**ADC-DA框架编码规范**

**中国汽车技术研究中心**

**数据资源中心**

**2018年3月**

**目录**

[第一章 前言 3](#_Toc508089709)

[1.1 编写目的: 3](#_Toc508089710)

[第二章 包命名规范 4](#_Toc508089711)

[第三章 Controller层返回值封装 6](#_Toc508089712)

[第四章 添加Filter 7](#_Toc508089713)

[第五章 异常处理方式 9](#_Toc508089714)

[第六章 前后端消息传递方式 10](#_Toc508089715)

[第七章 国际化 11](#_Toc508089716)

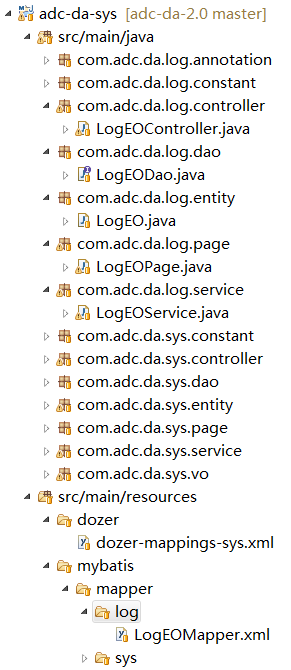
# 前言

## 编写目的:

统一使用ADC-DA框架进行项目研发时的规范

# 包命名规范

ADC-DA框架每一个模块一般都有Controller,Service,Dao,Entity,Page,Dozer,Vo以及Mapper层



包命名为com.adc.da.{模块名}.\*，Entity实体类应以EO结尾，Controller,Service,Dao,Page应以类似EOController,EOService,EODao,EOPage结尾。

Vo类以VO结尾，Vo层主要为与前台交互的类，EO层为后台和数据库对应的实体类，VO与EO的转换通过Dozer，EO与VO转换规则xml文件置于src/main/resources/dozer

mybatis sql配置文件应放到src/main/resources/mybatis/mapper/模块名 下，命名以EOMapper结尾。

Dozer的使用方法：在Controller层注入com.adc.da.util.utils.BeanMapper类，调用其*map, mapList, mapPage*等方法

*@RestController*

*@RequestMapping("/${restPath}/sys/user")*

*@Api(description = "用户管理")*

*public class UserEOController extends BaseController<UserEO> {*

*@Autowired*

*BeanMapper beanMapper;*

*@ApiOperation(value = "|UserEO|详情")*

*@GetMapping("/{id}")*

*@RequiresPermissions("sys:user:get")*

*public ResponseMessage<UserVO> getById(@NotNull @PathVariable("id") String id) throws Exception {*

*UserVO userVO =* ***beanMapper.map****(userEOService.getUserWithRoles(id), UserVO.class);*

*return Result.success(userVO);*

*}*

# Controller层返回值封装

ADC-DA框架要求对Controller层返回数据进行封装以便与前台程序良好交互，对Controller层返回数据套上ResponseMessage类即可。

@ApiOperation(value = "|UserEO|详情")

@GetMapping("/{id}")

@RequiresPermissions("sys:user:get")

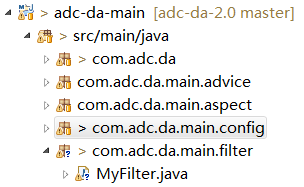
**public** ResponseMessage<UserVO> getById(@NotNull @PathVariable("id") String id) **throws** Exception {

UserVO userVO = beanMapper.map(userEOService.getUserWithRoles(id), UserVO.**class**);

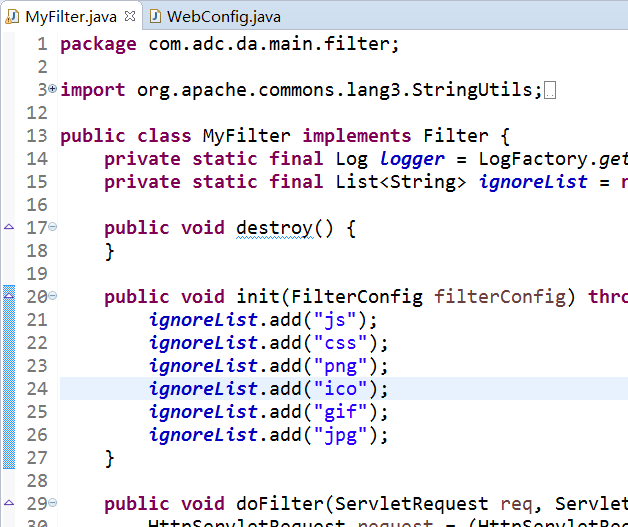
**return** Result.*success*(userVO); }

# 添加Filter

ADC-DA框架的工具类都在adc-da-util模块src/main/java/com/adc/da/util/filter路径下，由于adc-da-util是单独打成jar包的模块，所以如果要自定义Filter，可以在adc-da-main主模块src/main/java/com/adc/da/main/filter路径下定义：



比如定义一个MyFilter类:



之后还要在com.adc.da.main.config. WebConfig类中注册自定义Filter：



# 异常处理方式

ADC-DA框架要求统一在Controller层抛出异常，如下：

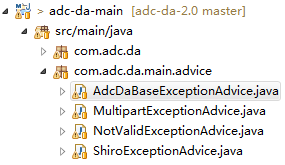
@ApiOperation(value = "|UserEO|新增")

@PostMapping(consumes = ***APPLICATION\_JSON\_UTF8\_VALUE***)

@RequiresPermissions("sys:user:save")

**public** ResponseMessage<UserVO> create(@RequestBody UserVO userVO) **throws** Exception {

异常处理类在主模块adc-da-main的com.adc.da.main.advice路径下：



开发可以抛出AdcDaBaseException也可以自定义异常，自定义异常可以如下，在抛出异常时返回自定义的error code和error message

@ControllerAdvice

@Order(value=2)

public class AdcDaBaseExceptionAdvice {

private static final Logger logger = LoggerFactory.getLogger(AdcDaBaseExceptionAdvice.class);

@ResponseStatus(HttpStatus.OK)

@ExceptionHandler(AdcDaBaseException.class)

@ResponseBody

public ResponseMessage handlerAdcDaBaseException(AdcDaBaseException exception) {

logger.warn(exception.getMessage(), exception);

return Result.error(exception.getErrorCode(), exception.getMessage());

}

# 前后端消息传递方式

前台调用后台如果后台Controller参数是对象的话，以json格式调用，参数前加

后: 当传输方式为POST时通过param拼装参数传输到后台，后台返回json格式字符串，如果传输成功通过success中data接受返回内容。如果链接失败通过error中data返回失败信息。如传输方式为GET通过url拼装传输到后台。前台接收返回内容方式相同。

@RequestBody

*public ResponseMessage<PageInfo<OrderEO>> page(@RequestBody OrderEOPage page)*

前台:

*var param = {'page':pageNo,'pageSize':pageSize};*

param ***= JSON***.stringify(params);

***$***.ajax({  
 **url**:***url***,  
 **type**:**type**,  
 **data**: param,  
 **contentType**:**"application/json; charset=UTF-8"**,  
 success:**function** (data) {  
 **layer**.msg(msg+**'成功！'**);  
 },  
 error:**function** (data) {  
 **layer**.msg(msg+**'失败！'**);  
 }  
});

# 国际化

要使用国际化，首先后台在返回给前台的值时要返回一个自定义code码，

***return*** *Result.success("r0003", userVO)*

***return*** *Result.error("r0004", "mistake")*

前台

*jQuery.****i18n****.properties({* ***name****:****'Messages'****,* ***path****:****'bundle/'****,* ***mode****:****'both'****,* ***language****: lan ,//'pt\_BR',* ***async****:* ***true****,  
 callback:* ***function****() {  
 jQuery.****i18n****.prop(****'msg\_hello'****);  
 jQuery.****i18n****.prop(****'msg\_complex'****,* ***'John'****);  
 }  
});*

通过 msg读取内容。