

springMVC统一异常处理

```
1 com.msb.framework.web.exception.GlobalExceptionHandler
```

全局捕获异常

```
1 @ResponseBody
2 @ExceptionHandler(Exception.class)
3 public ResultWrapper<Object> exception(Exception exception) {
4     log.error("未知全局捕获", exception);
5     return ResultWrapper.fail(BaseResultCodeEnum.SYSTEM_ERROR);
6 }
7
8 @ResponseBody
9 @ExceptionHandler(RpcException.class)
10 public ResultWrapper<Object> rpcException(RpcException rpcException) {
11     log.error("dubbo调用异常", rpcException);
12     return ResultWrapper.fail(BaseResultCodeEnum.SYSTEM_ERROR);
13 }
14
15 @ResponseBody
16 @ExceptionHandler(BizException.class)
17 public ResultWrapper<Object> bizException(BizException bizException) {
18     log.warn("业务异常 {}", bizException.getMessage());
19     return ResultWrapper.fail(bizException.getCode(), bizException.getMs
20 }
21
22 @ResponseBody
23 @ExceptionHandler(MethodArgumentNotValidException.class)
24 public ResultWrapper<Object> methodArgumentNotValidException(MethodArgum
25     ....
26     return ResultWrapper.fail(BaseResultCodeEnum.ILLEGAL_ARGUMENT.getCod
27 }
```

包含

1, Exception

程序未知异常捕获，统一返回系统异常，打印error日志

2, RpcException

统一调用Dubbo接口时出现的异常会被统一捕获，返回系统异常，在调用dubbo接口时如需可以自己捕获Rpc异常

3, BizException

统一业务异常处理，打印warn日志，响应到前端的则是具体的业务异常信息

4, MethodArgumentNotValidException

统一参数校验异常，使用@NotNull时则会统一处理

当需要自定义全局异常捕获时，可根据需要全局异常捕获类，并设置@Order注解设置为ExceptionHandlerConstant.EXCEPTION_ORDER_HIGH，为覆盖以框架内部的全局异常捕获

Dubbo异常统一处理

```
1 com.msb.framework.web.dubbo.DubboExceptionHandler
```

```
1 public Result invoke(Invoker<?> invoker, Invocation invocation) throws RpcException {
2     Result invoke;
3     try {
4         invoke = invoker.invoke(invocation);
5         if (invoke != null) {
6             if (invoke.getException() != null) {
7                 Throwable exception = invoke.getException();
8                 String errorInfo = "Got unchecked and undeclared exception when calling the service: " + invoker.getInterface().getName() + ", exception: " + exception.getClass().getName() + ": " + exception.getMessage();
9                 throw new RpcException(errorInfo);
10            }
11        }
12    } catch (RpcException e) {
13        throw e;
14    } catch (Throwable e) {
15        log.error("dubbo rpc error" + RpcContext.getContext().getRemoteHost() + ": " + e.getMessage());
16        throw e;
17    }
18    return invoke;
19 }
```

统一dubbo异常处理，在调用dubbo接口时，如上层抛出异常，统一包装为RpcException，内容包含了dubbo生产者的业务异常信息，调用方可以在已知情况下，捕获rpc异常

rpc异常会把被调用方信息拼接成RpcException，包括ip 实例，接口，方法等信息

