1 通过注解 对所有json格式返回的异常返回页面做处理

```
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ResponseBody;

@ControllerAdvice
public class GlobalExceptionHandler {

        @ExceptionHandler(Exception.class)
        @ResponseBody
    public String handleException(Exception e) {
        e.printStackTrace();
        return "exception msg : " + e.getMessage();
    }
}
```

2 自定义一个异常处理类 注册到 spring bean中

xml文件

```
<bean id="myExceptionHandler"
class="com.huaao.micropolice.root.commons.exception.MyExceptionHandler
"/>
```

对应的Java异常处理类

```
public class MyExceptionHandler implements HandlerExceptionResolver{

        @Override
        public ModelAndView resolveException(HttpServletRequest request,
                        HttpServletResponse response, Object handler, Exception ex) {
```

```
        if (!(request.getHeader("accept").contains("application/json") ||
(request
                .getHeader("X-Requested-With") != null && request
                .getHeader("X-Requested-
With").contains("XMLHttpRequest")))) {
            ex.printStackTrace();
            System.out.println("JSP格式返回");
            return new ModelAndView("/exception/500");
        } else {// JSON格式返回
            try {
                PrintWriter writer = response.getWriter();
                Map<String, Object> map = new HashMap<String,
Object>();
                map.put(BaseController.CODE, 500);
                map.put(BaseController.MESSAGE,
BaseController.MESSAGE_FAIL);
                map.put("errorMessage", ex.getMessage());
                String  json = JsonUtil.toJson(map);
                writer.write(json);
                writer.flush();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }
    }

}
```