# 说明

版权归属：鑫学猿公司

微信：yuanhouduanyanfa

Java开发工程师转行辅导体系,保障拿到java工程师offer

图形用户界面, 文本, 应用程序

描述已自动生成

**Springboot全局异常统一处理**

**背景**

在以前的Java Web开发中，异常处理通常是通过try-catch语句块来实现的。这种方法在应用程序规模较小的情况下还可以，但是在大型应用中，可能存在大量的重复代码和不一致性问题。此外，当抛出未处理的异常时，用户会看到系统生成的默认错误页面，这对于用户体验是非常差的

**优点**

* 有助于保持代码整洁和模块化

如果没有全局异常处理，每个Controller方法都需要实现自己的异常处理，当应用程序变得越来越复杂时，这种代码会导致过多的冗余代码和混乱的异常处理逻辑。

* 提供用户体验

全局异常处理允许应用程序捕获未处理的异常并提供更友好的异常提示信息。这可以帮助用户更好地理解发生了什么错误，并且可以提高应用程序的可用性和用户满意度。

* 便于日志记录和监控：

全局异常处理可以帮助应用程序捕获和记录异常信息，在出现问题时快速定位问题。此外，它还可以与监控系统集成，以实时跟踪应用程序中出现的异常情况。

* 增强安全性

全局异常处理可以防止应用程序出现潜在的安全漏洞，例如SQL注入和XSS攻击。在异常处理过程中，应用程序可以检查输入并确保其不会被滥用。

**教程**

1. 自定义个异常类（当然，也可以不自定义，比如去统一针对NPE做处理也可以）

|  |
| --- |
| 需要依赖spring-boot-starter-web |

|  |
| --- |
| Java @Data public class BizException extends RuntimeException {  private String errorCode;   public BizException(String msg) {  super(msg);  }   public BizException(String errorCode, String msg) {  super(msg);  this.errorCode = errorCode;  }   public BizException(String errorCode, String msg, Throwable throwable) {  super(msg, throwable);  this.errorCode = errorCode;  }   public BizException(ErrorEnum errorEnum) {  super(errorEnum.getMessage());  this.errorCode = errorEnum.getCode();  } } |

1. 编写统一异常处理类，统一捕获处理返回

|  |
| --- |
| Java @RestControllerAdvice public class CommonExceptionHandler {   @ExceptionHandler(BizException.class)  public Result handleBizException(BizException e) {  //TODO 记录日志  System.out.println("记录exception");  return Result.Result(e.getErrorCode(), e.getMessage(), null);  } } |

1. 写测试例子

|  |
| --- |
| Java @Service public class UserServiceImpl implements UserService {  @Autowired  private UserMapper userMapper;   @Override  public void createUser(String name) {  userMapper.insertUser(name);  }   @Override  public UserDTO queryUserByName(String name) {  UserDO userDO = userMapper.findUserByName(name);  //不判空，我们看直接抛出异常的样子 // if (Objects.isNull(userDO)) { // return null; // }  //主动捕获异常  try {  return new UserDTO(userDO.getName(), userDO.getId().intValue());  } catch (Exception e) {  throw new BizException(ErrorEnum.DATA\_ERROR.getCode(), "根据姓名查询用户可能为空，请检查");  }  } } |

1. 结果

