**SPRING-34 - Exception Handling in spring projects**

1. Created Exceptions:

@ResponseStatus(value = HttpStatus.*BAD\_REQUEST*)  
@Slf4j  
public class Sample404ExceptionClass extends RuntimeException {  
 public Sample404ExceptionClass(String msg) {  
 super(msg);  
 }  
}

public class GenericExceptionClass extends Exception{  
  
 public GenericExceptionClass(String msg){  
 super(msg);  
 }  
  
}

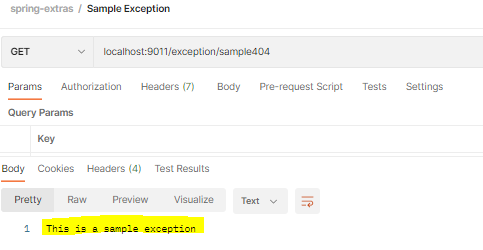
1. Created exception Handlers:

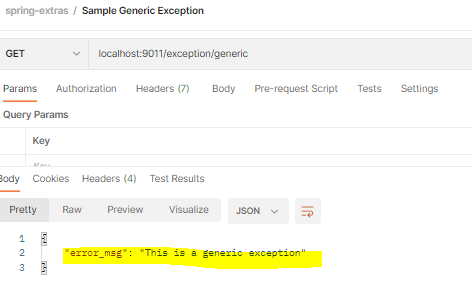
@ControllerAdvice  
public class AllExceptionHandler {  
  
 @ExceptionHandler(Sample404ExceptionClass.class)  
 @ResponseStatus(HttpStatus.*BAD\_REQUEST*)  
 public ResponseEntity<String> handleSample404Exception(Sample404ExceptionClass e) {  
 *log*.info("handling Sample404 Exception using Exception handler");  
 return ResponseEntity  
 .*status*(HttpStatus.*BAD\_REQUEST*)  
 .body(e.getMessage());  
 }  
  
 @ExceptionHandler(GenericExceptionClass.class)  
 @ResponseStatus(HttpStatus.*BAD\_REQUEST*)  
 public ResponseEntity<Map<String, String>> handleSampleGenericException(GenericExceptionClass e) {  
 *log*.info("handling Sample404 Exception using Exception handler");  
 return ResponseEntity  
 .*status*(HttpStatus.*BAD\_REQUEST*)  
 .body(new HashMap<>(){{  
 put("error\_msg", "This is a generic exception");  
 }});  
 }  
}

1. Methods throwing exception

@RestController  
@RequestMapping("/exception")  
public class ExceptionDemo {  
 Logger logger= LoggerFactory.*getLogger*(ExceptionDemo.class);  
   
 @GetMapping("/sample404")  
 public String throwException(){  
 logger.info("throwing an exception now");  
 throw new Sample404ExceptionClass("This is a sample exception");  
 }  
 @GetMapping("/generic")  
 public String throwGenericException() throws GenericExceptionClass {  
 logger.info("throwing generic exception now");  
 throw new GenericExceptionClass("This is a sample GENERIC exception");  
 }  
}

1. Exceptions thrown by application:





1. Logs:

