
■ Django アプリ 結合テスト仕様書（正常系 + 異常系）

※ GitHub Wiki 用 Markdown 版

1. テスト概要

項目	内容
テスト名称	Django Topics & Entries 結合テスト
対象機能	トピック一覧、トピック詳細、エントリー追加、エントリー編集
目的	画面遷移・フォーム処理・バリデーション・エラー処理の総合確認
前提条件	Django サーバー起動済み、DB 初期化済み

2. テスト対象画面

- / (トップページ)
 - /topics/ (トピック一覧)
 - /topics/<id>/ (トピック詳細)
 - /topics/<id>/new_entry/ (新規エントリー作成)
 - /entries/<id>/edit/ (エントリー編集)
-

3. 正常系テストケース

TC-N01：トップページ → トピック一覧へ遷移

項目	内容
目的	トップページから一覧へ遷移できる
手順	/ → 「Topics」クリック
期待結果	/topics/ に遷移し、トピック一覧が表示される

TC-N02：トピック一覧 → トピック詳細へ遷移

項目	内容
目的	トピック一覧から詳細へ遷移できる
手順	/topics/ → 任意のトピック名クリック

項目	内容
期待結果	<code>/topics/<id>/</code> に遷移し、エントリー一覧が表示される

TC-N03：トピック詳細 → 新規エントリー作成ページへ遷移

項目	内容
目的	新規エントリー作成画面が開ける
手順	<code>/topics/<id>/</code> → 「Add new entry」クリック
期待結果	<code>/topics/<id>/new_entry/</code> に遷移し、フォームが表示される

TC-N04：新規エントリーの登録

項目	内容
目的	正常にエントリーが登録される
手順	フォームにテキスト入力 → Save
期待結果	<code>/topics/<id>/</code> にリダイレクトされ、入力内容が一覧に表示される

TC-N05：エントリー編集ページへ遷移

項目	内容
目的	編集画面が開ける
手順	<code>/topics/<id>/</code> → 任意のエントリーの「Edit」クリック
期待結果	<code>/entries/<id>/edit/</code> に遷移し、フォームが表示される

TC-N06：エントリー編集の保存

項目	内容
目的	編集内容が正常に保存される
手順	テキスト変更 → Save
期待結果	<code>/topics/<topic_id>/</code> にリダイレクトされ、変更内容が反映される

4. 異常系テストケース

TC-E01：存在しないトピックIDにアクセス

項目	内容
目的	不正IDアクセス時のエラー確認
手順	/topics/9999/ にアクセス
期待結果	404 ページが表示される

TC-E02 : 存在しないエントリーIDにアクセス

項目	内容
目的	不正IDアクセス時のエラー確認
手順	/entries/9999/edit/ にアクセス
期待結果	404 ページが表示される

TC-E03 : 新規エントリーで空入力のまま保存

項目	内容
目的	バリデーションエラー確認
手順	フォーム未入力 → Save
期待結果	エラーメッセージ表示、登録されない

TC-E04 : エントリー編集で空入力のまま保存

項目	内容
目的	編集時のバリデーション確認
手順	内容削除 → Save
期待結果	エラーメッセージ表示、更新されない

TC-E05 : 極端に長い文字列を入力

項目	内容
目的	max_length 超過時の挙動確認
手順	10000文字入力 → Save
期待結果	バリデーションエラー表示

TC-E06 : HTMLタグを入力 (XSS対策)

項目	内容
----	----

項目	内容
目的	HTMLタグが無害化されるか確認
手順	<script>alert(1)</script> を入力
期待結果	表示時にタグがエスケープされる

TC-E07 : 別トピックのエントリー編集ページにアクセス

項目	内容
目的	トピックとエントリーの整合性確認
手順	/topics/1/entries/999/edit/ のように不正URLアクセス
期待結果	404 または適切なエラー

TC-E08 : CSRF トークンを削除して送信

項目	内容
目的	CSRF 保護が有効か確認
手順	開発者ツールで CSRF トークン削除 → POST
期待結果	403 Forbidden

TC-E09 : GET メソッドで保存処理を実行

項目	内容
目的	不正メソッドで保存されないことを確認
手順	/topics/<id>/new_entry/ に GET で保存リクエスト
期待結果	保存されずフォーム画面が表示される

TC-E10 : DB 障害時の挙動確認

項目	内容
目的	DB エラー時の挙動確認
手順	settings.py の DB 設定を無効化 → アクセス
期待結果	500 エラーページ表示

5. まとめ

この仕様書は以下をすべて網羅しています：

- 正常系（画面遷移・登録・編集）
- 異常系（バリデーション・不正URL・セキュリティ・DB障害）
- GitHub Wiki で読みやすい Markdown 形式

Wiki に貼ると見栄えも良く、レビューしやすい構成になっている。

必要なら：

- **Wiki の目次 (TOC) 自動生成版**
- **pytest 用の自動テストコード**
- **Excel 形式のテスト仕様書**

なども作れるよ。