**Today’s Agenda**

1. **Designing The Repository Layer**
2. **Designing The Custom Exception Classes**
3. **Designing The ErrorResponse Class**
4. **Designing The Global Exception Handler Class**
5. **Designing The Repository Layer**

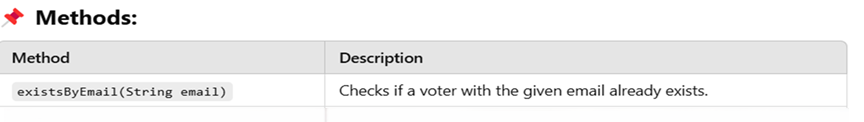
**1️. VoterRepository.java**

**📌 Explanation**

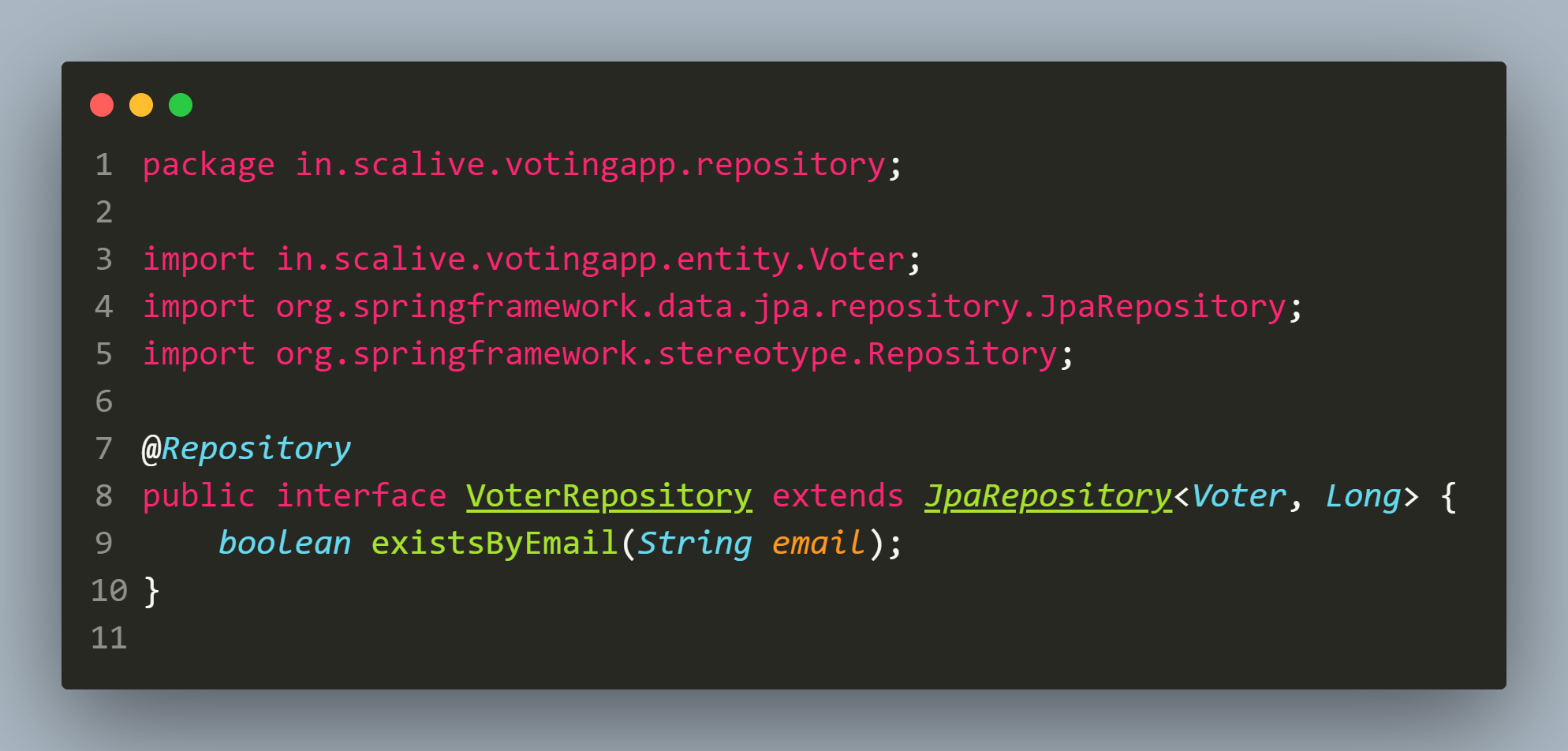
**Purpose:** **Handles** **database operations** for **Voter** entity.

**📌 Annotations Used:**

* **@Repository** → **Declares** this as a **Spring Data JPA repository**.
* **JpaRepository<Voter, Long>** → **Provides** built-in **CRUD operations**.



**Code For VoterRepository Class**



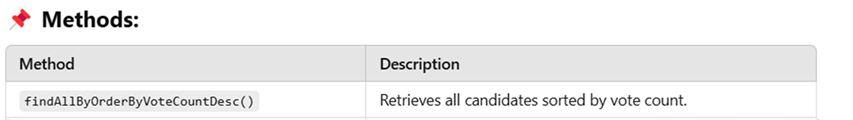
**2. CandidateRepository.java**

**📌 Explanation**

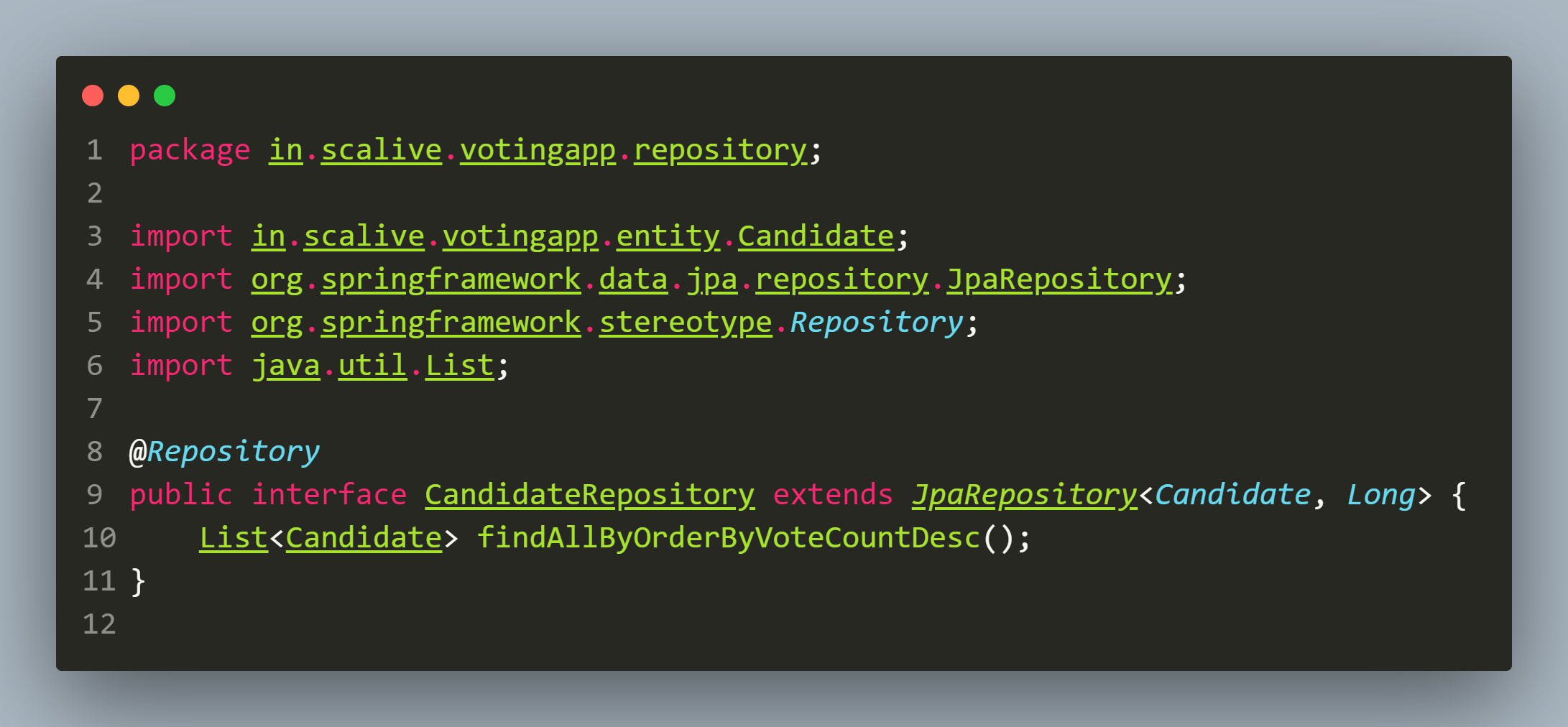
**Purpose:** **Manages** **database interactions** for **Candidate** entity.

**📌 Annotations Used:**

* **@Repository** → **Declares** this as a **repository**.
* **JpaRepository<Candidate, Long>** → **Provides** default **CRUD operations**.



**Code For CandidateRepository Class**



**3.VoteRepository.java**

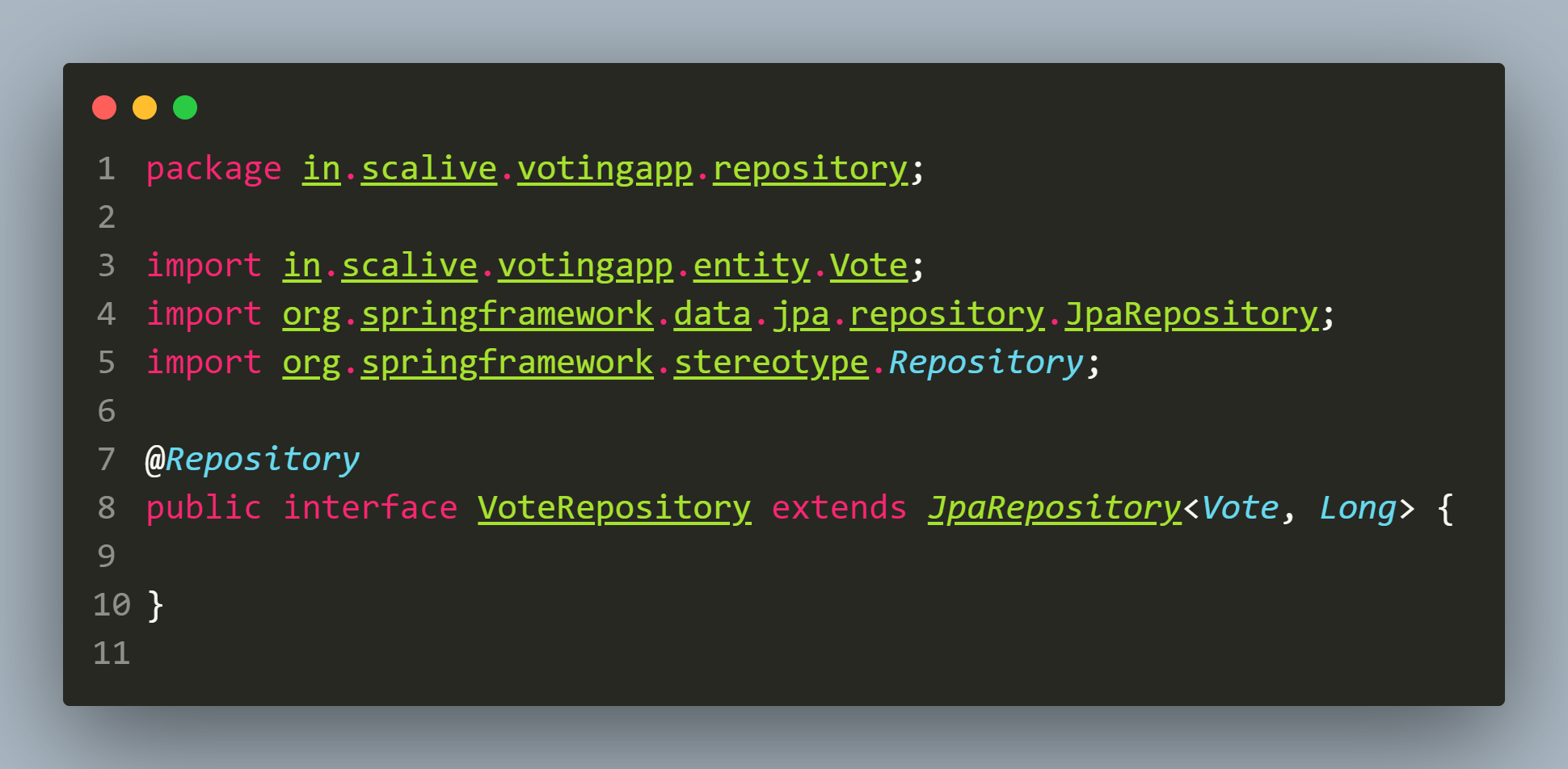
**📌 Explanation**

**Purpose:** **Handles database interactions** for **Vote** entity.

**📌 Annotations Used:**

* **@Repository** → **Declares** this as a **repository**.
* **JpaRepository<Vote, Long>** → **Enables** standard **CRUD operations**.

**Code For VoteRepository Class**



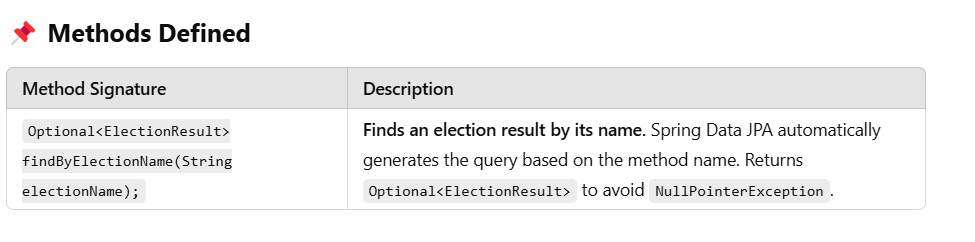
**4. ElectionResultRepository.java**

**📌 Explanation**

**Purpose:** **Manages database interactions** for **ElectionResult** entity.

**📌 Annotations Used:**

* **@Repository** → **Declares** this as a **repository**.
* **JpaRepository<ElectionResult, Long>** → **Enables** standard **CRUD operations**.



**Code For ElectionResultRepository Class**

****

1. **Designing The Custom Exception Classes**

**1. ResourceNotFoundException.java**

**📌 Explanation**

**Purpose:** Handles **cases** where an **entity** (**Voter**, **Candidate**, **Vote**, **ElectionResult**) is not found.

**Code For ResourceNotFoundException Class**

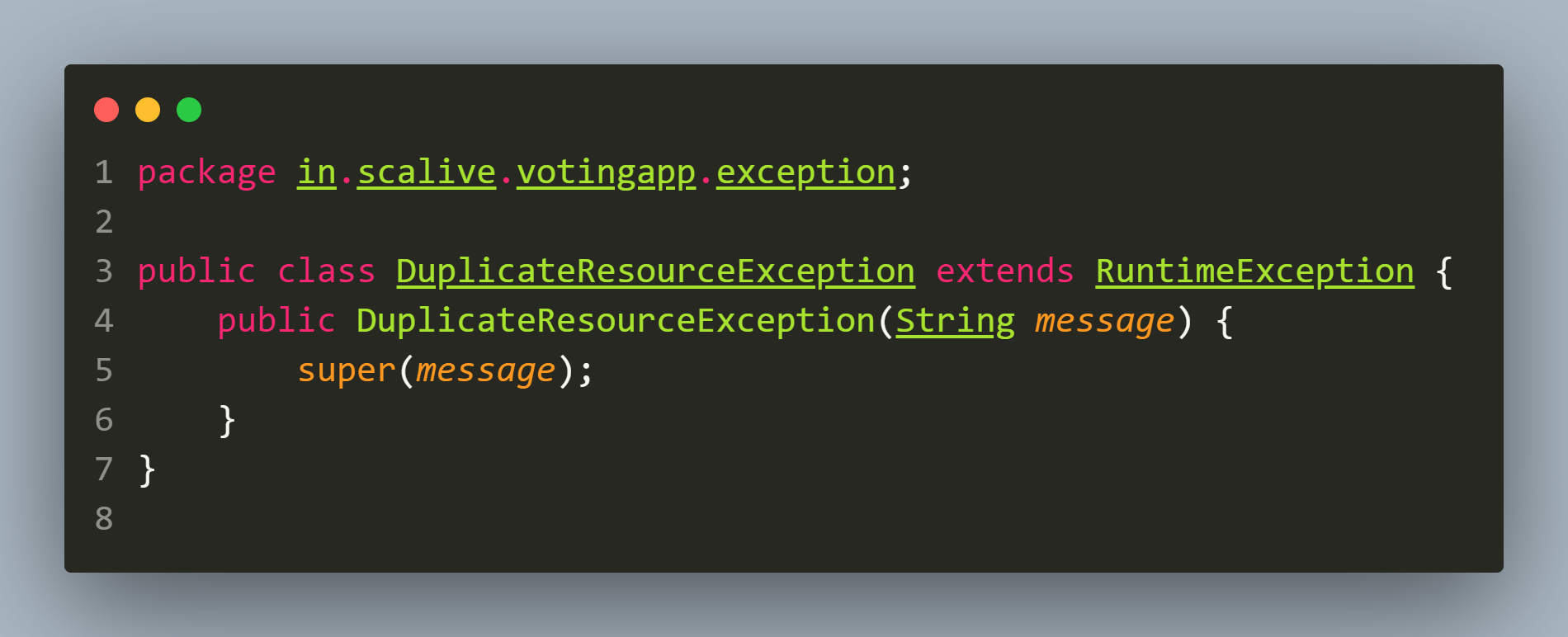
****

**2. DuplicateResourceException.java**

**📌 Explanation**

**Purpose**: **Handles cases** where a **resource already exists** (e.g., **registering** a **voter** with the **same email**).

**Code For ResourceNotFoundException Class**

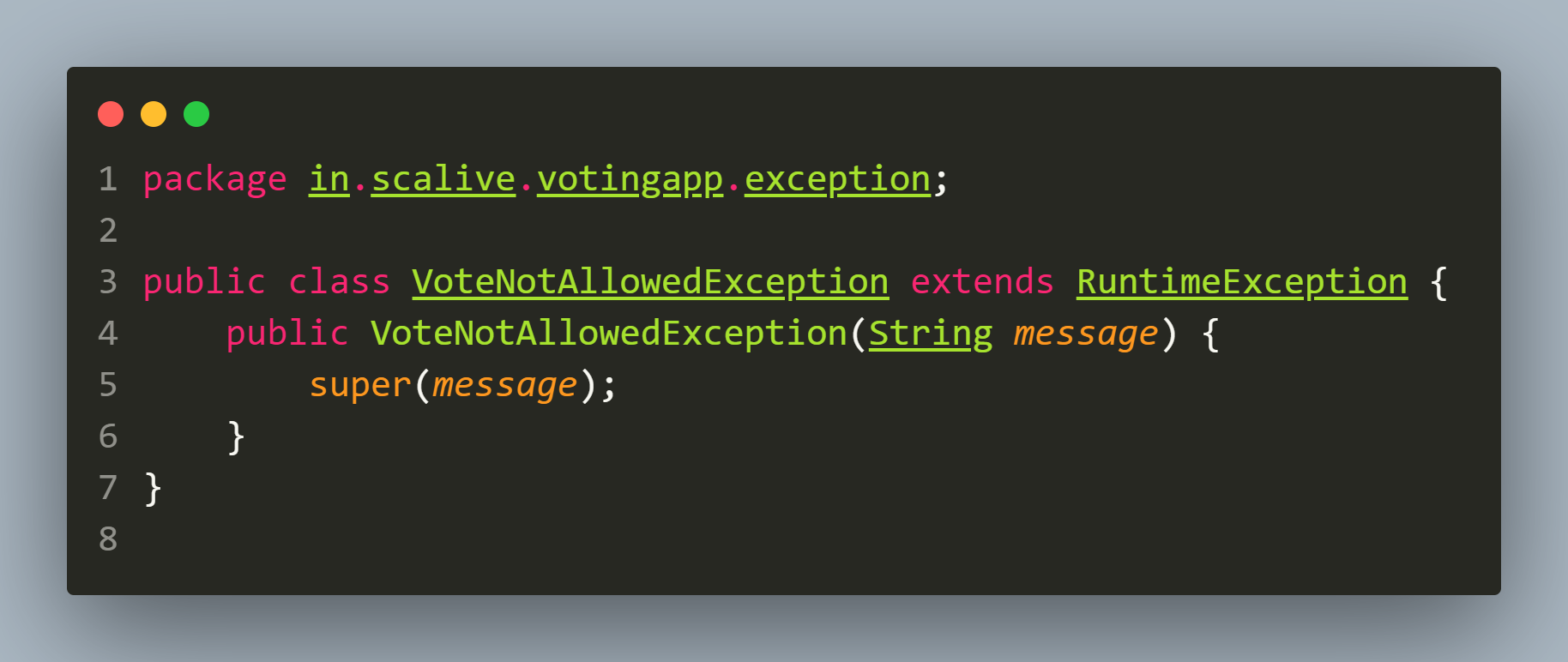
****

**3.VoteNotAllowedException.java**

**📌 Explanation**

**Purpose**: **Handles cases** where a **voter** tries **to vote more than once**.

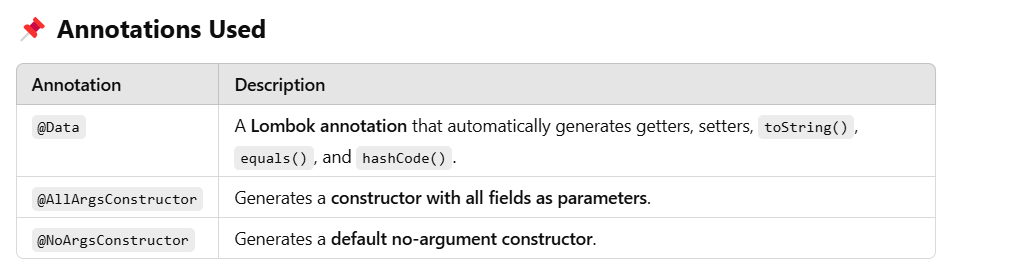
**Code For VoteNotAllowedException Class**

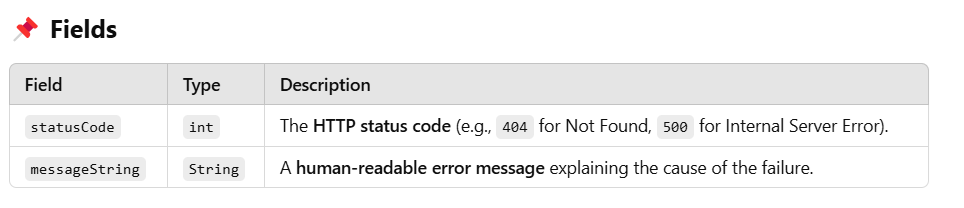
****

1. **Designing The ErrorResponse Classes**

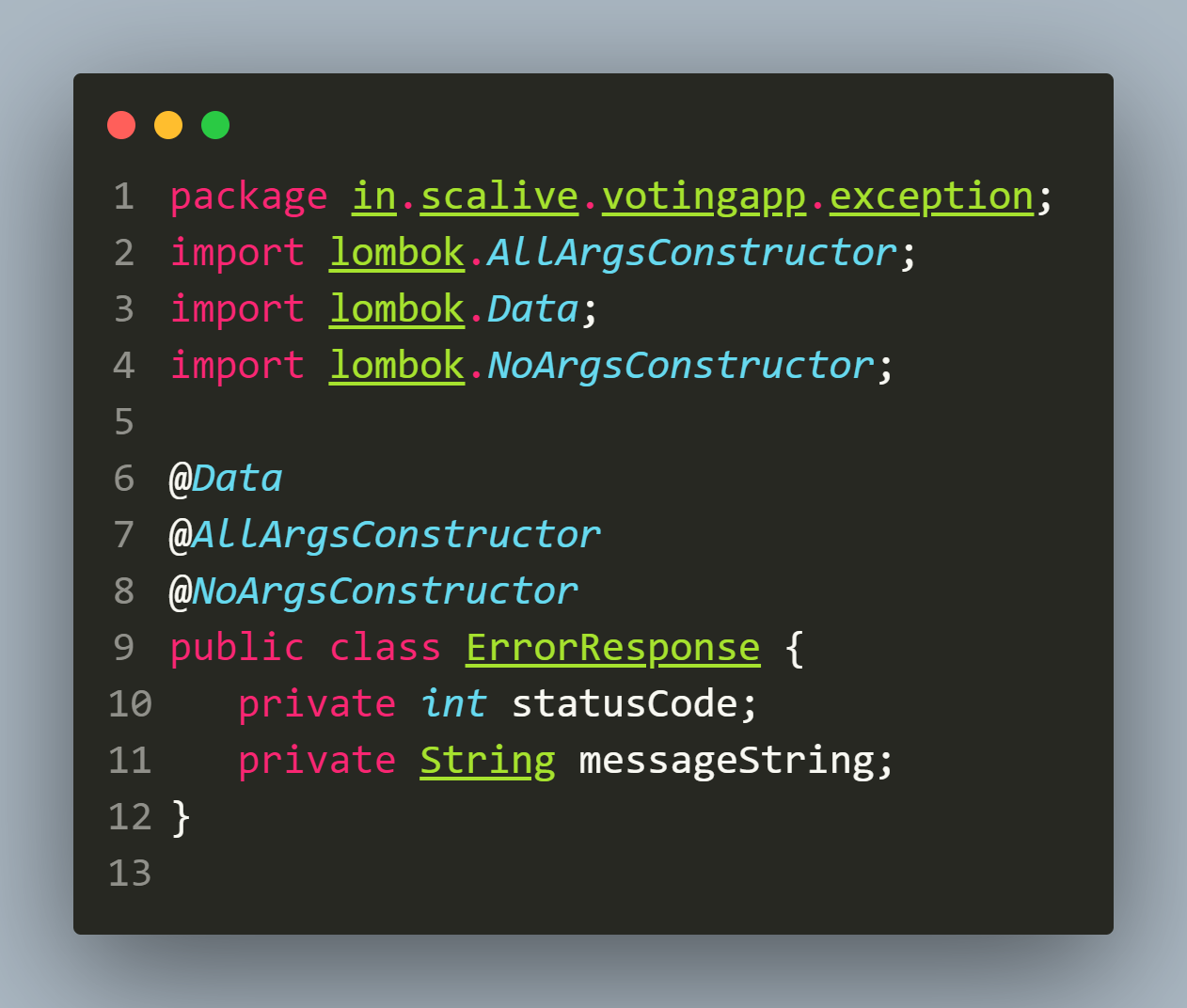
**📌 Purpose**

The **ErrorResponse** class is a **data model** used to represent **structured error responses** in the **application**. It **ensures** that when an **error** occurs, a **consistent and readable JSON response** is **returned** to the client.





**Code For ErrorResponse Class**



**4.Designing The Global Exception Handler Class**

**Code For GlobalExceptionHandler Class**

