

## 12、异常处理

笔记本: spring  
创建时间: 2022/4/4 17:24  
作者: 雷丰阳  
URL: http://localhost:8080/11.SpringMVC\_i18n/handle03

默认就是这个几个HandlerExceptionResolver

ExceptionHandlerExceptionResolver

ResponseStatusExceptionHandler

DefaultHandlerExceptionResolver

如果异常解析器都不能处理就直接抛出去;

```
private void processDispatchResult(HttpServletRequest request, HttpServletResponse response,
    HandlerExecutionChain mappedHandler, ModelAndView mv, Exception exception) throws Exception {

    boolean errorView = false;

    if (exception != null) { //如果有异常
        if (exception instanceof ModelAndViewDefiningException) {
            logger.debug("ModelAndViewDefiningException encountered", exception);
            mv = ((ModelAndViewDefiningException) exception).getModelAndView();
        }
        else { //处理异常
            Object handler = (mappedHandler != null ? mappedHandler.getHandler() : null);
            mv = processHandlerException(request, response, handler, exception);
            errorView = (mv != null);
        }
    }

    // Did the handler return a view to render?
    if (mv != null && !mv.wasCleared()) {
        //来到页面
        render(mv, request, response);
        if (errorView) {
            WebUtils.clearErrorRequestAttributes(request);
        }
    }
    else {
        if (logger.isDebugEnabled()) {
            logger.debug("Null ModelAndView returned to DispatcherServlet with name '" + getServletName() +
                "': assuming HandlerAdapter completed request handling");
        }
    }

    if (WebAsyncUtils.getAsyncManager(request).isConcurrentHandlingStarted()) {
        // Concurrent handling started during a forward
        return;
    }

    if (mappedHandler != null) {
        mappedHandler.triggerAfterCompletion(request, response, null);
    }
}
```

所以异常解析器尝试解析，解析完成进行后续，解析失败下一个解析器

```
protected ModelAndView processHandlerException(HttpServletRequest request, HttpServletResponse response,
    Object handler, Exception ex) throws Exception {

    // Check registered HandlerExceptionResolvers...
    ModelAndView exMv = null;
    for (HandlerExceptionResolver handlerExceptionResolver : this.handlerExceptionResolvers) {
        exMv = handlerExceptionResolver.resolveException(request, response, handler, ex);
        if (exMv != null) {
            break;
        }
    }
    if (exMv != null) {
        if (exMv.isEmpty()) {
            return null;
        }
        // We might still need view name translation for a plain error model...
        if (!exMv.hasView()) {
            exMv.setViewName(getDefaultViewName(request));
        }
        if (logger.isDebugEnabled()) {
            logger.debug("Handler execution resulted in exception - forwarding to resolved error view: " + exMv, ex);
        }
        WebUtils.exposeErrorRequestAttributes(request, ex, getServletName());
        return exMv;
    }

    throw ex;
}
```

ExceptionHandlerExceptionHandler:ExceptionHandler  
ResponseStatusExceptionHandler:ResponseStatus;  
给自定义异常上标注的

```
@ResponseStatus(reason="用户被拒绝登陆",value=HttpStatus.NOT_ACCEPTABLE)
public class UsernameNotFoundException extends RuntimeException {

    private static final long serialVersionUID = 1L;

}
```

## HTTP Status 406 - 用户被拒绝登陆

**type** Status report

**message** 用户被拒绝登陆

**description** The resource identified by this request is only capable of generating responses with characteristics not acceptable according to the request "accept" headers.

Apache Tomcat/7.0.59

DefaultHandlerExceptionHandler:判断是否SpringMVC自带的异常  
Spring自己的异常: 如: HttpRequestMethodNotSupportedException。如果没人处理

## HTTP Status 405 - Request method 'GET' not supported

**type** Status report

**message** Request method 'GET' not supported

**description** The specified HTTP method is not allowed for the requested resource.

```
... this.handlerExceptionResolvers,
re v handlerExceptionResolvers= ArrayList<E> (id=104)
    v elementData= Object[3] (id=110)
        > [0]= ExceptionHandlerExceptionHandler (id=119)
        > [1]= ResponseStatusExceptionHandler (id=120)
        > [2]= DefaultHandlerExceptionHandler (id=121)
    modCount= 1
```

默认的异常们?

```
try {
    if (ex instanceof NoSuchRequestHandlingMethodException) {
        return handleNoSuchRequestHandlingMethod((NoSuchRequestHandlingMethodException) ex, request, response,
            handler);
    }
    else if (ex instanceof HttpRequestMethodNotSupportedException) {
        return handleHttpRequestMethodNotSupported((HttpRequestMethodNotSupportedException) ex, request,
            response, handler);
    }
    else if (ex instanceof HttpMediaTypeNotSupportedException) {
        return handleHttpMediaTypeNotSupported((HttpMediaTypeNotSupportedException) ex, request, response,
            handler);
    }
    else if (ex instanceof HttpMediaTypeNotAcceptableException) {
        return handleHttpMediaTypeNotAcceptable((HttpMediaTypeNotAcceptableException) ex, request, response,
            handler);
    }
    else if (ex instanceof MissingServletRequestParameterException) {
        return handleMissingServletRequestParameter((MissingServletRequestParameterException) ex, request,
            response, handler);
    }
    else if (ex instanceof ServletRequestBindingException) {
        return handleServletRequestBindingException((ServletRequestBindingException) ex, request, response,
            handler);
    }
    else if (ex instanceof ConversionNotSupportedException) {
        return handleConversionNotSupported((ConversionNotSupportedException) ex, request, response, handler);
    }
    else if (ex instanceof TypeMismatchException) {
        return handleTypeMismatch((TypeMismatchException) ex, request, response, handler);
    }
    else if (ex instanceof HttpMessageNotReadableException) {
        return handleHttpMessageNotReadable((HttpMessageNotReadableException) ex, request, response, handler);
    }
    else if (ex instanceof HttpMessageNotWritableException) {
        return handleHttpMessageNotWritable((HttpMessageNotWritableException) ex, request, response, handler);
    }
    else if (ex instanceof MethodArgumentNotValidException) {
        return handleMethodArgumentNotValidException((MethodArgumentNotValidException) ex, request, response,
            handler);
    }
    else if (ex instanceof MissingServletRequestPartException) {
```

```

        return handleMissingServletRequestPartException(((MissingServletRequestPartException) ex, request, response,
handler));
    }
    else if (ex instanceof BindException) {
        return handleBindException((BindException) ex, request, response, handler);
    }
    else if (ex instanceof NoHandlerFoundException) {
        return handleNoHandlerFoundException((NoHandlerFoundException) ex, request, response, handler);
    }
}
catch (ExceptionHandlerException) {
    logger.warn("Handling of [" + ex.getClass().getName() + "] resulted in Exception", handlerException);
}
return null;
}

```

HandlerExceptionResolver - org.springframework.web.servlet

- AbstractHandlerExceptionResolver - org.springframework.web.servlet.handler
  - AbstractHandlerMethodExceptionResolver - org.springframework.web.servlet.mvc.method.annotation
    - ExceptionHandlerExceptionResolver - org.springframework.web.servlet.mvc.method.annotation
    - AnnotationMethodHandlerExceptionResolver - org.springframework.web.servlet.mvc.annotation
    - DefaultHandlerExceptionResolver - org.springframework.web.servlet.mvc.support
    - ResponseStatusExceptionHandlerResolver - org.springframework.web.servlet.mvc.annotation
    - SimpleMappingExceptionHandlerResolver - org.springframework.web.servlet.handler

SimpleMappingExceptionHandlerResolver:

通过配置的方式进行异常处理

handlerExceptionResolvers = ArrayList<E> (id=78)

- elementData = Object[4] (id=88)
  - [0] = ExceptionHandlerExceptionResolver (id=98)
  - [1] = ResponseStatusExceptionHandlerResolver (id=99)
  - [2] = DefaultHandlerExceptionResolver (id=100)
  - [3] = SimpleMappingExceptionHandlerResolver (id=101)

modCount = 1

<bean

class="org.springframework.web.servlet.handler.SimpleMappingExceptionHandlerResolver">

<!-- exceptionMappings: 配置哪些异常去哪些页面 -->

<property name="exceptionMappings">

<props>

<!-- key: 异常全类名; value: 要去的页面视图名; -->

<prop

key="java.lang.NullPointerException">myerror</prop>

</props>

</property>

<!--指定错误信息取出时使用的key -->

<property name="exceptionAttribute" value="ex"></property>

</bean>