# 行政区依赖库

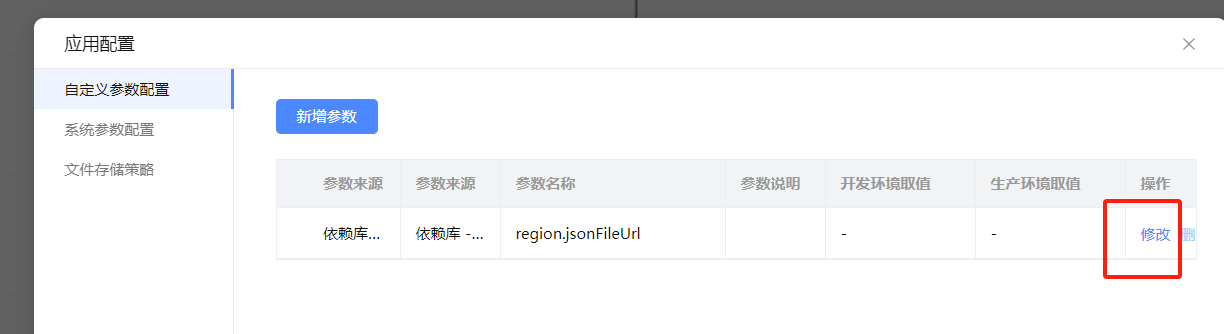
# 依赖库作用

本依赖库封装了一些常用的行政区方法

# 内置逻辑

|  |  |  |
| --- | --- | --- |
| 逻辑标识 | 逻辑名称 | 逻辑介绍 |
| getPrefectureCity | 提供输入省份，给出其地级市列表 | 入参：String 省份名，例如：浙江省  出参：List<Region>  citycode:城市编码，adcode:区域编码，name:行政区名称,zipCode:邮编,center:区域中心点,level:行政区划级别(country:国家,province:省份（直辖市会在province显示）,city:市（直辖市会在province显示）, district:区县,street:街道)  districts:下级行政区列表，包含district元素 |
| getCountry | 提供输入地级市，给出其区县列表 | 入参：String 地级市名，例如：杭州市  出参：List<Region>  citycode:城市编码，adcode:区域编码，name:行政区名称,zipCode:邮编,center:区域中心点,level:行政区划级别(country:国家,province:省份（直辖市会在province显示）,city:市（直辖市会在province显示）, district:区县,street:街道)  districts:下级行政区列表，包含district元素 |
| estimateProvinceAndCity | 提供输入省份和地级市,判断两者是否有所属关系 | 入参：String 省份名，例如：浙江省,String 地级市名，例如：杭州市  出参: boolean true代表存在所属关系 false代表不存在所属关系 |
| estimateCityAndCounty | 提供输入地级市和区县，判断两者是否有所属关系 | 入参：String 地级市名，例如：杭州市,String 区县名，例如：滨江区  出参: boolean true代表存在所属关系 false代表不存在所属关系 |
| estimateProvinceAndCounty | 提供输入省份和区县，判断两者是否有所属关系 | 入参：String 省份名，例如：浙江省,String 区县名，例如：滨江区  出参: boolean true代表存在所属关系 false代表不存在所属关系 |
| getRegion | 获取现有的行政区信息 | 出参：Region  citycode:城市编码，adcode:区域编码，name:行政区名称,zipCode:邮编,center:区域中心点,  level:行政区划级别(country:国家,province:省份（直辖市会在province显示）,city:市（直辖市会在province显示）, district:区县,street:街道)  districts:下级行政区列表，包含district元素 |

如果要更新行政区数据可以先调用getRegion(),获取现有的行政区信息，在此基础上修改，然后在应用可视化中心点击更多-》应用配置填入 region.jsonFileUrl 的值，依赖库会自动从指定的url中获取最新行政区的json信息

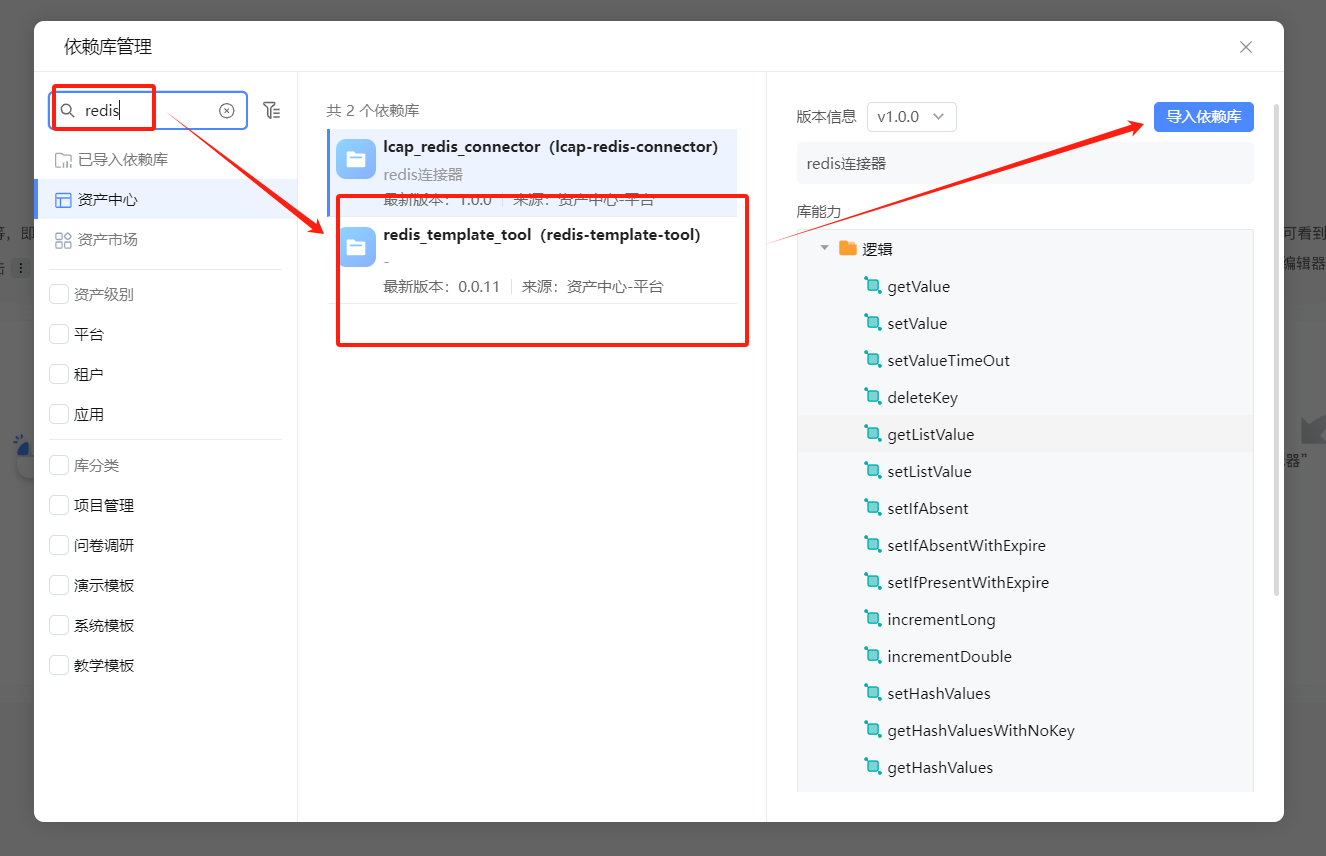


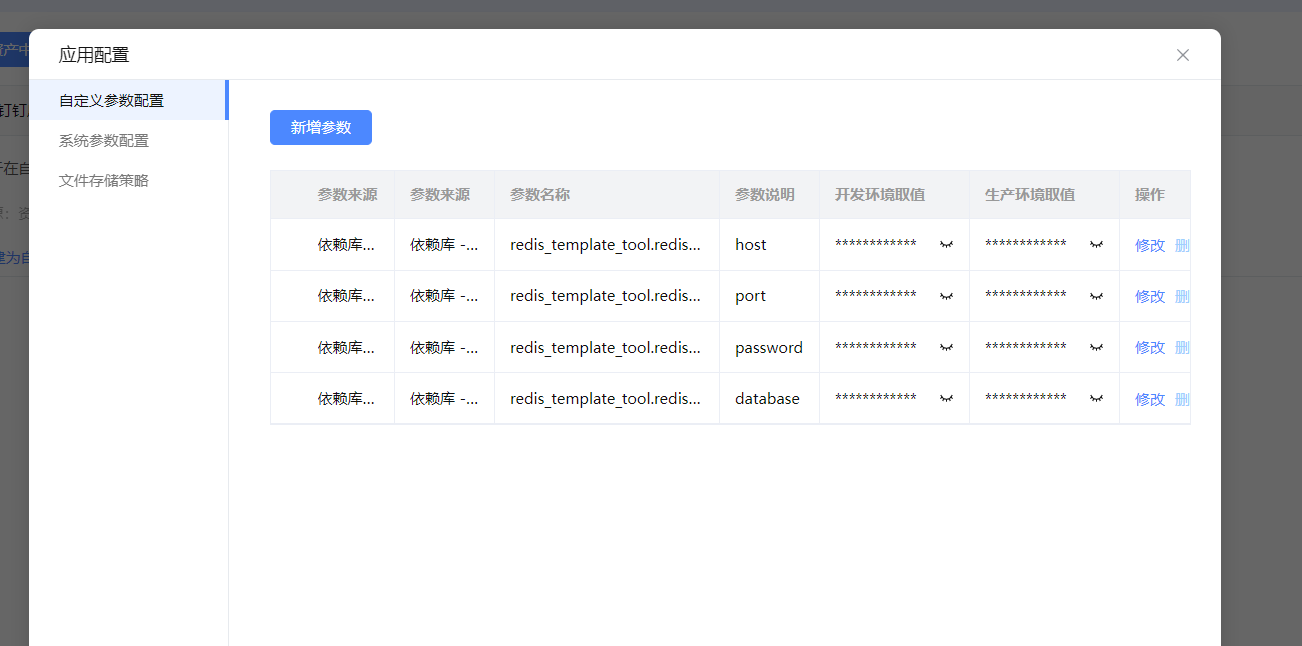


# 如何使用依赖库

本依赖库依赖RedisTemplate,在使用本依赖库之前需要在平台端先引入redis的依赖包并配置好。以下为操作示例

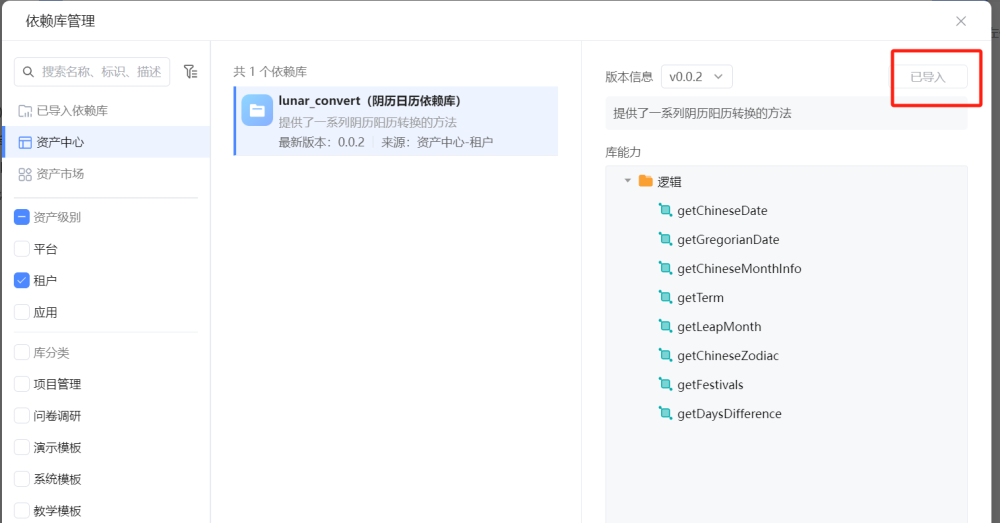
1. 在应用可视化中心界面，点击右上角更多依赖库管理，找到平台提供的redis\_template\_tool依赖库进行导入



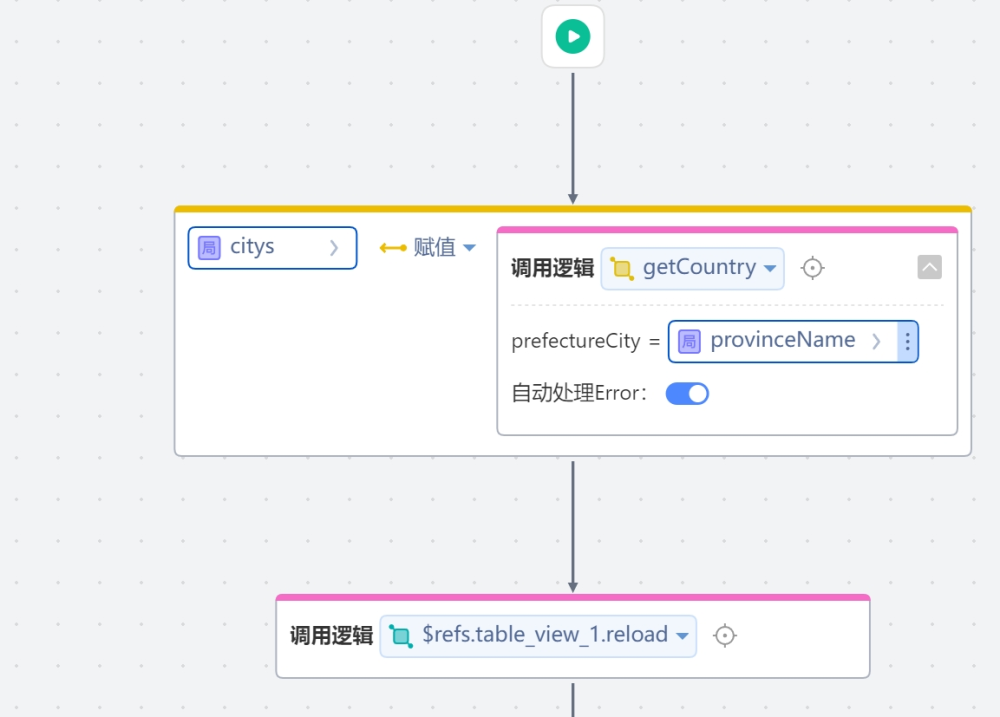
1. 导入成功后，点击更多-》应用集成下面会出现redis的配置项，根据实际需要进行填入参数值
2. 
3. 然后在可视化中心界面，点击右上角“更多”-》依赖库管理-》导入本依赖库



1. 在依赖库管理下面点击资产中心，找到阳历日历依赖库，点击右上角进行导入



1. 然后新建逻辑，在逻辑中调用依赖库中定义的逻辑，并按要求传入相应参数。



1. 发布应用后，点击触发相应的逻辑调用事件查看调用结果