

# Java开发企业级权限管理系统

## 第5章 Spring MVC开发环境搭建与配置

### 5-1 Spring MVC开发环境搭建-pom.xml配置

#### 创建maven项目

.....

#### 配置pom.xml

```
<?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>

  <groupId>com.trt.permission</groupId>
  <artifactId>permission</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>permission Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
    <springframework.version>4.3.10.RELEASE</springframework.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-beans</artifactId>
      <version>${springframework.version}</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
```

```
<version>${springframework.version}</version>
</dependency>

<!--Spring MVC + Spring web -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>${springframework.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${springframework.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${springframework.version}</version>
</dependency>

<!-- mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.4.0</version>
</dependency>
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>1.3.0</version>
</dependency>

<!-- druid -->
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>druid</artifactId>
  <version>1.0.20</version>
</dependency>

<!-- mysql -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.30</version>
</dependency>

<!-- Lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.16.12</version>
</dependency>

<!-- Jackson -->
<dependency>
  <groupId>com.fasterxml.jackson.datatype</groupId>
  <artifactId>jackson-datatype-guava</artifactId>
  <version>2.5.3</version>
```

```

    </dependency>

    <build>
        <finalName>permission</finalName>
    </build>
</project>

```

## 5-2 Spring MVC开发环境搭建-web.xml配置

### 配置web.xml

```

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
    <display-name>Archetype Created Web Application</display-name>

    <!--监听上下文spring bean的加载 -->
    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>

    <!--Spring beans 配置文件所在目录 -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:applicationContext.xml</param-value>
    </context-param>

    <!--spring mvc 的配置-->
    <servlet>
        <servlet-name>spring</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
    </servlet>
    <servlet-mapping>
        <servlet-name>spring</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <!--Encoding（编码）-->
    <filter>
        <filter-name>encodingFilter</filter-name>
        <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
        <init-param>
            <param-name>encoding</param-name>
            <param-value>UTF-8</param-value>
        </init-param>
        <!--强制编码-->
        <init-param>
            <param-name>forceEncoding</param-name>
            <param-value>true</param-value>
        </init-param>
    </filter>

```

```

<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>

</web-app>

```

/和/\*的区别：

/：会匹配到/login这样路径型的url，不会匹配到\*.jsp这样后缀型的url

/\*：请求匹配到所有的url，路径型的，有后缀型的

## 5-3 Spring MVC开发环境搭建-spring-servlet.xml配置

### 配置spring-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
  http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
  http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

  <context:annotation-config/>

  <!-- 支持mvc注解驱动 -->
  <!--启动注解驱动的spring mvc功能-->
  <mvc:annotation-driven/>

  <!--启动包扫描功能-->
  <context:component-scan base-package="com.trt.permission.controller"/>
  <context:component-scan base-package="com.trt.permission.service"/>

  <!--
  当我们定义一个spring mvc接口的时候会使用@RequestMapping这个注解，
  RequestMappingHandlerMapping这个类把我定义好的@RequestMapping的
  注解，能找到对应的接口来进行处理，相当于是告诉我们请求是使用什么样的代码来处理，
  -->
  <bean
class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandl
erMapping"/>

  <!--基于BeanNameViewResolver解析器，自定义视图-->
  <!--
  BeanNameViewResolver相当于是告诉spring mvc项目，当我来一个请求之后，以什么格式数据返
  回给它，

```

```

    返回json数据, 还是.jsp页面, 甚至说可以下载一个文件pdf .js .css等, 做试图渲染的
    -->
    <bean class="org.springframework.web.servlet.view.BeanNameViewResolver"/>

    <!--Json返回用MappingJackson2JsonView处理-->
    <bean id="jsonView"
class="org.springframework.web.servlet.view.json.MappingJackson2JsonView"/>

    <!-- 视图解析器 -->
    <!--jsp返回时用InternalResourceViewResolver处理-->
    <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/views/" />
        <property name="suffix" value=".jsp" />
    </bean>

</beans>

```

## 5-4 Spring MVC开发环境搭建-applicationContext.xml配置

### 配置applicationContext.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd">

    <bean id="propertyConfigurer"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
        <property name="ignoreUnresolvablePlaceholders" value="true"/>
        <property name="locations">
            <list>
                <value>classpath:settings.properties</value>
            </list>
        </property>
    </bean>

    <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-
method="init" destroy-method="close">
        <property name="driverClassName" value="${db.driverClassName}"/>
        <property name="url" value="${db.url}"/>
        <property name="username" value="${db.username}"/>
        <property name="password" value="${db.password}"/>
        <property name="initialSize" value="3"/>
        <property name="minIdle" value="3"/>
        <property name="maxActive" value="20"/>
        <property name="maxWait" value="60000"/>
        <property name="filters" value="stat,wall"/>
    </bean>

```

```

<bean id="sqlSessionFactory"
class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="configLocation" value="classpath:mybatis-config.xml"/>
    <property name="dataSource" ref="dataSource"/>
    <property name="mapperLocations" value="classpath:mapper/*.xml"/>
</bean>

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <property name="basePackage" value="com.trt.permission.dao"/>
    <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>
</bean>

<!--tx-->
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"/>
</bean>

<tx:annotation-driven transaction-manager="transactionManager"/>

<!--druid-->
<bean id="stat-filter" class="com.alibaba.druid.filter.stat.StatFilter">
    <property name="slowSqlMillis" value="3000"/>
    <property name="logSlowSql" value="true"/>
    <property name="mergeSql" value="true"/>
</bean>
<bean id="wall-filter" class="com.alibaba.druid.wall.WallFilter">
    <property name="dbType" value="mysql"/>
</bean>
</beans>

```

## 配置settings.properties

```

db.driverClassName = com.mysql.jdbc.Driver
db.url = jdbc:mysql://127.0.0.1:3306/permission?
useUnicode=true&characterEncoding=UTF-8
db.username = root
db.password = root

```

## 5-5 Spring MVC开发环境搭建-druid配置

### 配置druid

web.xml

```

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
    <display-name>Archetype Created Web Application</display-name>

```

```

<!--druid-->
<servlet>
  <servlet-name>DruidStatServlet</servlet-name>
  <servlet-class>com.alibaba.druid.support.http.StatViewServlet</servlet-
class>
  <init-param>
    <param-name>loginUsername</param-name>
    <param-value>druid</param-value>
  </init-param>
  <init-param>
    <param-name>loginPassword</param-name>
    <param-value>druid</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>DruidStatServlet</servlet-name>
  <url-pattern>/sys/druid/*</url-pattern>
</servlet-mapping>
<filter>
  <filter-name>DruidwebStatFilter</filter-name>
  <filter-class>com.alibaba.druid.support.http.WebStatFilter</filter-class>
  <init-param>
    <param-name>exclusions</param-name>
    <param-value>*.js,*.css,*.jpg,*.png,*.ico,*.gif,/sys/druid/*</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>DruidwebStatFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

</web-app>

```

## 5-6 SpringMVC开发环境搭建-mybatis-config.xml配置

### 配置mybatis-config.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
  <settings>
    <setting name="safeResultHandlerEnabled" value="true"/>
    <setting name="cacheEnabled" value="false"/>
    <setting name="useGeneratedKeys" value="true"/>
  </settings>

  <!--type取别名，这里type就是本地的class-->
  <!--
  <typeAliases>
    <typeAlias type="com.xxxx" alias="bb"/>
  </typeAliases>
  -->

```

```

    <!-- 监听-->
    <!--
    <plugins>
        <plugin interceptor=""></plugin>
    </plugins>
    -->

    <!--
    <typeHandlers>
        <typeHandler handler="">
    </typeHandlers>
    -->

</configuration>

```

## 5-7 SpringMVC开发环境搭建-logback.xml配置

### 引入logback依赖

pom.xml

```

<!-- logback -->
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-core</artifactId>
    <version>1.1.8</version>
</dependency>
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.1.8</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.22</version>
</dependency>

```

### logback和log4j区别

logback相同情况，执行速度更快，它原生实现了的slf4jAPI，有更丰富的文档，同时支持xml,grald的方式配置，使用logback的时候还可以对配置文件做自动的热加载，如果项目在运行时，改掉logback配置文件，让它自动生效，这个时候系统出什么问题的话，或者对log做些处理的话直接修改文件，不需要重启项目，还有特别多的好处自动删除日志的归档并且自动压缩日志成为归档文件，支持特别大的过滤器，支持appender，appender通过不同的方式处理日志，支持参数化输出...等等；log4j需要中间转换层，所以执行相对慢；

### 配置logback.xml



```

<?xml version="1.0" encoding="UTF-8"?>
<!--
scan="true": 配置文件发生改变将重新加载，默认为true
scanPeriod="60 seconds": 设置配置文件是否有修改的时间间隔，
如果没有时间单位，默认为毫秒，scan为true的时候才生效，默认的时间间隔为一分钟，
scanPeriod="60"不指定单位默认是毫秒
-->
<configuration scan="true" scanPeriod="60 seconds">

    <!--ConsoleAppender将日志文件输出到控制台，当tomcat启动后，将日志文件输出到
Appender.log文件-->
    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <!--日志输出格式-->
        <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger -
%msg%n</pattern>
        </encoder>
    </appender>

    <!--
    <appender name="permission"
class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>${catalina.home}/logs/permission.log</file>
        <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <FileNamePattern>${catalina.home}/logs/permission.%d{yyyy-MM-
dd}.log.gz</FileNamePattern>
        </rollingPolicy>
        <layout class="ch.qos.logback.classic.PatternLayout">
            <pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger -
%msg%n</pattern>
        </layout>
    </appender>

    <logger name="xxx" level="INFO">
        <appender-ref ref="permission"/>
    </logger>
-->

    <!-- TRACE < DEBUG < INFO < WARN < ERROR -->
    <root level="INFO">
        <appender-ref ref="STDOUT"/>
    </root>

</configuration>

```

## 5-8 SpringMVC开发环境验证

### 测试

TestController类

```

package com.trt.permission.controller;

import jdk.nashorn.internal.runtime.options.LoggingOption;

```

```

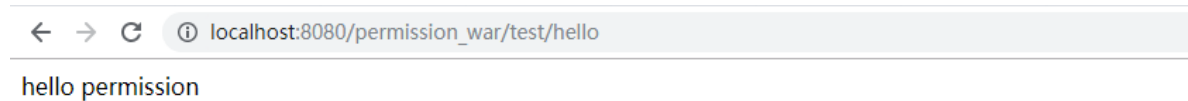
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.logging.Log;
import org.slf4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import sun.reflect.generics.reflectiveObjects.LazyReflectiveObjectGenerator;

/**
 * @version 1.0
 * @author: trt
 * @description: 测试
 * @create: 2020-01-19 15:51
 */
@Controller
@RequestMapping("/test")
@Slf4j
public class TestController {

    @RequestMapping("/hello")
    @ResponseBody
    public String hello(){
        //log.info("hello");
        return "hello permission";
    }
}

```

## 效果



## 第6章 项目准备与核心辅助工具类开发

### 权限管理 - 编码部分

#### 权限管理开发 - 准备

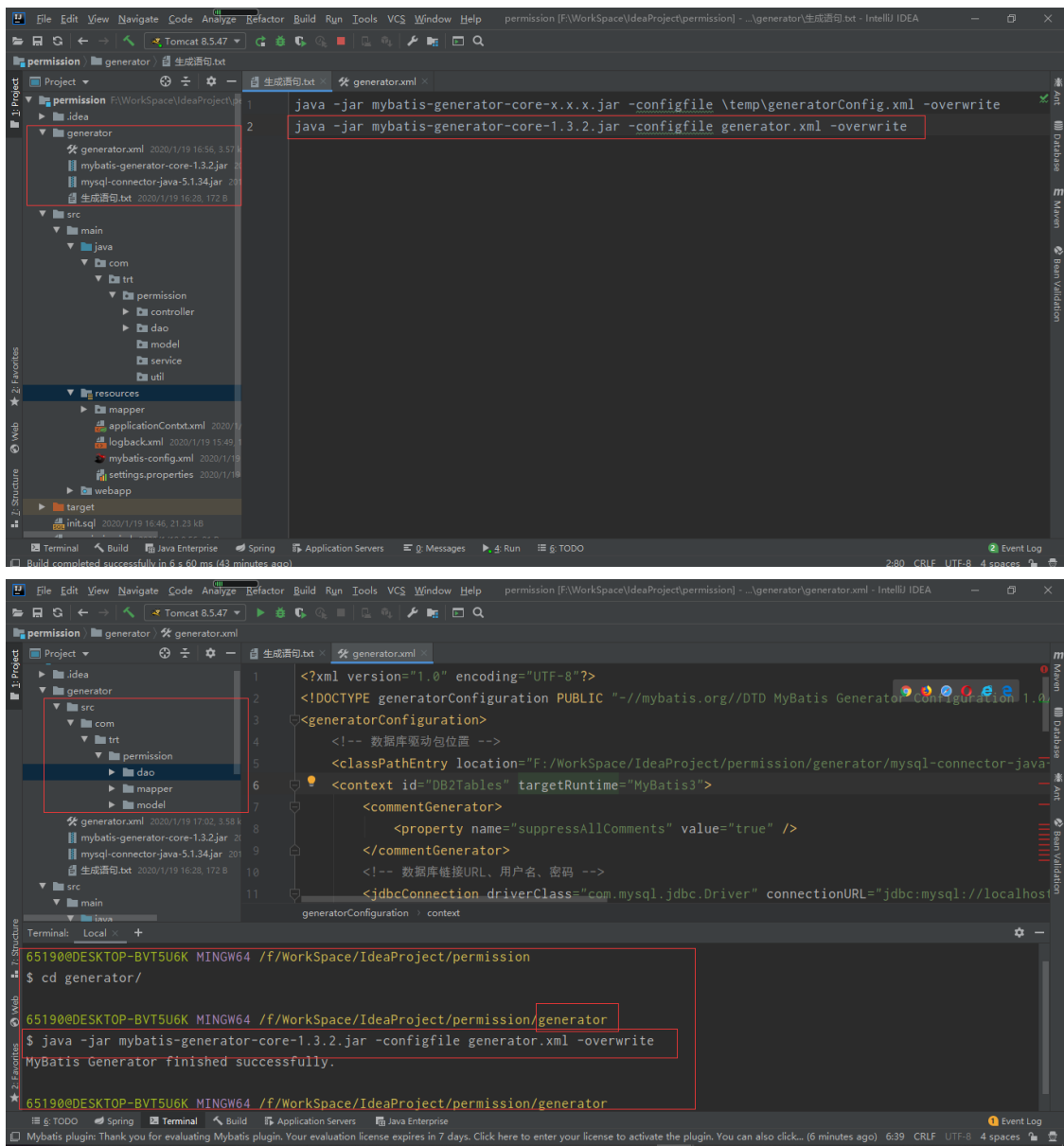
- 核心类生成- Mybatis generate
- 项目接口定义 - json, page

#### 权限管理开发 - 工具相关

- 校验工具 - validator
- Json转换工具类 - jackson convert
- 获取Spring上下文 - applicationContext
- Http请求前后端监听 - interceptor

### 6-1 核心类生成-Mybatis Generator

- 核心类生成- Mybatis generate



- 配置generator.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org//DTD MyBatis
Generator Configuration 1.0//EN" "http://mybatis.org/dtd/mybatis-generator-
config_1_0.dtd">
<generatorConfiguration>
  <!-- 数据库驱动包位置 -->
  <classPathEntry
location="F:/Workspace/IdeaProject/permission/generator/mysql-connector-
java-5.1.34.jar" /> <!-- 1 -->
  <context id="DB2Tables" targetRuntime="MyBatis3">
    <commentGenerator>
      <property name="suppressAllComments" value="true" />
    </commentGenerator>
    <!-- 数据库链接URL、用户名、密码 -->
    <jdbcConnection driverClass="com.mysql.jdbc.Driver"
connectionURL="jdbc:mysql://localhost:3306/permission?
characterEncoding=utf8" userId="root" password="root"> <!-- 2 -->
  </jdbcConnection>
  <jdbcTypeResolver>
    <property name="forceBigDecimals" value="false" />
  </jdbcTypeResolver>

```

```

<!-- 生成模型的包名和位置 --> <!-- 3 -->
<javaModelGenerator targetPackage="com.trt.permission.model"
targetProject="F:/WorkSpace/IdeaProject/permission/generator/src">
    <property name="enableSubPackages" value="true" />
    <property name="trimStrings" value="true" />
</javaModelGenerator>
<!-- 生成的映射文件包名和位置 --> <!-- 4 -->
<sqlMapGenerator targetPackage="com.trt.permission.mapper"
targetProject="F:/WorkSpace/IdeaProject/permission/generator/src">
    <property name="enableSubPackages" value="true" />
</sqlMapGenerator>
<!-- 生成DAO的包名和位置 --> <!-- 5 -->
<javaClientGenerator type="XMLMAPPER"
targetPackage="com.trt.permission.dao"
targetProject="F:/WorkSpace/IdeaProject/permission/generator/src">
    <property name="enableSubPackages" value="true" />
</javaClientGenerator>
<!-- 要生成那些表(更改tableName和domainObjectName就可以) --><!-- 6 -->
<table tableName="sys_user" domainObjectName="SysUser"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_dept" domainObjectName="SysDept"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_acl" domainObjectName="SysAcl"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_acl_module" domainObjectName="SysAclModule"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_role" domainObjectName="SysRole"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_role_acl" domainObjectName="SysRoleAcl"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_role_user" domainObjectName="SysRoleUser"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
    <table tableName="sys_log" domainObjectName="SysLog"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
</context>
</generatorConfiguration>

```

- 在项目的permission/generator路径下运行

```
$ java -jar mybatis-generator-core-1.3.2.jar -configfile generator.xml -overwrite
```

- 效果

.....

生成SysLog和SysLogWithBLOBS原因:

因为数据库字段如果是text, 会把text字段单独拿出来生成一个类, 并且继承原来的类, 原因是mysql存储机制造成。

## 6-2 项目接口定义-json,page

### JsonData类

```
package com.trt.permission.common;

import lombok.Getter;
import lombok.Setter;

/**
 * @version 1.0
 * @author: trt
 * @description: Json请求返回的结构
 * @create: 2020-01-19 17:24
 */
@Getter
@Setter
public class JsonData {

    private boolean ret; //返回结果

    private String msg; //有异常的时候, 返回的msg

    private Object data; //正常返回的时候, 返回给前台的数据

    public JsonData(boolean ret) {
        this.ret = ret;
    }

    public static JsonData success(Object object, String msg){
        JsonData jsonData = new JsonData(true);
        jsonData.data = object;
        jsonData.msg = msg;
        return jsonData;
    }

    public static JsonData success(Object object){
        JsonData jsonData = new JsonData(true);
        jsonData.data = object;
        return jsonData;
    }

    public static JsonData success(){
        return new JsonData(true);
    }

    public static JsonData fail(String msg){
        JsonData jsonData = new JsonData(false);
```

```

        jsonData.msg = msg;
        return jsonData;
    }
}

```

## 6-3 接口请求全局异常处理-设计与验证

### 加载jsp api依赖

```

<!-- jsp api -->
<dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>jsp-api</artifactId>
    <version>6.0.36</version>
</dependency>

```

### SpringExceptionHandler

```

package com.trt.permission.common;

import com.trt.permission.exception.PermissionException;
import lombok.extern.slf4j.Slf4j;
import org.springframework.web.servlet.HandlerExceptionResolver;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * @version 1.0
 * @author: trt
 * @description: 全局异常的处理类
 * 该类（HandlerExceptionResolver）被Spring管理的话，我们的全局异常，在http进行返回的时候
 就会被SpringExceptionHandler捕捉到
 * @create: 2020-01-19 17:39
 */
@Slf4j
public class SpringExceptionHandler implements HandlerExceptionResolver {
    @Override
    public ModelAndView resolveException(HttpServletRequest request,
        HttpServletResponse response, Object handler, Exception ex) {
        String url = request.getRequestURL().toString();
        ModelAndView mv;
        String defaultMsg = "system error";

        // 这里我们要求项目中所有请求json数据，都使用.json结尾
        if (url.endsWith(".json")) {
            if (ex instanceof PermissionException) {
                JsonData result = JsonData.fail(ex.getMessage());
                mv = new ModelAndView("jsonView", result.toMap());
            } else {
                //log.error("unknown json exception, url:" + url, ex);
                JsonData result = JsonData.fail(defaultMsg);
                mv = new ModelAndView("jsonView", result.toMap());
            }
        }
    }
}

```

```

    }
    } else if (url.endsWith(".page")){ // 这里我们要求项目中所有请求page页面，都使用.page结尾
        //log.error("unknown page exception, url:" + url, ex);
        JsonData result = JsonData.fail(defaultMsg);
        mv = new ModelAndView("exception", result.toMap());
    } else {
        //log.error("unknow exception, url:" + url, ex);
        JsonData result = JsonData.fail(defaultMsg);
        mv = new ModelAndView("jsonview", result.toMap());
    }
    return mv;
}
}
}

```

## PermissionException

```

package com.trt.permission.exception;

/**
 * @version 1.0
 * @author: trt
 * @description: 自定义异常
 * @create: 2020-01-19 17:54
 */
public class PermissionException extends RuntimeException {

    public PermissionException() {
        super();
    }

    public PermissionException(String message) {
        super(message);
    }

    public PermissionException(String message, Throwable cause) {
        super(message, cause);
    }

    public PermissionException(Throwable cause) {
        super(cause);
    }

    protected PermissionException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
    }
}

```

## spring-servlet.xml

添加自定义全局异常到spring mvc中

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns:context="http://www.springframework.org/schema/context"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

<!--
当我们定义一个spring mvc接口的时候会使用@RequestMapping这个注解，
RequestMappingHandlerMapping这个类把我定义好的@RequestMapping的
注解，能找到对应的接口来进行处理，相当于是告诉我们请求是使用什么样的代码来处理，
-->
<!-- <bean
class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandl
erMapping"/>-->

<!--自定义全局异常处理-->
<bean class="com.trt.permission.common.SpringExceptionHandler"/>

</beans>

```

## JsonData添加到Map方法

```

package com.trt.permission.common;

import lombok.Getter;
import lombok.Setter;

import java.util.HashMap;
import java.util.Map;

/**
 * @version 1.0
 * @author: trt
 * @description: Json请求返回的结构
 * @create: 2020-01-19 17:24
 */
@Getter
@Setter
public class JsonData {

    private boolean ret; //返回结果

    private String msg; //有异常的时候，返回的msg

    private Object data; //正常返回的时候，返回给前台的数据

    public JsonData(boolean ret) {
        this.ret = ret;
    }

    public static JsonData success(Object object, String msg){
        JsonData jsonData = new JsonData(true);
        jsonData.data = object;
        jsonData.msg = msg;
    }
}

```



```

        return jsonData;
    }

    public static JsonData success(Object object){
        JsonData jsonData = new JsonData(true);
        jsonData.data = object;
        return jsonData;
    }

    public static JsonData success(){
        return new JsonData(true);
    }

    public static JsonData fail(String msg){
        JsonData jsonData = new JsonData(false);
        jsonData.msg = msg;
        return jsonData;
    }

    public Map<String, Object> toMap() {
        HashMap<String, Object> result = new HashMap<String, Object>();
        result.put("ret", ret);
        result.put("msg", msg);
        result.put("data", data);
        return result;
    }
}

```

## 测试

```

package com.trt.permission.controller;

import com.trt.permission.common.JsonData;
import com.trt.permission.exception.PermissionException;
import jdk.nashorn.internal.runtime.options.LoggingOption;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.logging.Log;
import org.slf4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import sun.reflect.generics.reflectiveObjects.LazyReflectiveObjectGenerator;

/**
 * @version 1.0
 * @author: trt
 * @description: 测试
 * @create: 2020-01-19 15:51
 */
@Controller
@RequestMapping("/test")
@Slf4j
public class TestController {

    @RequestMapping("/hello.json")
    @ResponseBody
    public JsonData hello(){

```

```

        //log.info("hello");
        //throw new PermissionException("test exception");
        //throw new RuntimeException("test exception");
        return JsonData.success("hello permission");
    }
}

```

← → ↻ ⓘ localhost:8080/permission\_war/test/hello.json

```

{"ret":true,"msg":null,"data":"hello permission"}

```

```

@RequestMapping("/hello.json")
@ResponseBody
public JsonData hello(){
    //log.info("hello");
    throw new PermissionException("test exception");
    //throw new RuntimeException("test exception");
    //return JsonData.success("hello permission");
}

```

← → ↻ ⓘ localhost:8080/permission\_war/test/hello.json

```

{"ret":false,"msg":"test exception","data":null}

```

```

@RequestMapping("/hello.json")
@ResponseBody
public JsonData hello(){
    //log.info("hello");
    //throw new PermissionException("test exception");
    throw new RuntimeException("test exception");
    //return JsonData.success("hello permission");
}

```

← → ↻ ⓘ localhost:8080/permission\_war/test/hello.json

```

{"ret":false,"msg":"System error","data":null}

```

## 6-4 校验工具-BeanValidator开发

### 引入依赖

pom.xml

```

<!-- validator -->
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>1.1.0.Final</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>5.2.4.Final</version>
</dependency>

```

## BeanValidator

```

package com.trt.permission.util;
import com.google.common.base.Preconditions;
import com.google.common.collect.Lists;
import com.google.common.collect.Maps;
import org.apache.commons.collections.MapUtils;

import javax.validation.ConstraintViolation;
import javax.validation.Validation;
import javax.validation.Validator;
import javax.validation.ValidatorFactory;
import java.util.Collection;
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Set;
/**
 * @version 1.0
 * @author: trt
 * @description:
 * @create: 2020-01-19 18:25
 */
public class BeanValidator {

    private static ValidatorFactory validatorFactory =
Validation.buildDefaultValidatorFactory();

    /**
     * 单个类检验方法
     * @param t
     * @param groups
     * @param <T>
     * @return
     */
    public static <T> Map<String, String> validate(T t, Class... groups) {
        Validator validator = validatorFactory.getValidator();
        Set validateResult = validator.validate(t, groups);
        if (validateResult.isEmpty()) {
            return Collections.emptyMap();
        } else {
            LinkedHashMap errors = Maps.newLinkedHashMap();
            Iterator iterator = validateResult.iterator();

```

```

        while (iterator.hasNext()) {
            ConstraintViolation violation =
            (ConstraintViolation)iterator.next();
            errors.put(violation.getPropertyPath().toString(),
            violation.getMessage());
        }
        return errors;
    }
}

/**
 * 集合校验方法
 * @param collection
 * @return
 */
public static Map<String, String> validateList(Collection<?> collection) {
    Preconditions.checkNotNull(collection);
    Iterator iterator = collection.iterator();
    Map errors;

    do {
        if (!iterator.hasNext()) {
            return Collections.emptyMap();
        }
        Object object = iterator.next();
        errors = validate(object, new Class[0]);
    } while (errors.isEmpty());

    return errors;
}

public static Map<String, String> validateObject(Object first, Object...
objects) {
    if (objects != null && objects.length > 0) {
        return validateList(Lists.asList(first, objects));
    } else {
        return validate(first, new Class[0]);
    }
}
}

```

## TestVo实体类

```

package com.trt.permission.param;

import lombok.Getter;
import lombok.Setter;
import org.hibernate.validator.constraints.NotBlank;
import javax.validation.constraints.NotNull;

/**
 * @version 1.0
 * @author: trt
 * @description:
 * @create: 2020-01-19 18:38
 */
@Getter

```

```

@Setter
public class TestVo {

    @NotNull
    private Integer id;

    @NotBlank
    private String msg;
}

```

## TestController

```

package com.trt.permission.controller;

import com.trt.permission.common.JsonData;
import com.trt.permission.dao.SysAclModuleMapper;
import com.trt.permission.exception.PermissionException;
import com.trt.permission.model.SysAclModule;
import com.trt.permission.param.TestVo;
import com.trt.permission.util.BeanValidator;
import jdk.nashorn.internal.runtime.options.LoggingOption;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.logging.Log;
import org.slf4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import sun.reflect.generics.reflectiveObjects.LazyReflectiveObjectGenerator;

import java.util.Map;

/**
 * @version 1.0
 * @author: trt
 * @description: 测试
 * @create: 2020-01-19 15:51
 */
@Controller
@RequestMapping("/test")
@Slf4j
public class TestController {

    @RequestMapping("/validate.json")
    @ResponseBody
    public JsonData validate(TestVo vo) {
        //log.info("validate");
        try {
            Map<String, String> map = BeanValidator.validateObject(vo);
            if (map != null && map.entrySet().size() > 0) {
                for (Map.Entry<String, String> entry: map.entrySet()) {
                    //log.info("{}->{}", entry.getKey(), entry.getValue());
                    System.out.println("{}->{}"+entry.getKey()+"
"+entry.getValue());
                }
            }
        } catch (Exception e){

```

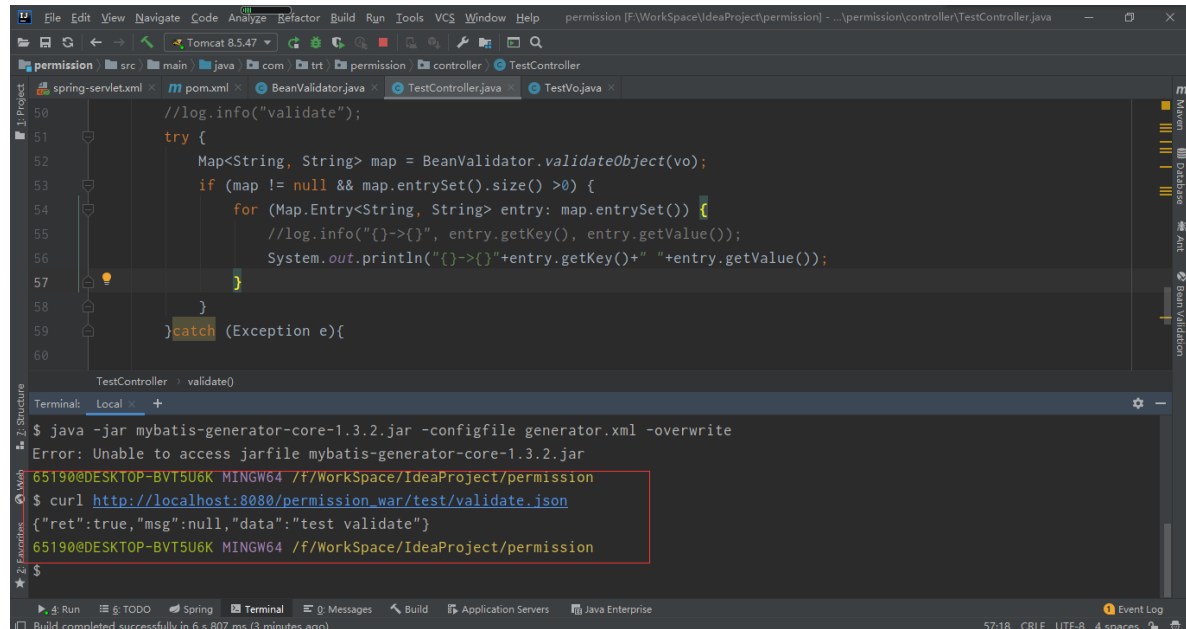
```

    }
    return JsonData.success("test validate");
}
}

```

## 测试

curl [http://localhost:8080/permission\\_war/test/validate.json](http://localhost:8080/permission_war/test/validate.json)



## ParamException

```

package com.trt.permission.exception;

/**
 * @version 1.0
 * @author: trt
 * @description:
 * @create: 2020-01-19 19:19
 */
public class ParamException extends RuntimeException {

    public ParamException() {
    }

    public ParamException(String message) {
        super(message);
    }

    public ParamException(String message, Throwable cause) {
        super(message, cause);
    }

    public ParamException(Throwable cause) {
        super(cause);
    }

    public ParamException(String message, Throwable cause, boolean
enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
    }

```

```
}  
}
```

## BeanValidator添加方法

```
package com.trt.permission.util;  
import com.google.common.base.Preconditions;  
import com.google.common.collect.Lists;  
import com.google.common.collect.Maps;  
import com.trt.permission.exception.ParamException;  
import org.apache.commons.collections.MapUtils;  
  
import javax.validation.ConstraintViolation;  
import javax.validation.Validation;  
import javax.validation.Validator;  
import javax.validation.ValidatorFactory;  
import java.util.Collection;  
import java.util.Collections;  
import java.util.Iterator;  
import java.util.LinkedHashMap;  
import java.util.Map;  
import java.util.Set;  
/**  
 * @version 1.0  
 * @author: trt  
 * @description:  
 * @create: 2020-01-19 18:25  
 */  
public class BeanValidator {  
  
    private static ValidatorFactory validatorFactory =  
Validation.buildDefaultValidatorFactory();  
  
    /**  
     * 单个类检验方法  
     * @param t  
     * @param groups  
     * @param <T>  
     * @return  
     */  
    public static <T> Map<String, String> validate(T t, Class... groups) {  
        Validator validator = validatorFactory.getValidator();  
        Set validateResult = validator.validate(t, groups);  
        if (validateResult.isEmpty()) {  
            return Collections.emptyMap();  
        } else {  
            LinkedHashMap errors = Maps.newLinkedHashMap();  
            Iterator iterator = validateResult.iterator();  
            while (iterator.hasNext()) {  
                ConstraintViolation violation =  
(ConstraintViolation)iterator.next();  
                errors.put(violation.getPropertyPath().toString(),  
violation.getMessage());  
            }  
            return errors;  
        }  
    }  
}
```

```

    }
}

/**
 * 集合校验方法
 * @param collection
 * @return
 */
public static Map<String, String> validateList(Collection<?> collection) {
    Preconditions.checkNotNull(collection);
    Iterator iterator = collection.iterator();
    Map errors;

    do {
        if (!iterator.hasNext()) {
            return Collections.emptyMap();
        }
        Object object = iterator.next();
        errors = validate(object, new Class[0]);
    } while (errors.isEmpty());

    return errors;
}

public static Map<String, String> validateObject(Object first, Object...
objects) {
    if (objects != null && objects.length > 0) {
        return validateList(Lists.asList(first, objects));
    } else {
        return validate(first, new Class[0]);
    }
}

public static void check(Object param) throws ParamException {
    Map<String, String> map = BeanValidator.validateObject(param);
    //MapUtils.isEmpty(map) 等于 map != null && map.entrySet().size()>0
    if (MapUtils.isEmpty(map)) {
        throw new ParamException(map.toString());
    }
}
}

```

MapUtils.isEmpty(map)引入依赖

```

<!-- tools -->
<dependency>
    <groupId>commons-collections</groupId>
    <artifactId>commons-collections</artifactId>
    <version>3.2.2</version>
</dependency>
<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.10</version>
</dependency>

```



## 在 SpringExceptionHandler 添加 ParamException 异常

```
package com.trt.permission.common;

import com.trt.permission.exception.ParamException;
import com.trt.permission.exception.PermissionException;
import lombok.extern.slf4j.Slf4j;
import org.springframework.web.servlet.HandlerExceptionHandler;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * @version 1.0
 * @author: trt
 * @description: 全局异常的处理类
 * 该类（HandlerExceptionHandler）被Spring管理的话，我们的全局异常，在http进行返回的时候
 * 就会被SpringExceptionHandler捕捉到
 * @create: 2020-01-19 17:39
 */
@Slf4j
public class SpringExceptionHandler implements HandlerExceptionHandler {
    @Override
    public ModelAndView resolveException(HttpServletRequest request,
        HttpServletResponse response, Object handler, Exception ex) {
        String url = request.getRequestURL().toString();
        ModelAndView mv;
        String defaultMsg = "System error";

        // 这里我们要求项目中所有请求json数据，都使用.json结尾
        if (url.endsWith(".json")) {
            if (ex instanceof PermissionException || ex instanceof
                ParamException) {
                JsonData result = JsonData.fail(ex.getMessage());
                mv = new ModelAndView("jsonview", result.toMap());
            } else {
                //log.error("unknown json exception, url:" + url, ex);
                JsonData result = JsonData.fail(defaultMsg);
                mv = new ModelAndView("jsonview", result.toMap());
            }
        } else if (url.endsWith(".page")) { // 这里我们要求项目中所有请求page页面，都使
            //log.error("unknown page exception, url:" + url, ex);
            JsonData result = JsonData.fail(defaultMsg);
            mv = new ModelAndView("exception", result.toMap());
        } else {
            //log.error("unknow exception, url:" + url, ex);
            JsonData result = JsonData.fail(defaultMsg);
            mv = new ModelAndView("jsonview", result.toMap());
        }
        return mv;
    }
}
```

## TestController

```
@RequestMapping("/validate.json")
@ResponseBody
public JsonData validate(TestVo vo) {
    //log.info("validate");
    try {
        Map<String, String> map = BeanValidator.validateObject(vo);
        //MapUtils.isEmpty(map) 等于 map != null && map.entrySet().size()>0
        if (map != null && map.entrySet().size() > 0) {
            for (Map.Entry<String, String> entry: map.entrySet()) {
                //log.info("{}->{}", entry.getKey(), entry.getValue());
                System.out.println("{}->{}"+entry.getKey()+"
"+entry.getValue());
            }
        }
    } catch (Exception e) {
    }
    return JsonData.success("test validate");
}

@RequestMapping("/validate.json")
@ResponseBody
public JsonData validate2(TestVo vo) throws ParamException {
    //log.info("validate");
    BeanValidator.check(vo);
    return JsonData.success("test validate");
}
```

## 6-5 Json转化工具-JsonMapper开发

### 添加json依赖

```
<!-- jackson -->
<dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-core-asl</artifactId>
    <version>1.9.13</version>
</dependency>
<dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-mapper-asl</artifactId>
    <version>1.9.13</version>
</dependency>
```

### JsonMapper

```
package com.trt.permission.util;

import lombok.extern.slf4j.Slf4j;
import org.codehaus.jackson.map.DeserializationConfig;
import org.codehaus.jackson.map.ObjectMapper;
```

```

import org.codehaus.jackson.map.SerializationConfig;
import org.codehaus.jackson.map.annotate.JsonSerialize;
import org.codehaus.jackson.map.ser.impl.SimpleFilterProvider;
import org.codehaus.jackson.type.TypeReference;

/**
 * @version 1.0
 * @author: trt
 * @description: Json转换工具类
 * @create: 2020-01-19 19:44
 */
@Slf4j
public class JsonMapper {

    private static ObjectMapper objectMapper = new ObjectMapper();

    static {
        // config

        objectMapper.disable(DeserializationConfig.Feature.FAIL_ON_UNKNOWN_PROPERTIES);
        objectMapper.configure(SerializationConfig.Feature.FAIL_ON_EMPTY_BEANS,
            false);
        objectMapper.setFilters(new
            SimpleFilterProvider().setFailOnUnknownId(false));

        objectMapper.setSerializationInclusion(JsonSerialize.Inclusion.NON_EMPTY);
    }

    public static <T> String obj2String(T src) {
        if (src == null) {
            return null;
        }
        try {
            return src instanceof String ? (String) src :
                objectMapper.writeValueAsString(src);
        } catch (Exception e) {
            //log.warn("parse object to String exception, error:{}", e);
            return null;
        }
    }

    public static <T> T string2Obj(String src, TypeReference<T> typeReference) {
        if (src == null || typeReference == null) {
            return null;
        }
        try {
            return (T) (typeReference.getType().equals(String.class) ? src :
                objectMapper.readValue(src, typeReference));
        } catch (Exception e) {
            //log.warn("parse String to Object exception, String:{},
                TypeReference<T>:{}, error:{}", src, typeReference.getType(), e);
            return null;
        }
    }
}

```

## 6-6 获取Spring上下文工具-ApplicationContextHelper开发

### ApplicationContextHelper

```
package com.trt.permission.common;

import org.springframework.beans.BeansException;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;
import org.springframework.stereotype.Component;
/**
 * @version 1.0
 * @author: trt
 * @description: 获取Spring上下文工具类
 * @create: 2020-01-19 19:52
 */
@Component("applicationContextHelper")
public class ApplicationContextHelper implements ApplicationContextAware {

    private static ApplicationContext applicationContext;

    public void setApplicationContext(ApplicationContext context) throws
BeansException {
        applicationContext = context;
    }

    public static <T> T popBean(Class<T> clazz) {
        if (applicationContext == null) {
            return null;
        }
        return applicationContext.getBean(clazz);
    }

    public static <T> T popBean(String name, Class<T> clazz) {
        if (applicationContext == null) {
            return null;
        }
        return applicationContext.getBean(name, clazz);
    }
}
```

### Spring mvc的配置

spring-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <!--获取Spring上下文工具类-->
    <bean class="com.trt.permission.common.ApplicationContextHelper" lazy-
init="false"/>

</beans>
```

## TestController

```
package com.trt.permission.controller;

import com.trt.permission.common.ApplicationContextHelper;
import com.trt.permission.common.JsonData;
import com.trt.permission.dao.SysAclModuleMapper;
import com.trt.permission.exception.ParamException;
import com.trt.permission.exception.PermissionException;
import com.trt.permission.model.SysAclModule;
import com.trt.permission.param.TestVo;
import com.trt.permission.util.BeanValidator;
import jdk.nashorn.internal.runtime.options.LoggingOption;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.logging.Log;
import org.slf4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import sun.reflect.generics.reflectiveObjects.LazyReflectiveObjectGenerator;

import java.util.Map;

/**
 * @version 1.0
 * @author: trt
 * @description: 测试
 * @create: 2020-01-19 15:51
 */
@Controller
@RequestMapping("/test")
@Slf4j
public class TestController {

    @RequestMapping("/hello")
    @ResponseBody
    public String hello1(){
        //log.info("hello");
    }
}
```

```

        return "hello permission";
    }

    @RequestMapping("/hello.json")
    @ResponseBody
    public JsonData hello(){
        //log.info("hello");
        //throw new PermissionException("test exception");
        throw new RuntimeException("test exception");
        //return JsonData.success("hello permission");
    }

    @RequestMapping("/validate.json")
    @ResponseBody
    public JsonData validate(TestVo vo) {
        //log.info("validate");
        try {
            Map<String, String> map = BeanValidator.validateObject(vo);
            //MapUtils.isEmpty(map) 等于 map != null && map.entrySet().size()>0
            if (map != null && map.entrySet().size() > 0) {
                for (Map.Entry<String, String> entry: map.entrySet()) {
                    //log.info("{}->{}", entry.getKey(), entry.getValue());
                    System.out.println("{}->{}"+entry.getKey()+"
"+entry.getValue());
                }
            }
        } catch (Exception e){

        }

        return JsonData.success("test validate");
    }

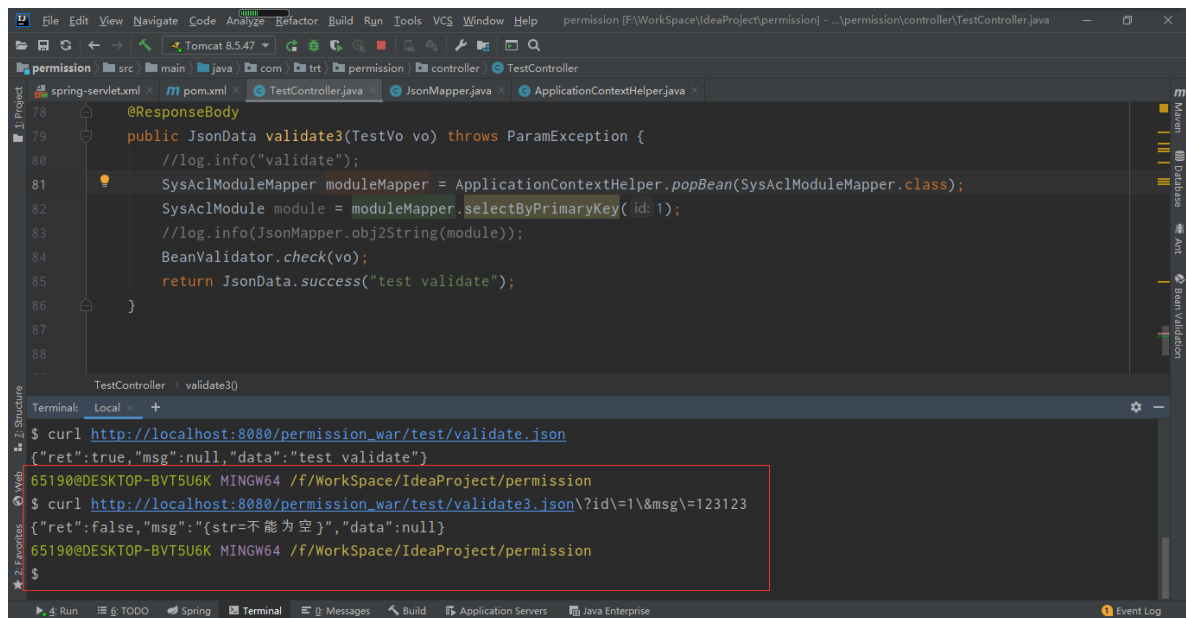
    @RequestMapping("/validate2.json")
    @ResponseBody
    public JsonData validate2(TestVo vo) throws ParamException {
        //log.info("validate");
        BeanValidator.check(vo);
        return JsonData.success("test validate");
    }

    @RequestMapping("/validate3.json")
    @ResponseBody
    public JsonData validate3(TestVo vo) throws ParamException {
        //log.info("validate");
        SysAclModuleMapper moduleMapper =
ApplicationContextHolder.getBean(SysAclModuleMapper.class);
        SysAclModule module = moduleMapper.selectByPrimaryKey(1);
        //log.info(JsonMapper.obj2String(module));
        BeanValidator.check(vo);
        return JsonData.success("test validate");
    }
}

```

## 测试

curl [http://localhost:8080/permission\\_war/test/validate3.json](http://localhost:8080/permission_war/test/validate3.json)?id=1&msg=123123



## 6-7 Http请求前后监听工具-HttpInterceptor开发

Http请求前后端监听 - interceptor

### HttpInterceptor

```
package com.trt.permission.common;

import com.trt.permission.util.JsonMapper;
import lombok.extern.slf4j.Slf4j;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.handler.HandlerInterceptorAdapter;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.Map;
/**
 * @version 1.0
 * @author: trt
 * @description: Http请求前后端监听工具类
 * @create: 2020-01-19 20:07
 */
@Slf4j
public class HttpInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
        String url = request.getRequestURI().toString();
        Map parameterMap = request.getParameterMap();
        //log.info("request start. url:{}, params:{}, url,
        JsonMapper.obj2String(parameterMap));
        return true;
    }

    @Override
    public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) throws Exception {
```

```

        String url = request.getRequestURI().toString();
        Map parameterMap = request.getParameterMap();
        //      log.info("request finished. url:{}, params:{}", url,
        JsonMapper.obj2String(parameterMap));
    }

    @Override
    public void afterCompletion(HttpServletRequest request, HttpServletResponse
response, Object handler, Exception ex) throws Exception {
        String url = request.getRequestURI().toString();
        Map parameterMap = request.getParameterMap();
        //log.info("request completed. url:{}, params:{}", url,
        JsonMapper.obj2String(parameterMap));
    }
}

```

## spring mvc配置

spring-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:context="http://www.springframework.org/schema/context"
        xmlns:mvc="http://www.springframework.org/schema/mvc"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <!--Http请求前后端监听-->
    <mvc:interceptors>
        <bean class="com.trt.permission.common.HttpInterceptor"/>
    </mvc:interceptors>

</beans>

```

## 测试

curl [http://localhost:8080/permission\\_war/test/validate3.json](http://localhost:8080/permission_war/test/validate3.json)

```

package com.trt.permission.param;

import lombok.Getter;
import lombok.Setter;
import org.hibernate.validator.constraints.NotBlank;
import org.hibernate.validator.constraints.NotEmpty;

import javax.validation.constraints.Max;
import javax.validation.constraints.Min;
import javax.validation.constraints.NotNull;
import java.util.List;

/**

```



```

* @version 1.0
* @author: trt
* @description: String 和 List集合注解不一样
* @create: 2020-01-19 18:38
*/
@Getter
@Setter
public class TestVo {

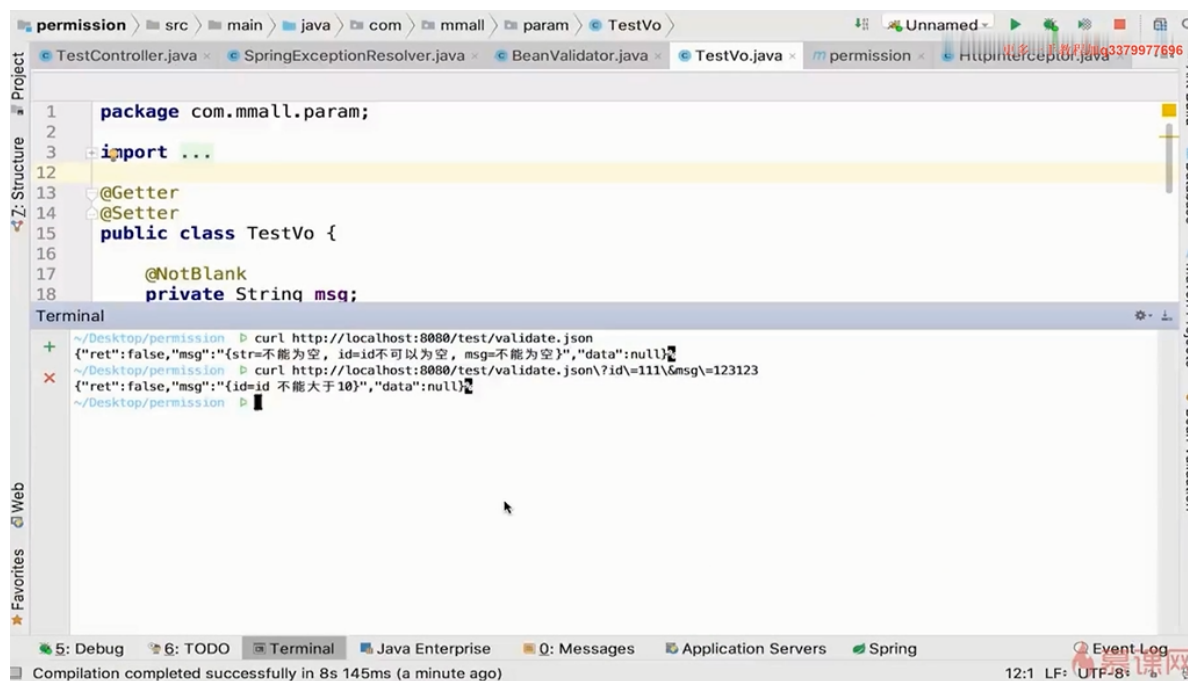
    @NotNull(message = "id 不可以为空")
    @Max(value = 10, message = "id 不能大于10")
    @Min(value = 0, message = "id 至少大于等于0")
    private Integer id;

    @NotBlank
    private String msg;

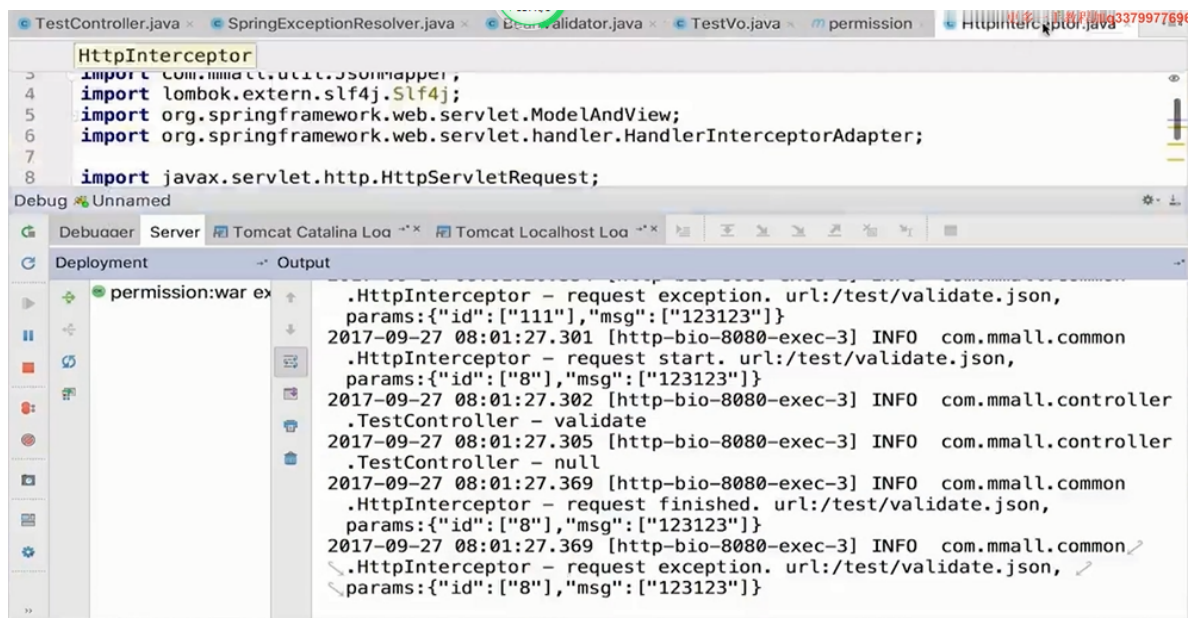
    private List<String> str;
}

```

curl [http://localhost:8080/permission\\_war/test/validate3.json?id=111&msg=123123](http://localhost:8080/permission_war/test/validate3.json?id=111&msg=123123)



curl [http://localhost:8080/permission\\_war/test/validate3.json?id=8&msg=123123](http://localhost:8080/permission_war/test/validate3.json?id=8&msg=123123)



## 修改HttpInterceptor

```
package com.trt.permission.common;

import com.trt.permission.util.JsonMapper;
import lombok.extern.slf4j.Slf4j;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.handler.HandlerInterceptorAdapter;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.Map;
/**
 * @version 1.0
 * @author: trt
 * @description: Http请求前后端监听工具类
 * @create: 2020-01-19 20:07
 */
@Slf4j
public class HttpInterceptor extends HandlerInterceptorAdapter {

    private static final String START_TIME = "requestStartTime";

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
        String url = request.getRequestURI().toString();
        Map parameterMap = request.getParameterMap();
        //log.info("request start. url:{}", url,
        JsonMapper.obj2String(parameterMap));
        long start = System.currentTimeMillis();
        request.setAttribute(START_TIME, start);
        return true;
    }

    /**
     * postHandle 和afterCompletion时间几乎一样
     * @param request
     * @param response
     */
}
```

```

    * @param handler
    * @param modelAndView
    * @throws Exception
    */
    @Override
    public void postHandle(HttpServletRequest request, HttpServletResponse
response, Object handler, ModelAndView modelAndView) throws Exception {
//        String url = request.getRequestURI().toString();
//        long start = (Long) request.getAttribute(START_TIME);
//        long end = System.currentTimeMillis();
//        log.info("request finished. url:{}, cost:{}", url, end - start);

    }

    @Override
    public void afterCompletion(HttpServletRequest request, HttpServletResponse
response, Object handler, Exception ex) throws Exception {
        String url = request.getRequestURI().toString();
        long start = (Long) request.getAttribute(START_TIME);
        long end = System.currentTimeMillis();
        //log.info("request completed. url:{}, cost:{}", url, end - start);

    }
}

```

## 通过GIT推送到码云上

```

65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git init
Reinitialized existing Git repository in
F:/WorkSpace/IdeaProject/permission/.git/
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        generator/
        init.sql
        pom.xml
        src/

nothing added to commit but untracked files present (use "git add" to track)

65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git add .
warning: LF will be replaced by CRLF in init.sql.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in
src/main/java/com/trt/permission/common/ApplicationContextHolder.java.
The file will have its original line endings in your working directory

```

warning: LF will be replaced by CRLF in  
src/main/java/com/trt/permission/common/HttpInterceptor.java.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in  
src/main/java/com/trt/permission/common/RequestHolder.java.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in  
src/main/java/com/trt/permission/util/JsonMapper.java.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in src/main/webapp/WEB-INF/web.xml.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in src/main/webapp/index.jsp.  
The file will have its original line endings in your working directory

65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)

\$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: .gitignore  
new file: generator/generator.xml  
new file: generator/mybatis-generator-core-1.3.2.jar  
new file: generator/mysql-connector-java-5.1.34.jar  
new file:

"generator/\347\224\237\346\210\220\350\257\255\345\217\245.txt"

new file: init.sql  
new file: pom.xml  
new file:

src/main/java/com/trt/permission/common/ApplicationContextHolder.java

new file: src/main/java/com/trt/permission/common/HttpInterceptor.java  
new file: src/main/java/com/trt/permission/common/JsonData.java  
new file: src/main/java/com/trt/permission/common/RequestHolder.java  
new file:

src/main/java/com/trt/permission/common/SpringExceptionHandler.java

new file:

src/main/java/com/trt/permission/controller/TestController.java

new file: src/main/java/com/trt/permission/dao/SysAclMapper.java  
new file: src/main/java/com/trt/permission/dao/SysAclModuleMapper.java  
new file: src/main/java/com/trt/permission/dao/SysDeptMapper.java  
new file: src/main/java/com/trt/permission/dao/SysLogMapper.java  
new file: src/main/java/com/trt/permission/dao/SysRoleAclMapper.java  
new file: src/main/java/com/trt/permission/dao/SysRoleMapper.java  
new file: src/main/java/com/trt/permission/dao/SysRoleUserMapper.java  
new file: src/main/java/com/trt/permission/dao/SysUserMapper.java  
new file: src/main/java/com/trt/permission/dao/TestDao.java  
new file:

src/main/java/com/trt/permission/exception/ParamException.java

new file:

src/main/java/com/trt/permission/exception/PermissionException.java

new file: src/main/java/com/trt/permission/model/SysAcl.java  
new file: src/main/java/com/trt/permission/model/SysAclModule.java  
new file: src/main/java/com/trt/permission/model/SysDept.java  
new file: src/main/java/com/trt/permission/model/SysLog.java  
new file: src/main/java/com/trt/permission/model/SysLogWithBLOBs.java  
new file: src/main/java/com/trt/permission/model/SysRole.java

```
new file:   src/main/java/com/trt/permission/model/SysRoleAcl.java
new file:   src/main/java/com/trt/permission/model/SysRoleUser.java
new file:   src/main/java/com/trt/permission/model/SysUser.java
new file:   src/main/java/com/trt/permission/param/TestVo.java
new file:   src/main/java/com/trt/permission/util/BeanValidator.java
new file:   src/main/java/com/trt/permission/util/JsonMapper.java
new file:   src/main/resources/applicationContxt.xml
new file:   src/main/resources/logback.xml
new file:   src/main/resources/mapper/SysAclMapper.xml
new file:   src/main/resources/mapper/SysAclModuleMapper.xml
new file:   src/main/resources/mapper/SysDeptMapper.xml
new file:   src/main/resources/mapper/SysLogMapper.xml
new file:   src/main/resources/mapper/SysRoleAclMapper.xml
new file:   src/main/resources/mapper/SysRoleMapper.xml
new file:   src/main/resources/mapper/SysRoleUserMapper.xml
new file:   src/main/resources/mapper/SysUserMapper.xml
new file:   src/main/resources/mapper/TestMapper.xml
new file:   src/main/resources/mybatis-config.xml
new file:   src/main/resources/settings.properties
new file:   src/main/webapp/WEB-INF/spring-servlet.xml
new file:   src/main/webapp/WEB-INF/views/exception.jsp
new file:   src/main/webapp/WEB-INF/web.xml
new file:   src/main/webapp/index.jsp
```

65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)

```
$ git commit -am 'project init commit'
```

[master (root-commit) 9b835b7] project init commit

53 files changed, 3387 insertions(+)

create mode 100644 .gitignore

create mode 100644 generator/generator.xml

create mode 100644 generator/mybatis-generator-core-1.3.2.jar

create mode 100644 generator/mysql-connector-java-5.1.34.jar

create mode 100644

"generator/\347\224\237\346\210\220\350\257\255\345\217\245.txt"

create mode 100644 init.sql

create mode 100644 pom.xml

create mode 100644

src/main/java/com/trt/permission/common/ApplicationContextHolder.java

create mode 100644 src/main/java/com/trt/permission/common/HttpInterceptor.java

create mode 100644 src/main/java/com/trt/permission/common/JsonData.java

create mode 100644 src/main/java/com/trt/permission/common/RequestHolder.java

create mode 100644

src/main/java/com/trt/permission/common/SpringExceptionHandler.java

create mode 100644

src/main/java/com/trt/permission/controller/TestController.java

create mode 100644 src/main/java/com/trt/permission/dao/SysAclMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysAclModuleMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysDeptMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysLogMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysRoleAclMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysRoleMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysRoleUserMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/SysUserMapper.java

create mode 100644 src/main/java/com/trt/permission/dao/TestDao.java

create mode 100644

src/main/java/com/trt/permission/exception/ParamException.java

```
create mode 100644 src/main/java/com/trt/permission/exception/PermissionException.java
create mode 100644 src/main/java/com/trt/permission/model/SysAcl.java
create mode 100644 src/main/java/com/trt/permission/model/SysAclModule.java
create mode 100644 src/main/java/com/trt/permission/model/SysDept.java
create mode 100644 src/main/java/com/trt/permission/model/SysLog.java
create mode 100644 src/main/java/com/trt/permission/model/SysLogWithBLOBS.java
create mode 100644 src/main/java/com/trt/permission/model/SysRole.java
create mode 100644 src/main/java/com/trt/permission/model/SysRoleAcl.java
create mode 100644 src/main/java/com/trt/permission/model/SysRoleUser.java
create mode 100644 src/main/java/com/trt/permission/model/SysUser.java
create mode 100644 src/main/java/com/trt/permission/param/TestVo.java
create mode 100644 src/main/java/com/trt/permission/util/BeanValidator.java
create mode 100644 src/main/java/com/trt/permission/util/JsonMapper.java
create mode 100644 src/main/resources/applicationContext.xml
create mode 100644 src/main/resources/logback.xml
create mode 100644 src/main/resources/mapper/SysAclMapper.xml
create mode 100644 src/main/resources/mapper/SysAclModuleMapper.xml
create mode 100644 src/main/resources/mapper/SysDeptMapper.xml
create mode 100644 src/main/resources/mapper/SysLogMapper.xml
create mode 100644 src/main/resources/mapper/SysRoleAclMapper.xml
create mode 100644 src/main/resources/mapper/SysRoleMapper.xml
create mode 100644 src/main/resources/mapper/SysRoleUserMapper.xml
create mode 100644 src/main/resources/mapper/SysUserMapper.xml
create mode 100644 src/main/resources/mapper/TestMapper.xml
create mode 100644 src/main/resources/mybatis-config.xml
create mode 100644 src/main/resources/settings.properties
create mode 100644 src/main/webapp/WEB-INF/spring-servlet.xml
create mode 100644 src/main/webapp/WEB-INF/views/exception.jsp
create mode 100644 src/main/webapp/WEB-INF/web.xml
create mode 100644 src/main/webapp/index.jsp
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git remote add origin git@gitee.com:ZiChenMuFeng/permission.git
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git branch
* master
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git push -u origin master
Warning: Permanently added the ECDSA host key for IP address '180.97.125.228' to
the list of known hosts.
To gitee.com:ZiChenMuFeng/permission.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@gitee.com:ZiChenMuFeng/permission.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git pull
warning: no common commits
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
```



```
remote: Total 4 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
From gitee.com:ZiChenMuFeng/permission
* [new branch]      master    -> origin/master
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.
```

```
git pull <remote> <branch>
```

If you wish to set tracking information for this branch you can do so with:

```
git branch --set-upstream-to=origin/<branch> master
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git push -u origin master
To gitee.com:ZiChenMuFeng/permission.git
 ! [rejected]      master -> master (non-fast-forward)
error: failed to push some refs to 'git@gitee.com:ZiChenMuFeng/permission.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git push -u -f origin master
Enumerating objects: 74, done.
Counting objects: 100% (74/74), done.
Delta compression using up to 4 threads
Compressing objects: 100% (67/67), done.
Writing objects: 100% (74/74), 1.34 MiB | 1.64 MiB/s, done.
Total 74 (delta 20), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-3.8]
To gitee.com:ZiChenMuFeng/permission.git
 + ed42ac7...9b835b7 master -> master (forced update)
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git branch
* master
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git branch -r
origin/master
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (master)
$ git checkout -b v1.0 origin/master
Switched to a new branch 'v1.0'
Branch 'v1.0' set up to track remote branch 'master' from 'origin'.
```

```
65190@DESKTOP-BVT5U6K MINGW64 /f/WorkSpace/IdeaProject/permission (v1.0)
```

```
$ git branch
  master
* v1.0

65190@DESKTOP-BVT5U6K MINGW64 /f/workSpace/IdeaProject/permission (v1.0)
$ git push origin HEAD -u
Total 0 (delta 0), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-3.8]
remote: Create a pull request for 'v1.0' on Gitee by visiting:
remote:
https://gitee.com/ZiChenMuFeng/permission/pull/new/ZiChenMuFeng:v1.0...ZiChenMuFeng:master
To gitee.com:ZiChenMuFeng/permission.git
 * [new branch]      HEAD -> v1.0
Branch 'v1.0' set up to track remote branch 'v1.0' from 'origin'.

65190@DESKTOP-BVT5U6K MINGW64 /f/workSpace/IdeaProject/permission (v1.0)
$
```