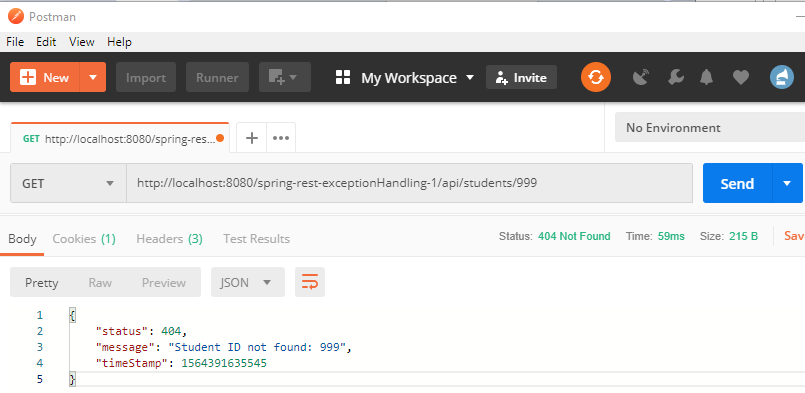
**61.3. Spring REST - Adding Generic Exception Handler**

**Generic Exception Handler**:

In previous example we create our own exception. When we try to find an invalid **student\_id** then we will get an JSON format error.



**Problem with this code**:

If we pass text as a **student\_id** then we will saw the ugly error message again.



WARNING: Failed to bind request element: org.springframework.web.method.annotation.MethodArgumentTypeMismatchException: Failed to convert value of type 'java.lang.String' to required type 'int'; nested exception is java.lang.NumberFormatException: For input string: "rrrrrrrr"

**Solve this problem**:

Now we modify our code to handle generic exception.

**Add code for generic exception in StudentRestController.java class**:

// add another exception handler ... to catch any exception (catch all)

@ExceptionHandler

**public** ResponseEntity<StudentErrorResponse> handleException(Exception exc) {

// create a StudentErrorResponse

StudentErrorResponse error = **new** StudentErrorResponse();

error.setStatus(HttpStatus.***BAD\_REQUEST***.value());

error.setMessage(exc.getMessage());

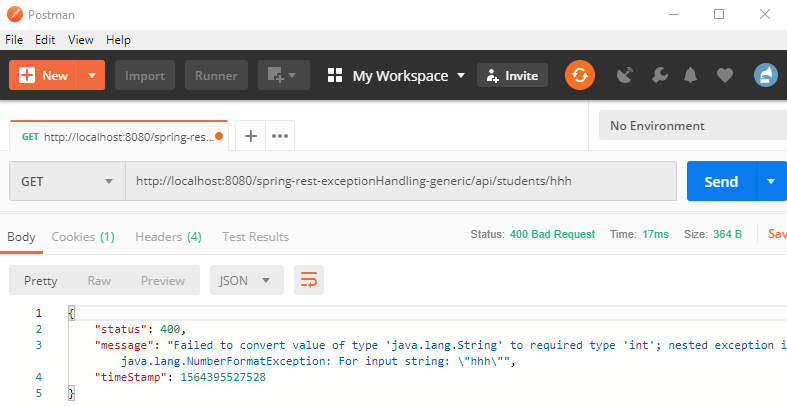
error.setTimeStamp(System.*currentTimeMillis*());

// return ResponseEntity

**return** **new** ResponseEntity<>(error, HttpStatus.***BAD\_REQUEST***);

}

Now if we pass text as student\_id we will get JSON error.



61.3. Spring REST - Adding Generic Exception Handler