GitHub Link: https://github.com/mtomasi17/clinic-directory.git

YouTube Link: https://youtu.be/y3o36G5o3kE

Project Synopsis: Clinic Directory Management API using Spring Boot

This Spring Boot project embodies a dynamic Clinic Directory RESTful Web API that enables streamlined administration of clinic operations, personnel, and patient records. The API seamlessly supports all CRUD operations (Create, Read, Update, Delete) while seamlessly persisting data within a MySQL database. The application leverages the power of Java Spring Boot to construct a RESTful interface, adhering to REST architectural principles for data manipulation.

Key Aspects:

1. **Database Schema and Entity Framework:**
    - The initiative encompasses three pivotal entities: Clinic, Employee, and Patient.
    - Clinic encapsulates the core clinic information such as name, address, city, state, ZIP code, and contact number.
    - The Employee entity profiles clinic staff, documenting attributes like first name, last name, contact number, and job designation.
    - The Patient entity encompasses patient particulars like first name, last name, and email.
    - These entities are intrinsically aligned with their respective database tables.
2. **Comprehensive CRUD Functionality:**
    - The API facilitates an exhaustive spectrum of CRUD operations on clinics, employees, and patients.
    - A spectrum of endpoints are provisioned to accomplish tasks such as record creation, retrieval, modification, and deletion.
3. **Establishing Relationships:**
    - The architecture endorses a many-to-many relationship between clinics and patients, affording the flexibility to associate patients with multiple clinics, and vice versa.
    - Furthermore, a one-to-many relationship is established between clinics and employees, granting the capacity for each clinic to host a diverse team of employees.
4. **Endpoint Routing and Controllers:**

- The focal point resides within the ClinicController class, housing a repertoire of endpoints that oversee clinic-oriented functionalities.
- The endpoint spectrum spans from creating, updating, retrieving, and deleting clinics to managing employees and patients associated with clinics.

5. **Efficient Service Layer:**
    - The ClinicService class implements the business logic while orchestrating interactions with the database.
    - The class furnishes methods to persist clinics, employees, and patients, alongside mechanisms for fetching and revising pertinent data.
6. **Robust Error Management:**
    - The GlobalErrorHandler class assumes global responsibility for managing exceptions, granting a custom-tailored approach to error reporting, complemented by judicious assignment of relevant HTTP status codes.
    - The class exhibits specialized prowess in handling the NoSuchElementException scenario, signifying resources that elude discovery.
7. **Structured Database Interaction:**
    - JPA repositories (ClinicDao, EmployeeDao, ClinicRepository, PatientRepository) are judiciously employed to establish database connectivity, facilitating seamless execution of CRUD operations.
8. **Data Model Representation:**
    - The ClinicData class is entrusted with the role of encapsulating clinic-specific data.
    - Nested within are auxiliary classes (ClinicPatient and ClinicEmployee) that proficiently manage patient and employee data, harmoniously embedded within the broader clinic context.
9. **Database Configuration Strategy:**
    - The infrastructure is skillfully configured to interface with a MySQL database for data storage.
10. **Strategic Maven Configuration:**
    - The project is judiciously organized and managed through Maven, thoughtfully leveraging dependencies for Spring Boot, JPA, MySQL, and comprehensive testing frameworks.

API Exploration and Assessment:

- Leveraging tools such as AdvancedRestClient (ARC can diligently test and navigate the API, affording insights into its capabilities**.**

This Spring Boot project embodies the amalgamation of Java Spring Boot and JPA to create a sophisticated RESTful Web API. It's adeptly poised to address the intricate requirements of clinic management, encompassing clinics, employees, and patients. The project demonstrates the utilization of Spring Boot to orchestrate a solution for medical practice administration.