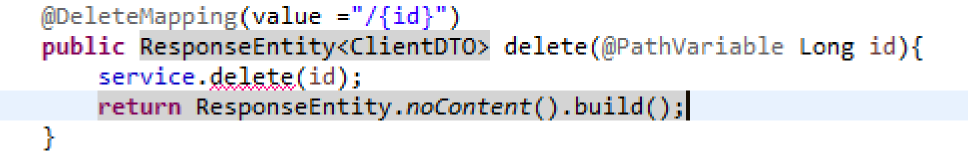
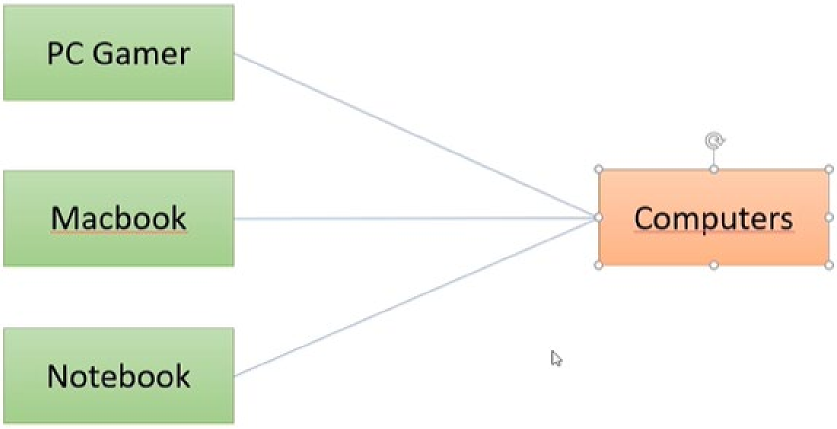
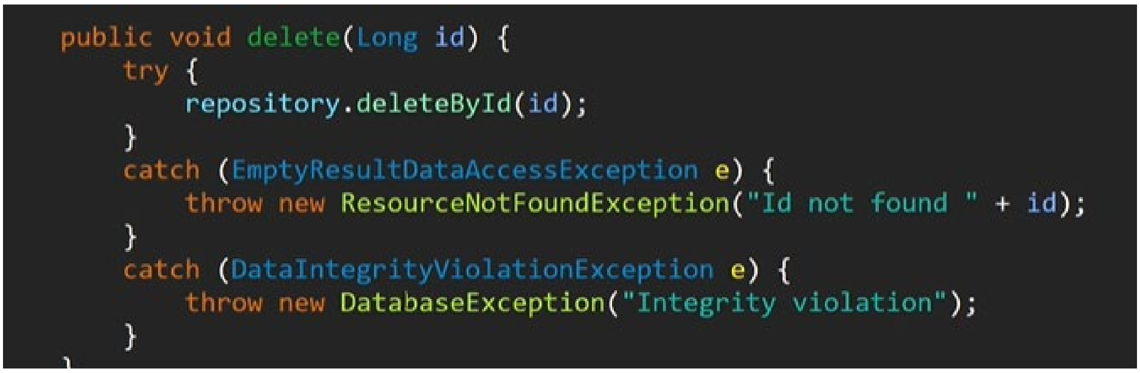
- Implement Delete Resource layer



```
@DeleteMapping(value ="/{id}")
public ResponseEntity<ClientDTO> delete(@PathVariable Long id){
    service.delete(id);
    return ResponseEntity.noContent().build();
}
```

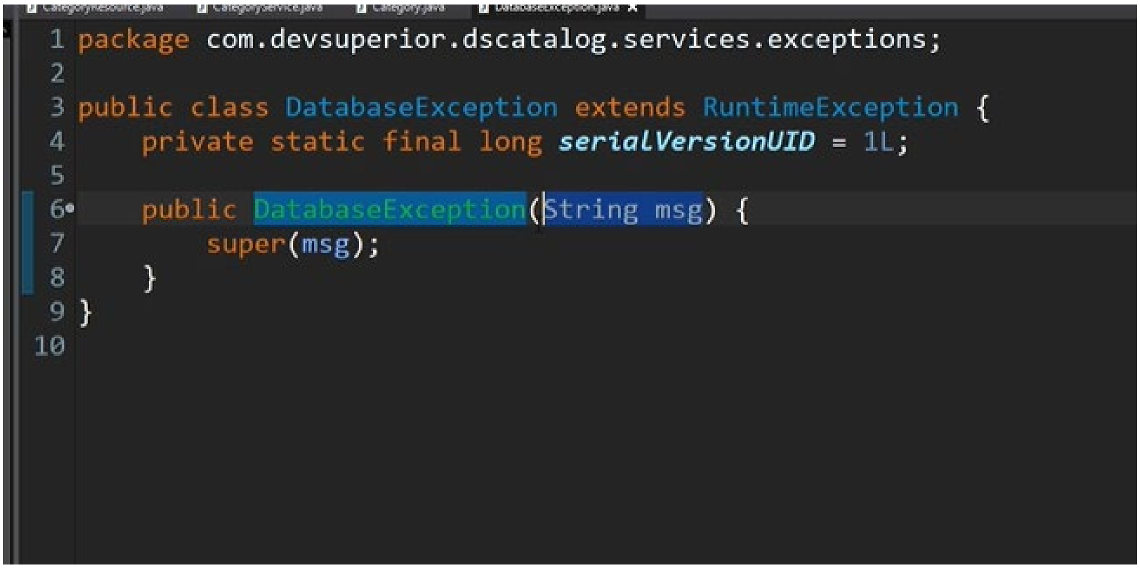
- Now, implement service layer.
- When we implement delete, we can throw two errors. One error if we put a wrong id and another error if we delete a object that is primary key from another object in the database. In this case, we just have the Client entity. If the entity has a relation with another entity, we have to treat this error. Is a DataIntegrityViolationException.





```
public void delete(Long id) {
    try {
        repository.deleteById(id);
    }
    catch (EmptyResultDataAccessException e) {
        throw new ResourceNotFoundException("Id not found " + id);
    }
    catch (DataIntegrityViolationException e) {
        throw new DatabaseException("Integrity violation");
    }
}
```

- Create a new Exception class

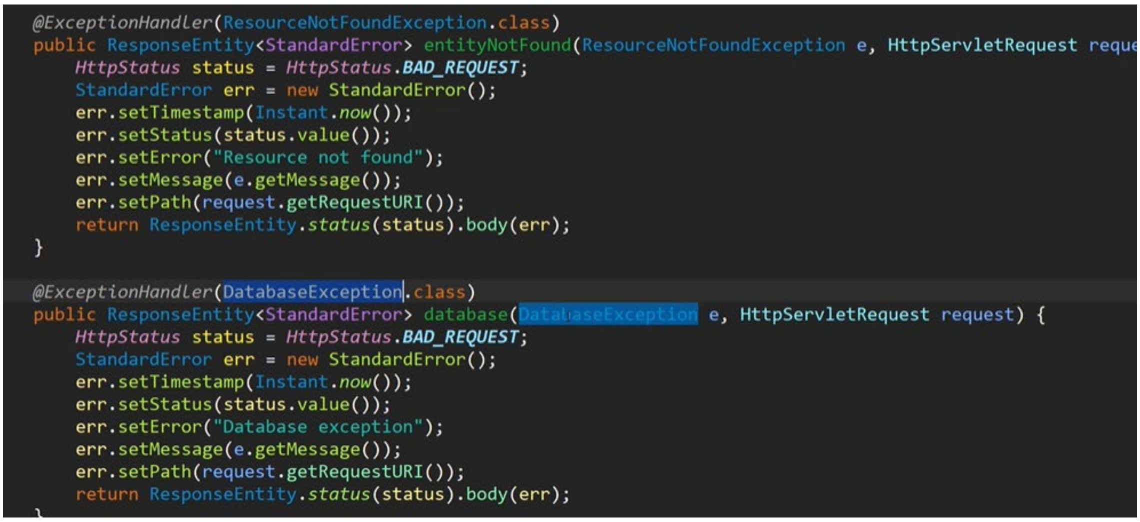


```java
package com.devsuperior.dscatalog.services.exceptions;

public class DatabaseException extends RuntimeException {
    private static final long serialVersionUID = 1L;

    public DatabaseException(String msg) {
        super(msg);
    }
}
```

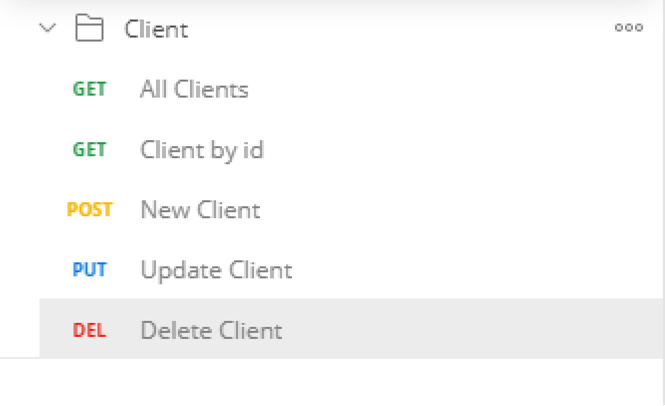
- Now, we gonna pass the response of the error if this occur

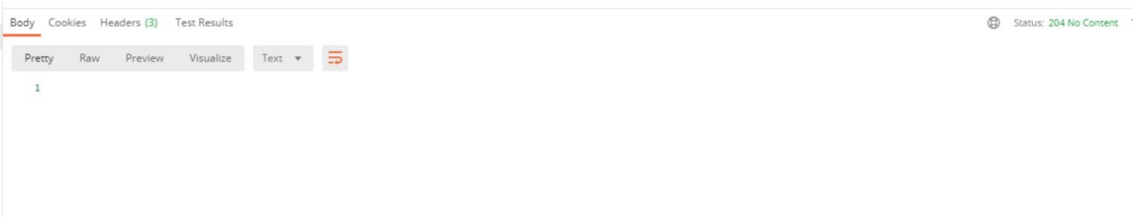


```java
@ExceptionHandler(ResourceNotFoundException.class)
public ResponseEntity<StandardError> entityNotFound(ResourceNotFoundException e, HttpServletRequest reque
    HttpStatus status = HttpStatus.BAD_REQUEST;
    StandardError err = new StandardError();
    err.setTimestamp(Instant.now());
    err.setStatus(status.value());
    err.setError("Resource not found");
    err.setMessage(e.getMessage());
    err.setPath(request.getRequestURI());
    return ResponseEntity.status(status).body(err);
}

@ExceptionHandler(DatabaseException.class)
public ResponseEntity<StandardError> database(DatabaseException e, HttpServletRequest request) {
    HttpStatus status = HttpStatus.BAD_REQUEST;
    StandardError err = new StandardError();
    err.setTimestamp(Instant.now());
    err.setStatus(status.value());
    err.setError("Database exception");
    err.setMessage(e.getMessage());
    err.setPath(request.getRequestURI());
    return ResponseEntity.status(status).body(err);
```

- Let test





Status: 204 No Content

- Give the response 204. Ok