thymeleaf数据字典方言

## 1、配置文件类加载生成Bean PubCodeDialect

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | /\*\* | |  | \* 在Thymeleaf的自动配置前先将附加的Thymeleaf自定义方言注入进去 | |  | \*/ | |  | @Configuration | |  | @AutoConfigureBefore(ThymeleafAutoConfiguration.class) | |  | public class ThymeleafAdditionalDialect {  @ConditionalOnMissingBean注解判断是否执行初始化代码，即如果用户已经创建了bean，则相关的初始化代码不再执行 | |  | @Bean | |  | @ConditionalOnMissingBean | |  | public PubCodeDialect pubcodeDialect() { | |  | return new PubCodeDialect(); | |  | } | |  |  | |  | @Bean | |  | @ConditionalOnMissingBean | |  | public ShiroDialect shiroDialect() { | |  | return new ShiroDialect(); | |  | } | |  | } | |

## 2、PubCodeDialect 编写方言，继承抽象方言AbstractDialect，实现表达的对象方言IExpressionObjectDialect，主要是用来返回一个扩展的PubCodeExpressionFactory

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | /\*\* | |  | \* 公共代码表达式处理的Thymeleaf方言 | |  | \*/ | |  | public class PubCodeDialect extends AbstractDialect implements IExpressionObjectDialect { | |  | public static final String NAME = "PubCode Dialect"; | |  | private final IExpressionObjectFactory PUBCODE\_EXPRESSION\_OBJECT\_FACTORY = new PubCodeExpressionFactory(); | |  |  | |  | public PubCodeDialect(){ | |  | super(NAME); | |  | } | |  |  | |  | @Override | |  | public IExpressionObjectFactory getExpressionObjectFactory(){ | |  | return PUBCODE\_EXPRESSION\_OBJECT\_FACTORY; | |  | } | |  | } | |

## 3、PubCodeExpressionFactory 类

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | /\*\* | |  | \* 公共代码表达式创建的工厂类 | |  | \*/ | |  | public class PubCodeExpressionFactory implements IExpressionObjectFactory { | |  | private static final String PUBCODE\_VARIABLE\_NAME = "pubcode"; | |  | private static final Set<String> ALL\_EXPRESSION\_OBJECT\_NAMES = Collections.unmodifiableSet( | |  | new LinkedHashSet<>(Arrays.asList(PUBCODE\_VARIABLE\_NAME))); | |  |  | |  | @Override | |  | public Set<String> getAllExpressionObjectNames(){ | |  | return ALL\_EXPRESSION\_OBJECT\_NAMES; | |  | } | |  |  | |  | @Override | |  | public Object buildObject(IExpressionContext context, final String expressionObjectName) { | |  | if (PUBCODE\_VARIABLE\_NAME.equalsIgnoreCase(expressionObjectName)) { | |  | return new PubCodeExpression(); | |  | } | |  | return null; | |  | } | |  |  | |  | @Override | |  | public boolean isCacheable(String expressionObjectName) { | |  | return expressionObjectName != null && PUBCODE\_VARIABLE\_NAME.equalsIgnoreCase(expressionObjectName); | |  | } | |  | } | |

## 4、从工厂中取出

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | /\*\* | |  | \* 公共代码处理的表达式 | |  | \*/ | |  | public final class PubCodeExpression { | |  | public PubCodeExpression(){} | |  |  | |  | /\*\* | |  | \* 根据类型和代码获取该公共代码的值 | |  | \* @param type 类型 | |  | \* @param code 代码 | |  | \* @return | |  | \*/ | |  | public String codeName(String type, String code) { | |  | return PubCodeUtil.getName(type, code); | |  | } | |  | } | |

|  |
| --- |
| /\*\* |
|  | \* 公共代码处理类 |
|  | \*/ |
|  | public final class PubCodeUtil { |
|  |  |
|  | // 类型-性别 |
|  | public static final String TYPE\_GENDER = "gender"; |
|  | // 类型-民族 |
|  | public static final String TYPE\_ETHNIC = "ethnic"; |
|  | // 类型-政治面貌 |
|  | public static final String TYPE\_POLITICAL = "political"; |
|  | // 类型-学历 |
|  |  |
|  | private static boolean isInited = false; |
|  | private static Map<String, Map<String, String>> pubDicts; |
|  |  |
|  | //TODO: 锁机制 |
|  | private static void init(){ |
|  | pubDicts = new HashMap<String, Map<String, String>>(); |
|  | //性别 |
|  | pubDicts.put(TYPE\_GENDER, new HashMap<String,String>(){ |
|  | { |
|  | put("01","男"); |
|  | put("02","女"); |
|  | } |
|  | }); |
|  | //民族 |
|  | pubDicts.put(TYPE\_ETHNIC, new HashMap<String,String>(){ |
|  | { |
|  | put("01","汉族"); |
|  | put("02","蒙古族"); |
|  | put("03","回族"); |
|  | put("04","藏族"); |
|  | put("05","维吾尔族"); |
|  | put("06","苗族"); |
|  | } |
|  | }); |
|  | //政治面貌 |
|  | pubDicts.put(TYPE\_POLITICAL, new HashMap<String,String>(){ |
|  | { |
|  | put("10","共青团员"); |
|  | put("11","中共预备党员"); |
|  | put("12","中共党员"); |
|  | put("13","群众"); |
|  | } |
|  | }); |
|  | //学历层次 |
|  | pubDicts.put(TYPE\_EDUDEGREE, new HashMap<String,String>(){ |
|  | { |
|  | put("00","专科毕业生"); |
|  | put("01","本科生毕业"); |
|  | put("02","硕士生毕业"); |
|  | put("03","博士生毕业"); |
|  | } |
|  | }); |
|  | isInited = true; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | /\*\* |
|  | \* 获取指定类别的公共代码表 |
|  | \* @param type |
|  | \* @return |
|  | \*/ |
|  | public static Map<String, String> getDictMap(String type){ |
|  | if (!isInited) init(); |
|  | return pubDicts.getOrDefault(type, null); |
|  | } |
|  |  |
|  |  |
|  | /\*\* |
|  | \* 获取指定类别、指定代码的名称值 |
|  | \* @param type |
|  | \* @param code |
|  | \* @return |
|  | \*/ |
|  | public static String getName(String type, String code) { |
|  | if (!isInited) init(); |
|  | return pubDicts.get(type).getOrDefault(code,""); |
|  | } |
|  |  |
|  |  |
|  | } |

|  |
| --- |
| @ModelAttribute |
|  | public void getMap(Map<String, Object> map){ |
|  | map.put("allGender", PubCodeUtil.getDictMap("gender")); |
|  | map.put("allMajor", PubCodeUtil.getDictMap("major")); |
|  | map.put("allEthnics", PubCodeUtil.getDictMap("ethnic")); |
|  | map.put("allEduMode", PubCodeUtil.getDictMap("eduMode")); |
|  | map.put("allEduYear", PubCodeUtil.getDictMap("eduYear")); |
|  | map.put("allAcademy", PubCodeUtil.getDictMap("academy")); |
|  | map.put("allAuditStatus", PubCodeUtil.getDictMap("auditStatus")); |
|  | map.put("allEduDegree", PubCodeUtil.getDictMap("eduDegree")); |
|  | map.put("allPolitical", PubCodeUtil.getDictMap("political")); |
|  | map.put("allLanguage", PubCodeUtil.getDictMap("language")); |
|  | map.put("allEthnics", PubCodeUtil.getDictMap("ethnic")); |
|  |  |
|  | } |

## 5、前段thyme开始使用

<div th:text="${#pubcode.codeName('eduDegree',studentInfo.eduDegree)}"></div>