

プロジェクト生成 - テンプレート

プロジェクトジェネレーターは、初期設定、セキュリティ、データベース、初期APIエンドポイントなどの多くが含まれているため、プロジェクトの開始に利用できます。

プロジェクトジェネレーターは常に非常に意見が分かれる設定がされており、ニーズに合わせて更新および調整する必要があります。しかしきっと、プロジェクトの良い出発点となるでしょう。

フルスタック FastAPI PostgreSQL

GitHub: <https://github.com/tiangolo/full-stack-fastapi-postgresql>

フルスタック FastAPI PostgreSQL - 機能

- 完全な **Docker** インテグレーション (Dockerベース)。
- Docker Swarm モードデプロイ。
- ローカル開発環境向けの **Docker Compose** インテグレーションと最適化。
- UvicornとGunicornを使用した **リリース可能な** Python web サーバ。
- Python **FastAPI** バックエンド:
 - **高速**: **NodeJS** や **Go** 並みのとても高いパフォーマンス (Starlette と Pydanticのおかげ)。
 - **直感的**: 素晴らしいエディタのサポートや **補完**... デバッグ時間の短縮。
 - **簡単**: 簡単に利用、習得できるようなデザイン。ドキュメントを読む時間を削減。
 - **短い**: コードの重複を最小限に。パラメータ宣言による複数の機能。
 - **堅牢性**: 自動対話ドキュメントを使用した、本番環境で使用できるコード。
 - **標準規格準拠**: API のオープンスタンダードに基く、完全な互換性: [OpenAPI](#) や [JSON スキーマ](#)。
 - 自動バリデーション、シリアライゼーション、対話的なドキュメント、OAuth2 JWT トークンを用いた認証などを含む、[その他多くの機能](#)。
- **セキュアなパスワード** ハッシュ化 (デフォルトで)。
- **JWT トークン** 認証。
- **SQLAlchemy** モデル (Flask用の拡張と独立しているので、Celery ワーカーと直接的に併用できます)。
- 基本的なユーザーモデル (任意の修正や削除が可能)。
- **Alembic** マイグレーション。
- **CORS** (Cross Origin Resource Sharing (オリジン間リソース共有))。
- **Celery** ワーカー。バックエンドの残りの部分からモデルとコードを選択的にインポートし、使用可能。
- Dockerと統合された**Pytest**ベースのRESTバックエンドテスト。データベースに依存せずに、全てのAPIをテスト可能。Docker上で動作するので、毎回ゼロから新たなデータストアを構築可能。(ElasticSearch、MongoDB、CouchDBなどを使用して、APIの動作をテスト可能)
- Atom HydrogenやVisual Studio Code Jupyterなどの拡張機能を使用した、リモートまたはDocker開発用の**Jupyterカーネル**との簡単なPython統合。
- **Vue** フロントエンド:
 - Vue CLIにより生成。
 - **JWT認証**の処理。
 - ログインビュー。
 - ログイン後の、メインダッシュボードビュー。
 - メインダッシュボードでのユーザー作成と編集。
 - セルフユーザー版
 - **Vuex**。
 - **Vue-router**。
 - 美しいマテリアルデザインコンポーネントのための**Vuetify**。
 - **TypeScript**。
 - **Nginx**ベースのDockerサーバ (Vue-routerとうまく協調する構成)。

- Dockerマルチステージビルド。コンパイルされたコードの保存やコミットが不要。
- ビルド時にフロントエンドテスト実行 (無効化も可能)。
- 可能な限りモジュール化されているのでそのまま使用できますが、Vue CLIで再生成したり、必要に応じて作成したりして、必要なものを再利用可能。
- PostgreSQLデータベースのための**PGAdmin**。(PHPMyAdminとMySQLを使用できるように簡単に変更可能)
- Celeryジョブ監視のための**Flower**。
- **Traefik**を使用してフロントエンドとバックエンド間をロードバランシング。同一ドメインに配置しパスで区切る、ただし、異なるコンテナで処理。
- Traefik統合。Let's Encrypt **HTTPS**証明書の自動生成を含む。
- GitLab **CI** (継続的インテグレーション)。フロントエンドおよびバックエンドテストを含む。

フルスタック FastAPI Couchbase

GitHub: <https://github.com/tiangolo/full-stack-fastapi-couchbase>

⚠ 警告 ⚠

ゼロから新規プロジェクトを始める場合は、ここで代替案を確認してください。

例えば、[フルスタック FastAPI PostgreSQL](#)のプロジェクトジェネレーターは、積極的にメンテナンスされ、利用されているのでより良い代替案かもしれません。また、すべての新機能と改善点が含まれています。

Couchbaseベースのジェネレーターは今も無償提供されています。恐らく正常に動作するでしょう。また、すでにそのジェネレーターで生成されたプロジェクトが存在する場合でも (ニーズに合わせてアップデートしているかもしれません)、同様に正常に動作するはずです。

詳細はレポジトリのドキュメントを参照して下さい。

フルスタック FastAPI MongoDB

...時間の都合等によっては、今後作成されるかもしれません。😄🍷

spaCyとFastAPIを使用した機械学習モデル

GitHub: <https://github.com/microsoft/cookiecutter-spacy-fastapi>

spaCyとFastAPIを使用した機械学習モデル - 機能

- **spaCy** のNERモデルの統合。
- **Azure Cognitive Search** のリクエストフォーマットを搭載。
- リリース可能な UvicornとGunicornを使用したPythonウェブサーバ。
- **Azure DevOps** のKubernetes (AKS) CI/CD デプロイを搭載。
- **多言語** プロジェクトのために、セットアップ時に言語を容易に選択可能 (spaCyに組み込まれている言語の中から)。
- **簡単に拡張可能**。spaCyだけでなく、他のモデルフレームワーク (Pytorch、Tensorflow) へ。