Error and Exception Handling:
https://www.youtube.com/c/NareshIT/search?query=error%20raghu

AOP:
https://www.youtube.com/c/NareshIT/search?query=AOP%20raghu

           Spring Boot ReST # Crud Application # Mini Project

a. Crud Application   [DONE]
b. Test using POSTMAN [DONE]
c. Exception Handling [DONE]
d. Logging
e. Swagger UI Config
f. Unit Testing
g. Connection Pooling
==
                    c. Exception Handling

S#1 Create one Custom Exception class

```java
package in.nareshit.raghu.exception;
public class EmployeeNotFoundException
        extends RuntimeException
{
        private static final long serialVersionUID = 1L;
        public EmployeeNotFoundException() {
                super();
        }
        public EmployeeNotFoundException(String message) {
                super(message);
        }
}
```

S#2 ExceptionHandler class (Global Catch Block)
```java
package in.nareshit.raghu.handler;
//ctrl+shift+O
@RestControllerAdvice //afterThrowingAdvice
public class MyCustomExceptionHandler {

        @ExceptionHandler(EmployeeNotFoundException.class)
        public ResponseEntity<String> handleEmployeeNotFoundException(
                        EmployeeNotFoundException enfe
                        )
        {
                return new ResponseEntity<String>(
                                enfe.getMessage(),
                                HttpStatus.NOT_FOUND);
        }
}
```

S#3 service is throwing exception to RestController

```
    re-throw back to Exception-Handler

add catch block for delete(), getOne(), update()


} catch(EmployeeNotFoundException enfe) {
        //re-throw back to handler
        throw enfe;
}

S#4 At service layer modified code is (3 methods)

        public void updateEmployee(Employee e) {
                getOneEmployee(e.getEmpId());
                util.calculateDetails(e);
                repo.save(e);
        }

        public void deleteEmployee(Integer id) {
                //repo.deleteById(id);
                repo.delete(getOneEmployee(id));
        }

        public Employee getOneEmployee(Integer id) {
                Optional<Employee> opt = repo.findById(id);
                if(opt.isPresent()) {
                        return opt.get();
                }else {
                    throw new EmployeeNotFoundException(
                    "Employee '"+id+"' Not exist");
                }
        }


saveEmployee()   : ResponseEntity<String>   Employee saved
getAllEmployees(): ResponseEntity<List<T>>  Display all List<T>
getOneEmployee() : ResponseEntity<Employee> Display one Type object
deleteEmployee() : ResponseEntity<String>   Employee Deleted
updateEmployee() : ResponseEntity<String>   Employee updated
```