

后端开发框架 集成开发手册

编写： 黄新房

审核：

批准：



NUCTECH COMPANY LIMITED

©Copyright 2016 NUCTECH COMPANY LIMITED All Rights Reserved

目录

目录.....	2
阅读指南.....	3
1. 功能介绍	4
1.1 限流功能	4
1.2 模板功能	4
1.3 注解功能	4
1.4 通用异常处理功能	5
1.5 分页查询功能	5
1.6 文件日志功能	5
1.7 审计日志功能	5
1.8 访问日志功能	5
1.9 配置加密功能	6
1.10 对象转换功能	6
1.11 核心常量包	7
1.12 自定义 Bean Name 功能	7
1.13 通用对象返回	7
1.14 分布式自增 ID 功能	8
1.15 XSS	8
1.16 Validator 分组	8
1.17 数据校验功能	8
1.18 核心 Controller、Service、Dao、DaoImpl	9
1.19 核心业务 Controller、Service、Dao、DaoImpl	9
1.20 增删改查功能	10
1.21 审计日志接口	10
2. 开发步骤（以 UPMS 为例）	13
2.1 包结构必须满足如下要求：	13
2.2 创建 Maven 工程	13
2.3 pom.xml 结构为：	16

阅读指南

【手册目标】

通过阅读本手册，用户能够对后端开发框架功能有全面的了解,并在日常工作中正确使用该框架。

【阅读对象】

本手册适用于研发人员，它提供了关于后端开发框架详细使用说明。

1. 功能介绍

1.1 限流功能

注解: @QPSAccessRateLimiter

使用:

```
<!-- 请求速率限制拦截器 -->
```

```
<bean class="com.nuctech.platform.common.Limit.RateLimiterInterceptor"><!-- IP流量限制 and 注解QPS限制, IP流量限制优先级最高 -->
```

```
    <constructor-arg name="threshold" value="10" type="java.lang.Integer" /><!-- 同一个IP每分钟最大访问量 -->
```

```
    <constructor-arg name="maximumSize" value="1024" type="java.lang.Long" /><!-- 同时最大可访问IP数量 -->
```

```
    <constructor-arg name="enable" value="false" type="java.lang.Boolean" /><!-- false/true, true为启用全局ip限制 -->
```

```
</bean>
```

1.2 模板功能

使用:

```
<bean class="com.nuctech.platform.common.template.thymeleaf.StringTemplateImpl" />
```

1.3 注解功能

支持的注解:

EnumPattern

使用:

```
@EnumPattern(abbrPrefix = "", abbrPrefixKey = "", nameKey = "", uppercase = false, prefixKey = "")
```

EnumPrefix

使用:

```
@Target({ ElementType.TYPE })
```

```
@Retention(RetentionPolicy.RUNTIME)
```

```
@EnumPrefix(Constants.ENUMPREFIX)
```

```
public @interface UpmsEnumPrefix {
```

```
}
```

```
@Getter
```

```
@UpmsEnumPrefix
```

```
public enum EnumRoleRsv implements EnumInterface {
```

GetJsonMapping

GetXmlMapping

JsonMapping

PostJsonMapping

PostXmlMapping

RequestHeader

XmlMapping

使用:

```
@JsonMapping(method = { RequestMethod.GET, RequestMethod.POST }, value = "/add")
```

1.4 通用异常处理功能

回滚异常ErrorCodeException

非回滚异常ErrorCodeNoRollBackException

使用: (回滚异常和非回滚异常使用相同)

```
orgDao.findById(pId).orElseThrow(() -> new  
ErrorCodeException(InfoErrorCode.ORG_PARENT_NOT_EXISTED));
```

```
throw new ErrorCodeException(InfoErrorCode.ORG_HAS_CHILD_PROHIBIT_DELETION);
```

1.5 分页查询功能

分页查询, 将自动将页码超过最后一页的数据定位到最后一页

1.6 文件日志功能

使用:

```
<!-- 操作日志拦截器 -->
```

```
<bean class="com.nuctech.platform.common.web.interceptor.OperateLogInterceptor" />
```

1.7 审计日志功能

使用:

```
<!-- 日志参数配置 -->
```

```
<bean class="com.nuctech.log.constant.SystemBaseConfig" p:workerId="${workerId}"/>
```

```
<!-- 配置字符串模板实现类 -->
```

```
<bean class="com.nuctech.platform.common.template.thymeleaf.StringTemplateImpl" />
```

```
<!-- 审计日志, 详细日志监听 -->
```

```
<bean class="com.nuctech.log.listener.HibernateEventListener"></bean>
```

```
@UpmsSysAuditLog(action = AuditLogActionConstants.ADD,  
operateContent = "Add Application(code:[*{appDto.code}]], name:
```

```
[*{appDto.name}]]]",
```

```
operateObject = AuditLogConstants.APPOBJECT)
```

```
@OperateLog(operateType = OperateType.POST, operateContent = "Add App")
```

```
@ResponseBody
```

```
@JsonMapping(method = { RequestMethod.GET, RequestMethod.POST }, value = "/add")
```

```
public ReturnResult addApp(HttpServletRequest request, HttpServletResponse response,  
Model model, @Validated(value = { Create.class }) AppDto appDto, BindingResult  
bindingResult) {  
}
```

1.8 访问日志功能

使用:

```
<dependency>
```

```
<groupId>com.nuctech.biz</groupId>
```

```
<artifactId>biz.accesslog</artifactId>
```

```
<version>1.1.0</version>
```

```
</dependency>
```

```
SysAccessLog sysAccessLog = new SysAccessLog();
```

```
long sysAccessLogId = idService.getId();
```

```
LogServletUtil.setSysAccessLogId(sysAccessLogId);
```

```
sysAccessLog.setId(sysAccessLogId).setOperationTime(new Date());
```

```
sysAccessLog.setAction(Action.LOGIN);
```

```
sysAccessLog.setClientIp(getClientIp());
```

```
sysAccessLog.setOperateAccount(userDto.getAccount());
```

```
sysAccessLogService.save(sysAccessLog);
```

1.9 配置加密功能

使用:

```
<bean id="propertyConfigurer"
      class="com.nuctech.platform.common.util.EncryptPropertyPlaceholderConfigurer">
  <property name="propertyNames">
    <array value-type="java.lang.String"><!--加密的字段-->
      <value>db.username</value>
      <value>db.password</value>
    </array>
  </property>
  <property name="Locations">
    <list>
      <value>classpath:etc/sys/SysConfig.properties</value>
      <value>classpath:etc/db/jdbc_psql.properties</value>
      <!-- <value>file:${catalina.base}/jdbc_psql.properties</value> -->
    </list>
  </property>
</bean>
```

1.10 对象转换功能

使用:

```
<!-- 自定义参数绑定 -->
<bean id="conversionService"
      class="org.springframework.format.support.FormattingConversionServiceFactoryBean">
  <!-- 请求转换器 -->
  <property name="converters">
    <set>
      <!-- 日期类型转换 -->
      <bean class="com.nuctech.platform.common.web.convert.DateConverter" />
    </set>
  </property>
</bean>

<mvc:message-converters register-defaults="false">
  <bean class="org.springframework.http.converter.StringHttpMessageConverter">
    <property name="supportedMediaTypes">
      <list>
        <value>application/json;charset=UTF-8</value>
      </list>
    </property>
  </bean>
  <bean id="mappingJackson2XmlHttpMessageConverter"
        class="org.springframework.http.converter.xml.MappingJackson2XmlHttpMessageConverter">
    <property name="supportedMediaTypes">
      <list>
        <value>application/xml</value>
      </list>
    </property>
    <property name="objectMapper">
      <bean class="com.nuctech.platform.common.web.databind.XmlMapper" />
    </property>
  </bean>
</mvc:message-converters>
```

```

        </bean>
        <bean id="mappingJackson2HttpMessageConverter"
            class="org.springframework.http.converter.json.MappingJackson2HttpMessageConve
rter">
            <property name="supportedMediaTypes">
                <list>
                    <value>application/json;charset=UTF-8</value>
                </list>
            </property>
            <property name="objectMapper">
                <bean
class="com.nuctech.platform.common.web.databind.ObjectMapper" />
            </property>
        </bean>
</mvc:message-converters>

```

1.11 核心常量包

集合：

```

/**
 * 查询关键字，根据类型匹配
 * 支持in/notin
 *
 * @author huangxinfang
 */
public enum EnumCollect {
    IN, NOTIN,
}

```

非集合：

```

/**
 * 查询关键字，根据类型匹配
 * 支持eq/noteq/gt/lt
 *
 * @author huangxinfang
 */
public enum EnumQuery {
    EQ, NOTEQ, GT, LT,
}

```

1.12 自定义Bean Name功能

使用：

```

<context:component-scan base-package="com.nuctech" name-
generator="com.nuctech.platform.common.web.generator.AnnotationBeanFullNameGenerator
">
    <context:exclude-filter type="annotation"
        expression="org.springframework.stereotype.Controller" />
</context:component-scan>

```

1.13 通用对象返回

```

/**
 * 通用返回类型
 */
@Setter
@Getter
@NoArgsConstructor

```

```

@Accessors(chain = true)
public class ReturnResult {

    /**
     * 返回成功失败标记
     */
    private Boolean flag = true; // 默认成功

    /**
     * 返回提示信息
     */
    private String message;

    /**
     * 返回错误编码信息
     */
    private String errorCode;

    /**
     * 返回数据
     */
    private Object result;
}

```

1.14 分布式自增ID功能

类JSnowFlake 和 IdService

使用：（内置在IdService中）

```

public long getId() {
    if (jSnowFlake == null) {
        jSnowFlake = new JSnowFlake(systemBaseConfig.getWorkerId());
    }
    return jSnowFlake.nextId();
}

```

1.15 XSS

```

<filter>
    <filter-name>xssFilter</filter-name>
    <filter-
class>com.nuctech.platform.common.web.xss.XssHttpServletFilter</filter-class>
        <async-supported>true</async-supported>
</filter>
<filter-mapping>
    <filter-name>xssFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

1.16 Validator 分组

Create
Delete
Retrieve
Update

1.17 数据校验功能

使用：

使用@Validated或@Valid，参数后紧跟BindingResult

```

public ReturnResult updateAppOrgs(HttpServletRequest request, HttpServletResponse

```



```
response, Model model,
        @Validated(value = { IdRequired.class }) OrgDto dto, BindingResult
        errors) {
}
```

1.18 核心Controller、Service、Dao、DaoImpl

CommonController、CommonService、CommonDao、CommonDaoImpl

1.19 核心业务Controller、Service、Dao、DaoImpl

BizController、BizService、BizDao、BizDaoImpl

核心业务具备的业务相关方法：

```
/**
```

```
 * @return 用户 id
```

```
 */
```

```
public String getLoginUserId()
```

```
/**
```

```
 * @return 用户账号
```

```
 */
```

```
public String getLoginAccount()
```

```
/**
```

```
 * @return 用户名
```

```
 */
```

```
public String getLoginUserName()
```

```
/**
```

```
 * @return 机构 id
```

```
 */
```

```
public String getLoginOrgId()
```

```
/**
```

```
 * @return 客户端 ip
```

```
 */
```

```
public String getClientIp()
```

```
/**
```

```
 * @return 应用签名
```

```
 */
```

```
public String getAppToken()
```

```
/**
```

```
 * @return 跟踪号
```

```
 */
```

```
public String getTraceId()
```

```
/**
```

```
 *
```

```

    * @return 审计日志 id
    */
public Long getSysAuditLogId()
/**
 *
 * @return 登录时间
 */
public Date getLoginTime()

```

1.20 增删改查功能

1.21 审计日志接口

审计日志分页查询接口

请求

请求网址

log/auditlog/getPages

请求方法

post

Content-Type

application/x-www-form-urlencoded; charset=UTF-8

请求参数

参数	类型	说明	是否必填
operationTimeStart	string	操作开始时间, 毫秒字符串	否
operationTimeEnd	string	操作结束时间, 毫秒字符串	否
clientIp	string	客户端ip地址	否
operateAccount	string	操作员账号	否
operateObject	string	操作对象	否
action	string	操作	否
result	string	操作结果(FAILURE/SUCCESS)	否

响应

Content-Type

application/json; charset=UTF-8

返回字段说明

参数	说明
id	审计日志id
operationTime	操作时间
clientIp	客户端ip
operateId	操作员ID
operateAccount	操作员账号
operateObject	操作对象
action	操作
result	操作结果

参数	说明
reasonCode	失败原因代码
operateContent	操作内容

返回样例

```
{
  "flag": true,
  "message": null,
  "errorCode": null,
  "result": {
    "content": [
      {
        "id": "330442730842886144",
        "operationTime": 1537338725385,
        "operateId": "2",
        "operateAccount": "admin",
        "operateObject": "323",
        "action": "",
        "result": "upms.op.success",
        "operateContent": ""
      }
    ],
    "pageable": {
      "sort": {
        "sorted": true,
        "unsorted": false
      },
      "offset": 0,
      "pageSize": 10,
      "pageNumber": 0,
      "unpaged": false,
      "paged": true
    },
    "last": true,
    "totalPages": 1,
    "totalElements": 1,
    "number": 0,
    "size": 10,
    "sort": {
      "sorted": true,
      "unsorted": false
    },
    "numberOfElements": 1,
    "first": true
  }
}
```

2. 审计日志详情

请求

请求网址

log/auditlog/detail/getPages

请求方法

post

Content-Type

application/x-www-form-urlencoded; charset=UTF-8

请求参数

参数	类型	说明	是否必填
auditLogId	long	审计日志id	是

响应

Content-Type

application/json; charset=UTF-8

返回字段说明

参数	说明
id	审计日志详情id
operationTime	操作时间
fieldName	数据项
oldValue	操作前内容
newValue	操作后内容

3. 审计日志Action 国际化key列表获取

请求

请求网址

log/auditlog/key/getActionKeys

请求方法

post

Content-Type

application/x-www-form-urlencoded; charset=UTF-8

请求参数

参数	类型	说明	是否必填
appCode	string	应用编码	否

响应

Content-Type

application/json; charset=UTF-8

4. 审计日志操作对象 国际化Key列表获取

请求

请求网址

log/auditlog/key/getOperateObjectKeys

请求方法

post

Content-Type

application/x-www-form-urlencoded; charset=UTF-8

请求参数

参数	类型	说明	是否必填
----	----	----	------

appCode	string	应用编码	否
---------	--------	------	---

响应

Content-Type

application/json;charset=UTF-8

2. 开发步骤（以UPMS为例）

2.1 包结构必须满足如下要求：

dao: com.nuctech..dao..*.*(..)

Service: com.nuctech..service..*.*(..)

Controller: com.nuctech..controller..*.*(..)

2.2 创建Maven工程

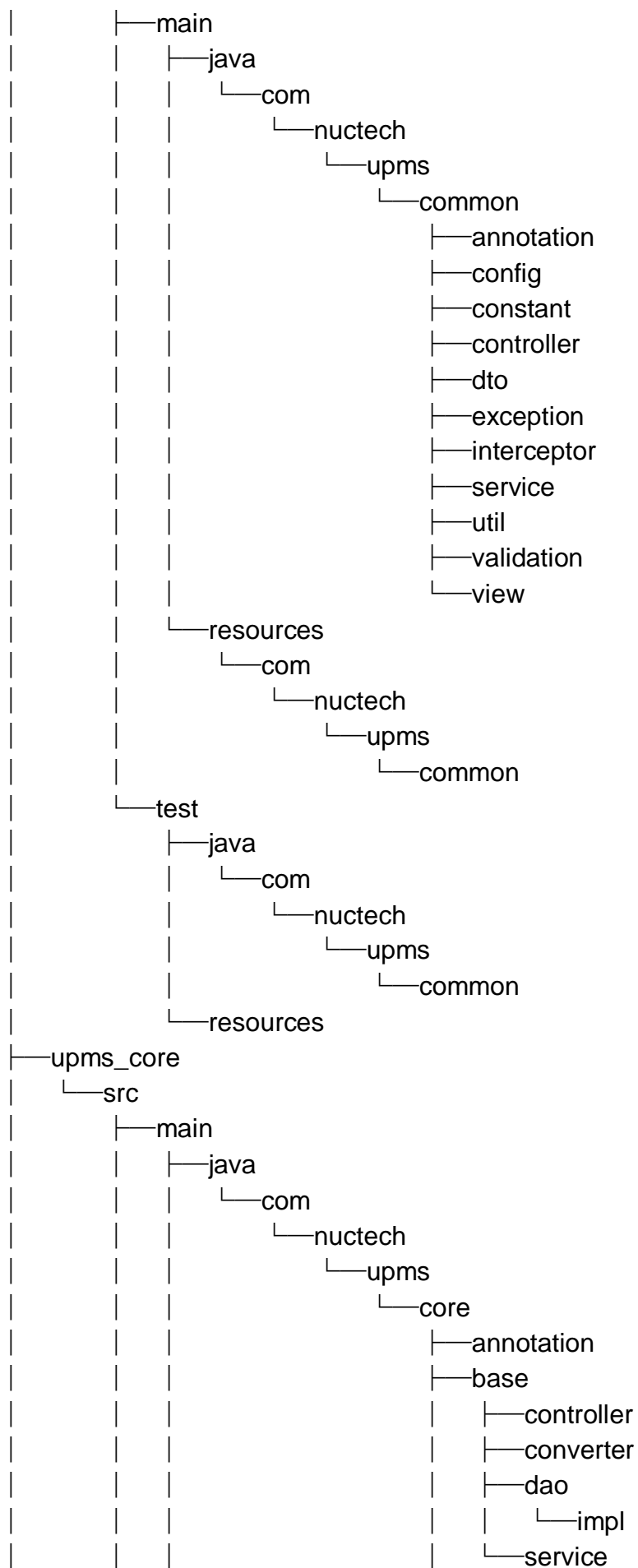
目录结构如下

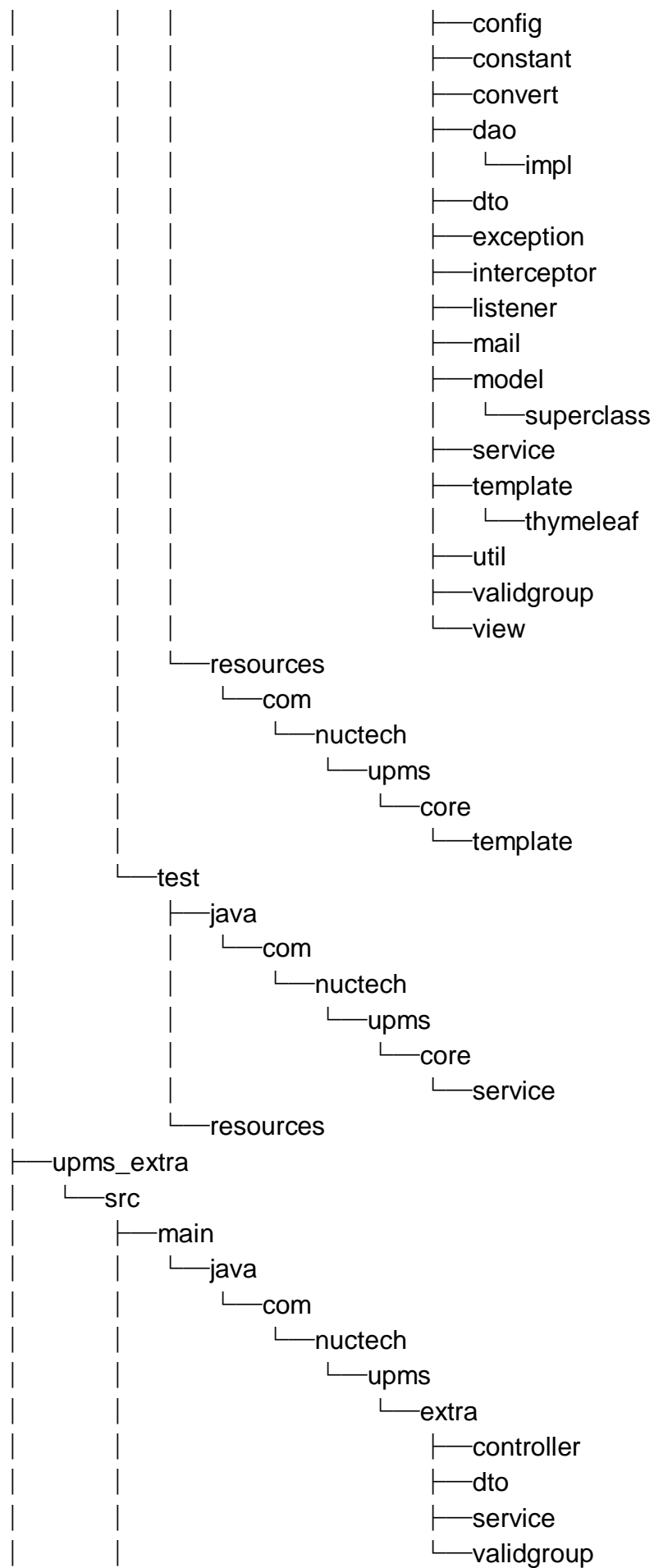
文件夹 PATH 列表

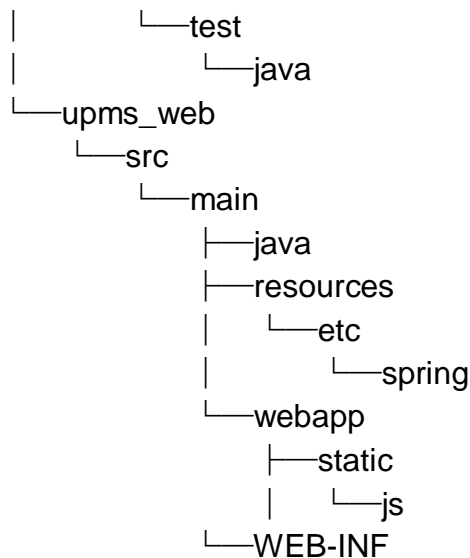
卷序列号为 0007-DDDF

D:.

```
├── jar
├── upms_api
│   ├── src
│   │   ├── main
│   │   │   ├── java
│   │   │   │   ├── com
│   │   │   │   │   ├── nuctech
│   │   │   │   │   │   ├── upms
│   │   │   │   │   │   │   ├── api
│   │   │   │   │   │   │   │   ├── controller
│   │   │   │   │   │   │   │   ├── dto
│   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   │   └── validgroup
│   │   │   └── test
│   │   │       ├── java
│   │   │       │   ├── com
│   │   │       │   │   ├── nuctech
│   │   │       │   │   │   ├── urpm
│   │   │       │   │   │   │   └── api
├── upms_common
│   └── src
```







2.3 pom.xml结构为:

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>com.nuctech.biz</groupId>
    <artifactId>parent</artifactId>
    <version>1.1.0</version>
  </parent>
  <groupId>com.nuctech.upms</groupId>
  <artifactId>upms</artifactId>
  <version>0.9.0</version>
  <packaging>pom</packaging>
  <name>upms</name>
  <modules>
    <module>upms_core</module>
    <module>upms_common</module>
    <module>upms_web</module>
    <module>upms_api</module><!-- api网关需要的接口 -->
    <module>upms_extra</module>
  </modules>
  <dependencies>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-email</artifactId>
      <version>1.5</version>
    </dependency>
    <dependency><!-- 正常项目无需引用该依赖 -->
      <groupId>com.nuctech.biz</groupId>
      <artifactId>biz.accesslog</artifactId>
      <version>1.1.0</version>
    </dependency>
    <dependency>

```



```
        <groupId>captcha</groupId>
        <artifactId>simplecaptcha</artifactId>
        <version>1.0.0</version>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <configuration>
                <archive>
                    <manifestEntries>
                        <Version>${rel_version}</Version>
                    </manifestEntries>
                </archive>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>
```