

Employee Management REST API

Project Submission – Python Backend Developer (HabotConnect)

1. Project Overview

This project is a Django REST Framework based Employee Management REST API. It is designed to demonstrate clean backend architecture, RESTful API design, JWT-based authentication, validation, pagination, filtering, and unit testing as part of the HabotConnect hiring assignment.

2. Technology Stack

- Python 3.x
- Django
- Django REST Framework
- SQLite (default database)
- JWT Authentication (SimpleJWT)

3. Key Features

- Secure CRUD operations for employees
- JWT-based authentication for all endpoints
- Validation and proper HTTP status codes
- Filtering by department and role
- Pagination with 10 records per page
- Unit tests covering major endpoints and edge cases

4. API Endpoints

- POST /api/employees/ – Create a new employee
- GET /api/employees/ – List employees (pagination & filtering)
- GET /api/employees/{id}/ – Retrieve a single employee
- PUT /api/employees/{id}/ – Update an employee
- DELETE /api/employees/{id}/ – Delete an employee

5. Authentication Flow

The API uses JWT-based authentication. A user first authenticates using the /api/token/ endpoint to obtain an access token. This token must be sent in the Authorization header as a Bearer token to access secured endpoints.

6. Testing Strategy

Unit tests are written using Django REST Framework's `APITestCase`. Tests cover authenticated CRUD operations, duplicate email validation, and invalid resource access. Django automatically creates and destroys a temporary test database during test execution.

7. Conclusion

This project follows RESTful best practices, secure authentication, clean code structure, and test-driven development principles. It is suitable for demonstration and can be easily extended for production use with a scalable database and role-based permissions.