンプレートの部品化	51
1.5.4	スニペット登録・編集フォームにスタイルを適用する	55
1.5.5	Pygmentsを使った構文ハイライト	56
1.5.6	まとめ	59
1.6	ユーザー認証をサクッと作る	60
1.6.1	認証用のDjangoアプリケーションの追加	60
1.6.2	認証用クラスベースビューを利用する	60
1.6.3	汎用クラスベースビューを使ったユーザー登録の実装	64
1.6.4	まとめ	67
1.7	宿題: コメント機能の実装	68
1.7.1	まとめ	70

Chapter

2

モデル定義とクエリ操作 71

2.1	Django ORMによるモデル定義	72
2.1.1	ORMによるモデル定義の基本	72
2.1.2	Metaオプション	74
2.1.3	モデルフィールドの定義	75
2.2	Django ORMによるクエリ操作	77
2.2.1	データベースの準備と設定	77
2.2.2	モデルAPIによるデータの作成・更新・削除	80

2.2.3	モデルマネージャーを使ったデータの操作	82
2.2.4	モデルマネージャーやQuerySetのカスタマイズ	85
2.2.5	明示的なトランザクションの管理	87
2.2.6	デッドロックに気をつける	88
2.3	N+1 問題の概要とその対策	89
2.3.1	select_relatedによるテーブル結合	90
2.3.2	prefetch_relatedによる回避策	92
2.4	インデックスによる効率的なデータの取り出し	96
2.4.1	InnoDBストレージエンジン	96
2.4.2	EXPLAINによるクエリ解析	100
2.4.3	複合インデックスとカバリングインデックス	102
2.5	マイグレーション	104
2.5.1	マイグレーションの処理をSQLで記述する	104
2.5.2	マイグレーションの処理をPythonで記述する	105
2.5.3	マイグレーションを慎重に行うための3つのコマンド	106
2.5.4	マイグレーションファイルのコンフリクトを修正する	107
2.6	SQLインジェクションによる攻撃を理解する	110

Chapter

3

ビュー 113

3.1 ビューの基本 114

3.1.1	ビューの役割と呼び出しの流れ	114
3.1.2	HttpResponseオブジェクト	114
3.1.3	HttpRequestオブジェクト	116
3.1.4	URL変数を受け取る	117
3.1.5	例外を使ったエラー処理	118

3.2 再利用性の高いビューの開発 119

3.2.1	デコレーターを利用する	119
3.2.2	クラスベースビューやMixinを利用する	121
3.2.3	Djangoミドルウェアに記述する	123

3.3 URL ディスパッチャー 126

- 3.3.1** URLconf の設定方法と URL の逆引き 126
- 3.3.2** 自前のパスコンバーターの利用 128
- 3.3.3** エラーページのカスタマイズ 130

3.4 HttpRequest と HttpResponse を使いこなす 132

- 3.4.1** HTTP の基礎知識 132
- 3.4.2** x-www-form-urlencoded 形式のデータの読み込み 135
- 3.4.3** multipart/form-data 形式のデータの読み込み 136
- 3.4.4** HttpResponse オブジェクトを理解する 138
- 3.4.5** condition デコレーターを使った ETag や Last-Modified ヘッダーの制御 139

Chapter

4

テンプレートエンジン 143**4.1 Django テンプレート言語の基本 144**

- 4.1.1** 描画までの基本の流れ 144
- 4.1.2** Python オブジェクトの評価や条件分岐・ループ 146
- 4.1.3** テンプレート継承・埋め込み 148

4.2 テンプレートエンジンの設定 151

- 4.2.1** テンプレートファイルの探索 151
- 4.2.2** 同じ名前のテンプレートファイルを配置する際の注意事項 152

**4.3 テンプレートタグと
テンプレートフィルターを使いこなす 152**

- 4.3.1** コンテキストプロセッサ 153
- 4.3.2** テンプレートフィルター 154
- 4.3.3** テンプレートタグ 157

4.4 テンプレートのセキュリティ対策 159

- 4.4.1** クロスサイトスクリプティング (XSS) による攻撃を理解する 159
- 4.4.2** Django における XSS の対策 162

Chapter

5

フォーム

165

5.1 フォームの基本と全体像 166**5.1.1** ModelFormを使ったフォームの定義 166**5.1.2** バリデーション処理 167**5.1.3** クリーニング(正規化)処理 168**5.1.4** HTMLの生成 169**5.2 バリデーション処理のカスタマイズ** 171**5.2.1** バリデーション処理の使い分け 171**5.2.2** フォームのバリデーション処理の流れと使い分け 173**5.3 HTML フォーム生成時のカスタマイズ** 175**5.3.1** widgetsによるデザインのカスタマイズ 175**5.3.2** widgetsではなくテンプレート側で
HTMLタグのアトリビュートを修正する 176**5.4 ModelFormをいつ使うべきか、
あるいはいつ使うべきではないか** 177**5.5 クロスサイトリクエストフォージェリ (CSRF)
による攻撃を理解する** 178

Chapter

6

テスト

181

6.1 テストの観点と考え方 182**6.1.1** 何をどこまでテストするべきか 182**6.1.2** モデルのテスト 182**6.1.3** フォームのテスト 184**6.1.4** ビューのテスト 186**6.1.5** テンプレートのテスト 189**6.1.6** まとめ 192

6.2 実践的なテストテクニック 193

- 6.2.1** トランザクションが絡むテストの実行 193
- 6.2.2** Fixture データの用意 194
- 6.2.3** モックを用いたテスト 197
- 6.2.4** サブテストによる構造化 200
- 6.2.5** タグによるテストの分類 202
- 6.2.6** まとめ 203

6.3 テストに関連するツールチェイン 203

- 6.3.1** テストランナー 203
- 6.3.2** 複数のバージョンのテスト 204
- 6.3.3** カバレッジでテストの状態を知る 204
- 6.3.4** Selenium で Web ブラウザを使った機能テストを記述する 205
- 6.3.5** BDD フレームワークの活用 206
- 6.3.6** まとめ 208

6.4 まとめ：さらにテストを使いこなすために 208

Chapter

7

認証・認可 209**7.1 認証処理の基本とカスタマイズ** 210

- 7.1.1** Django の提供する認証機能 210
- 7.1.2** クラスベースビューを使った認証処理のカスタマイズ 212
- 7.1.3** UserCreationForm を使ったユーザー登録画面の実装 213
- 7.1.4** 認証バックエンドによるログイン処理のカスタマイズ 215

7.2 ユーザーモデルのカスタマイズ 219

- 7.2.1** User モデルに対して 1 対 1 の関係をもつモデルを定義する 220
- 7.2.2** AbstractUser や AbstractBaseUser を継承する 221
- 7.2.3** username の取扱いに関する注意点 225

7.3 権限管理（認可） 226

- 7.3.1** BooleanField のフラグによる権限管理 227
- 7.3.2** Permissions や Group による権限管理 229
- 7.3.3** まとめ 232

7.4	パターン別認証処理実装ガイド	232
7.4.1	アカウントのEメールチェック	233
7.4.2	OAuth 認証の実装	234

Chapter

8

Web API開発 237

8.1	Django REST FrameworkによるJSON APIの実装	238
8.1.1	SerializerによるJSONとの変換	238
8.1.2	ViewSetによるビューの記述	240
8.1.3	Routerを使ったURLディスパッチ	243
8.1.4	まとめ	245
8.2	Web APIの設計プラクティス	246
8.2.1	ページネーションの実装	246
8.2.2	JWT (JSON Web Token) による認証	250
8.2.3	APIのリクエスト制限	252

Chapter

9

ケース別レシピ集 255

9.1	静的ファイルの配信	256
9.2	メッセージフレームワーク	257
9.3	国際化 (多言語対応)	260
9.4	Eメールの送信	264
9.5	GunicornやuWSGIを使用する	266
9.6	デーモン管理ツールの導入	269

9.7	Nginx を HTTP Proxy として導入する	271
9.8	キャッシュフレームワーク	273
9.9	設定ファイルの分割	275
9.10	Django 管理サイトのカスタマイズ	276
9.11	管理コマンドの作成	280

Appendix

関連リソース	283
---------------	-----

A.1	参考資料（書籍）	284
A.2	参考資料（Web サイト）	284
A.3	本書で紹介した サードパーティーライブラリー一覧	284

おわりに	287
謝辞	289
索引	290
著者プロフィール	293