**10-2 全局异常捕获器的设计**

**设计异常的接口定义**

|  |
| --- |
| Java public interface QiyuBaseError {   int getErrorCode();  String getErrorMsg(); } |

**将常用的异常类型实现给定义出来**

|  |
| --- |
| Java package org.qiyu.live.web.starter.error;  /\*\*  \* @Author idea  \* @Date: Created in 11:25 2023/8/2  \* @Description  \*/ public enum BizBaseErrorEnum implements QiyuBaseError{   PARAM\_ERROR(100001,"参数异常"),  TOKEN\_ERROR(100002,"用户token异常");   private int errorCode;  private String errorMsg;   BizBaseErrorEnum(int errorCode, String errorMsg) {  this.errorCode = errorCode;  this.errorMsg = errorMsg;  }   @Override  public int getErrorCode() {  return 0;  }   @Override  public String getErrorMsg() {  return null;  } } |

**自定义断言组件**

|  |
| --- |
| Java package org.qiyu.live.web.starter.error;   /\*\*  \* @Author idea  \* @Date: Created in 11:18 2023/8/2  \* @Description  \*/ public class ErrorAssert {    /\*\*  \* 判断参数不能为空  \*  \* @param obj  \* @param qiyuBaseError  \*/  public static void isNotNull(Object obj, QiyuBaseError qiyuBaseError) {  if (obj == null) {  throw new QiyuErrorException(qiyuBaseError);  }  }   /\*\*  \* 判断字符串不能为空  \*  \* @param str  \* @param qiyuBaseError  \*/  public static void isBlank(String str, QiyuBaseError qiyuBaseError) {  if (str == null || str.trim().length() == 0) {  throw new QiyuErrorException(qiyuBaseError);  }  }   /\*\*  \* flag == true  \*  \* @param flag  \* @param qiyuBaseError  \*/  public static void isTure(boolean flag, QiyuBaseError qiyuBaseError) {  if (!flag) {  throw new QiyuErrorException(qiyuBaseError);  }  } } |

**全局异常捕获器**

|  |
| --- |
| Java package org.qiyu.live.web.starter.error;  import jakarta.servlet.http.HttpServletRequest; import org.qiyu.live.common.interfaces.vo.WebResponseVO; import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.springframework.web.bind.annotation.ControllerAdvice; import org.springframework.web.bind.annotation.ExceptionHandler; import org.springframework.web.bind.annotation.ResponseBody;  /\*\*  \* @Author idea  \* @Date: Created in 11:11 2023/8/2  \* @Description  \*/ @ControllerAdvice public class GlobalExceptionHandler {   private static final Logger LOGGER = LoggerFactory.getLogger(GlobalExceptionHandler.class);   @ExceptionHandler(value = Exception.class)  @ResponseBody  public WebResponseVO errorHandler(HttpServletRequest request, Exception e) {  LOGGER.error(request.getRequestURI() + ",error is ", e);  return WebResponseVO.sysError("系统异常");  }    @ExceptionHandler(value = QiyuErrorException.class)  @ResponseBody  public WebResponseVO sysErrorHandler(HttpServletRequest request, QiyuErrorException e) {  //业务异常，参数传递有误,都会走到这里  LOGGER.error(request.getRequestURI() + ",error code is {},error msg is {}", e.getErrorCode(), e.getErrorMsg());  return WebResponseVO.bizError(e.getErrorCode(), e.getErrorMsg());  } } |

|  |
| --- |
| Java public static WebResponseVO sysError(String msg) {  WebResponseVO webResponseVO = new WebResponseVO();  webResponseVO.setCode(500);  webResponseVO.setMsg(msg);  return webResponseVO; } |