

This is a Java class named `VoterService`, which provides service methods to interact with the voter data. It is marked with the `@Service` annotation, indicating that it is a Spring service and eligible for automatic dependency injection and component scanning. Let's go through the code and explain each part:

1. `@Service`: This annotation marks the class as a Spring service, making it eligible for automatic dependency injection and component scanning.
2. `@RequiredArgsConstructor`: This Lombok annotation automatically generates a constructor with required arguments for the class fields. In this case, it will create a constructor with an argument for the `voterRepository` field.
3. `private final VoterRepository voterRepository;`: This field holds an instance of `VoterRepository`, which is presumably a custom repository for managing voters' data.
4. `public List<Voter> getVoterDetails()`: This method retrieves all voter details by calling the `findAll()` method of the `voterRepository`. It returns a list of all voters present in the database.
5. `public List<Voter> getAllEligibleVoterDetails()`: This method retrieves details of all eligible voters (age 18 and above). It first fetches all voters from the database using `voterRepository.findAll()`. Then, it iterates through the voters, checks their age, and adds them to the `eligibleVoters` list if they are 18 years or older. Finally, it returns the list of eligible voters.
6. `public Optional<Voter> getVoterDetailsById(int id)`: This method retrieves voter details by voter ID. It uses the `voterRepository.findById(id)` method, which returns an `Optional<Voter>`. An `Optional` is used here to handle the possibility that the voter with the given ID may not exist in the database.

In summary, the `VoterService` class provides methods to fetch voter details from the database. It allows retrieving all voters, filtering eligible voters based on age, and fetching voters by their unique ID. This service layer acts as a bridge between the `VoterController` (which handles HTTP requests) and the `VoterRepository` (which manages data access to the database). It encapsulates the business logic related to voter data manipulation and retrieval.