

一、什么是统一异常处理

1、制造异常

除以0

```
1 int a = 10/0;
```

Response Body

```
{
  "timestamp": "2019-12-17T06:08:35.649+0000",
  "status": 500,
  "error": "Internal Server Error",
  "message": "/ by zero",
  "path": "/serviceedu/teacher"
}
```

2、什么是统一异常处理

我们想让异常结果也显示为统一的返回结果对象，并且统一处理系统的异常信息，那么需要统一异常处理

二、统一异常处理

1、创建统一异常处理器

在service_utils中创建统一异常处理类GlobalExceptionHandler.java:

```
1 /**
2  * 统一异常处理类
3  */
4 @ControllerAdvice
5 public class GlobalExceptionHandler {
6
```

```
7   @ExceptionHandler(Exception.class)
8   @ResponseBody
9   public R error(Exception e){
10      e.printStackTrace();
11      return R.error();
12  }
13 }
```

2、测试

返回统一错误结果

Response Body

```
{
  "success": false,
  "code": 20001,
  "message": "失败",
  "data": {}
}
```

三、处理特定异常

1、添加异常处理方法

GlobalExceptionHandler.java中添加

```
1 @ExceptionHandler(ArithmeticException.class)
2 @ResponseBody
3 public R error(ArithmeticException e){
4     e.printStackTrace();
5     return R.error().message("执行了特定异常");
6 }
```

2、测试

Response Body

```
{
  "success": false,
  "code": 20001,
  "message": "执行了特定异常",
  "data": {}
}
```

四、自定义异常

1、创建自定义异常类

```
1 @Data
2 @AllArgsConstructor
3 @NoArgsConstructor
4 public class YyghException extends RuntimeException {
5
6     @ApiModelProperty(value = "状态码")
7     private Integer code;
8
9     private String msg;
10
11 }
```

2、业务中需要的位置抛出GuliException

```
1 try {
2     int a = 10/0;
3 } catch (Exception e) {
4     throw new YyghException(20001, "出现自定义异常");
5 }
```

3、添加异常处理方法

GlobalExceptionHandler.java中添加

```
1 @ExceptionHandler(YyghException.class)
2 @ResponseBody
3 public R error(YyghException e){
4     e.printStackTrace();
5     return R.error().message(e.getMsg()).code(e.getCode());
6 }
```

4、测试

Response Body

```
{
  "success": false,
  "code": 20001,
  "message": "出现自定义异常",
  "data": {}
}
```