**Analisi Tecnica**

|  |  |
| --- | --- |
| **Autore** | Sotamba Acosta Jean Pierre |
| **Classificazione** | Ad uso interno |
| **Stato** | Bozza |
| **Versione** | 1 |
| **Data** **creazione** |  |

**Panoramica del progetto**

Il progetto "orders" è un'applicazione web basata su Django che gestisce ordini e prodotti. La sua struttura segue una divisione modulare per mantenere un'organizzazione chiara e scalabile. Vengono utilizzati Django Rest Framework (DRF) per creare API RESTful e drf-yasg per la documentazione delle API.

**Struttura delle Cartelle e dei File**

**Directory principale: ‘orders/’**

La directory principale contiene i file di configurazione del progetto Django e il gestore di pacchetti

* **manage.py: script per interagire con il progetto django(es.avviare il server, esegure migrazioni, ecc..).**
* **requirements.txt: elenco delle dipendenze del progetto.**
* **Dockerfile: definisce come creare l’immagine Docker del progetto.**
* **docker-compose.yml: configura i servizi Docker per l’applicazione**

**Directory delle applicazioni: ‘apps/’**

1. **order/**
   1. **models.py: definisce il modello ‘Order’**
   2. **api/:** 
      1. **api.py: implementa le viste API per ‘Order.**
      2. **serializers.py: definisce i serializzatori per ‘Order’.**
      3. **urls.py: Mappa le URL alle viste API.**
      4. **test/: contiene i test per le API, i modelli.**
2. **product/**
   1. **models.py: definisce il modello ‘Product’**
   2. **api/:**
      1. **api.py: implementa le viste API per ‘Product’.**
      2. **serializers.py: definisce i serializzatori per ‘Product’.**
      3. **urls.py: mappa le URL alle viste API.**
      4. **test/: contiene i test per le API, i modelli.**

**Configurazione del Progetto: ‘orders/**

Contiene i file di configurazione del progetto DRF.

* **settings/**
  + **base.py:** configurazione base comune a tutti gli ambienti
  + **local.py:** configurazione per l’ambiente di sviluppo
  + **production.py:** configurazione per l’ambiente di produzione
* **urls.py:** Mappa le URL di livello superiore alle aplicazioni.

**Dettaglio dei Moduli**

1. **Modulo order:**
   1. **Modello ‘Order’:** classe python che rappresenta la tabella nel database
   2. **API:**
      1. **order\_api\_view:** gestisce le operazioni di creazione e lettura di ordini
      2. **order\_detail\_api\_view:** gestisce le operazioni di lettura, aggiornamento e cancellazione di un singolo ordine
      3. **filter\_api\_view:** filtra gli ordini in base a criteri specifici
   3. **Serializers:**
      1. **OrderSerializer:** serializza i dati dell’ordine per operazioni di lettura
      2. **OrderCreateUpdateSerializer:** serializza i dati dell’ordine per operazioni di creazione e aggiornamento
2. **Modulo product:**
   1. **Modello ‘Product’:** classe python che rappresenta la tabella nel database
   2. **API:**
      1. **Product\_api\_view:** gestisce le operazioni di creazione e lettura di prodotti
      2. **Product\_detail\_api\_view:** gestisce le operazioni di lettura, aggiornamento e cancellazione di un singolo prodotto
   3. **Serializers:**
      1. **ProductSerializer:** serializza i dati del prodotto per operazioni di lettura e scrittura

**Configurazioni**

1. **Base Settings(‘base.py’):** contiene configurazioni comuni
2. **Local Settings(‘local.py):** contiene configurazioni specifiche per l’ambiente di sviluppo
3. **Production Settings(‘production.py’):** ‘contiene’ specifiche per l’ambiente di produzione, per l’applicazione in questione esso non è impostato correttamente

**Documentazione**

Uso di ‘drf-yasg’ per generare automaticamente la documentazione Swagger delle API.