Invoice Analytics API

1. Overview

The Invoice Analytics API is a RESTful backend system designed to manage and analyze invoice data. It supports operations such as creating, retrieving, updating, and deleting invoices. In addition, it provides real-time analytics features such as total revenue and average invoice amount, supporting multiple currencies through integration with a currency exchange API.

2. System Architecture

Technology Stack

- Backend Framework: Django (with Django REST Framework)
- Database: MongoDB (containerized with Docker)
- ODM: MongoEngine
- External API: ExchangeRate-API for real-time currency conversion
- Containerization: MongoDB runs via Docker image for local development

Project Setup

The project uses a requirements.txt file for dependency management. MongoDB is set up using a Docker container to allow consistent local development without requiring a native installation. Once the Docker container is running and dependencies are installed, the Django application connects to the MongoDB instance using MongoEngine ODM.

3. Core Features

Invoice Management (CRUD)

- POST `/invoices/`: Create a new invoice with amount and currency.
- GET `/invoices/`: List all invoices.
- GET `/invoices/<pk>`: Retrieve a specific invoice.
- PUT `/invoices/<pk>`: Update an existing invoice.
- DELETE `/invoices/<pk>`: Remove an invoice from the system.

When a valid invoice is submitted, it is converted to USD using a live exchange rate and stored in the database along with the conversion metadata.

Analytics Endpoints

- GET `/analytics/total-revenue/?currency=<CUR>`: Calculates and returns the total revenue from all invoices, converted to the specified currency.
- GET `/analytics/average-invoice/?currency=<CUR>`: Calculates and returns the average invoice value, converted to the specified currency.
- GET `/invoices/<pk>/exchange-rate`: Returns the exchange rate used during the creation of a specific invoice.

These endpoints dynamically call the external exchange rate API as needed.

4. Data Model

Invoice

id	String	Auto-generated unique identifier
amount	Float	The invoice's original value
currency	String	The original currency used
converted_amount	Float	Value converted to USD at creation
exchange_rate	Float	Exchange rate used at creation

Invoices are stored in MongoDB, and the schema is defined using MongoEngine.

5. External Integration

Exchange Rate API

The app integrates with the ExchangeRate-API to:

- Retrieve the list of supported currencies.
- Get live conversion rates between two currencies.

Exchange logic is abstracted in utility functions (get_supported_currencies, get_exchange_rate) to allow easier testing and mocking.

6. Deployment & Execution

To run the project locally:

1. Install Dependencies

```
pip install -r requirements.txt
```

2. Run MongoDB Docker Container

```
docker run -d -p 27017:27017 --name invoices-mongo mongo
```

3. Start Django Server

```
python manage.py runserver
```

7. Developer Notes

- A Postman Collection JSON file is included in the project root directory. This can be imported into Postman to test each endpoint interactively.
- Unit tests are written for all key views and functionalities. You can run all tests using:

```
python manage.py test
```