**1.背景**

Redis是一种基于内存的功能强大的高性能键值存储系统，通过存储在内存中的数据，提供了快速、灵活和可靠的数据访问能力，常用于缓存、会话管理、消息队列等场景。它具有以下特点：

○ 高性能：Redis将数据存储在内存中，读写速度非常快，适用于高并发场景。

○ 数据结构丰富：Redis支持多种数据结构，包括字符串、哈希表、列表、集合、有序集合等，可以满足不同场景的需求。

○ 持久化支持：Redis支持数据的持久化存储，可以将数据保存到磁盘上，以防止数据丢失。

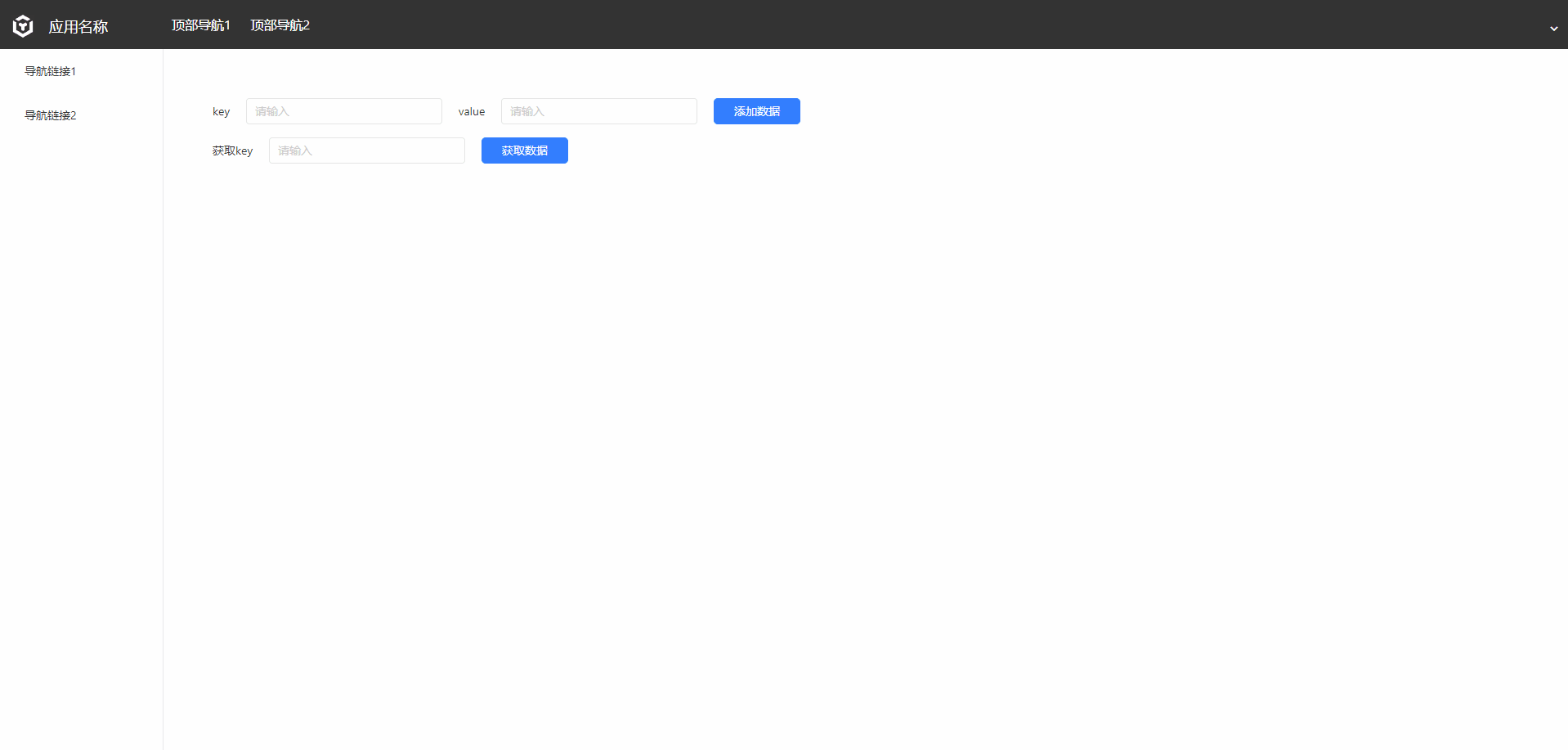
○ 分布式支持：Redis提供了主从复制、哨兵和集群等机制，可以实现高可用和数据分片。

○ 多语言支持：Redis提供了多种语言的客户端库，方便在不同的编程语言中使用。

因此，Redis被广泛应用于各种互联网应用中，用于提高系统性能、解决高并发问题、加速数据访问等。

**注：**业务若需要在redis基础依赖库的基础上，开发自己的业务信息，则需要另外开发Redis业务依赖库。具体请参考文档中心 [文档中心-CodeWave智能开发平台 (163.com)](https://community.codewave.163.com/CommunityParent/fileIndex?filePath=40.%E6%89%A9%E5%B1%95%E4%B8%8E%E9%9B%86%E6%88%90%2F10.%E6%89%A9%E5%B1%95%E5%BC%80%E5%8F%91%E6%96%B9%E5%BC%8F%2F30.%E6%9C%8D%E5%8A%A1%E7%AB%AF%E6%89%A9%E5%B1%95%E5%BC%80%E5%8F%91%2F10.%E4%BE%9D%E8%B5%96%E5%BA%93%E5%BC%80%E5%8F%91%2F25.%E4%BE%9D%E8%B5%96%E5%BA%93%E5%BC%80%E5%8F%91%E8%BF%9B%E9%98%B6%E6%95%99%E7%A8%8B-Redis%E4%B8%9A%E5%8A%A1%E4%BA%8C%E6%AC%A1%E5%BC%80%E5%8F%91%E6%A1%88%E4%BE%8B.md&version=3.6)

示例：



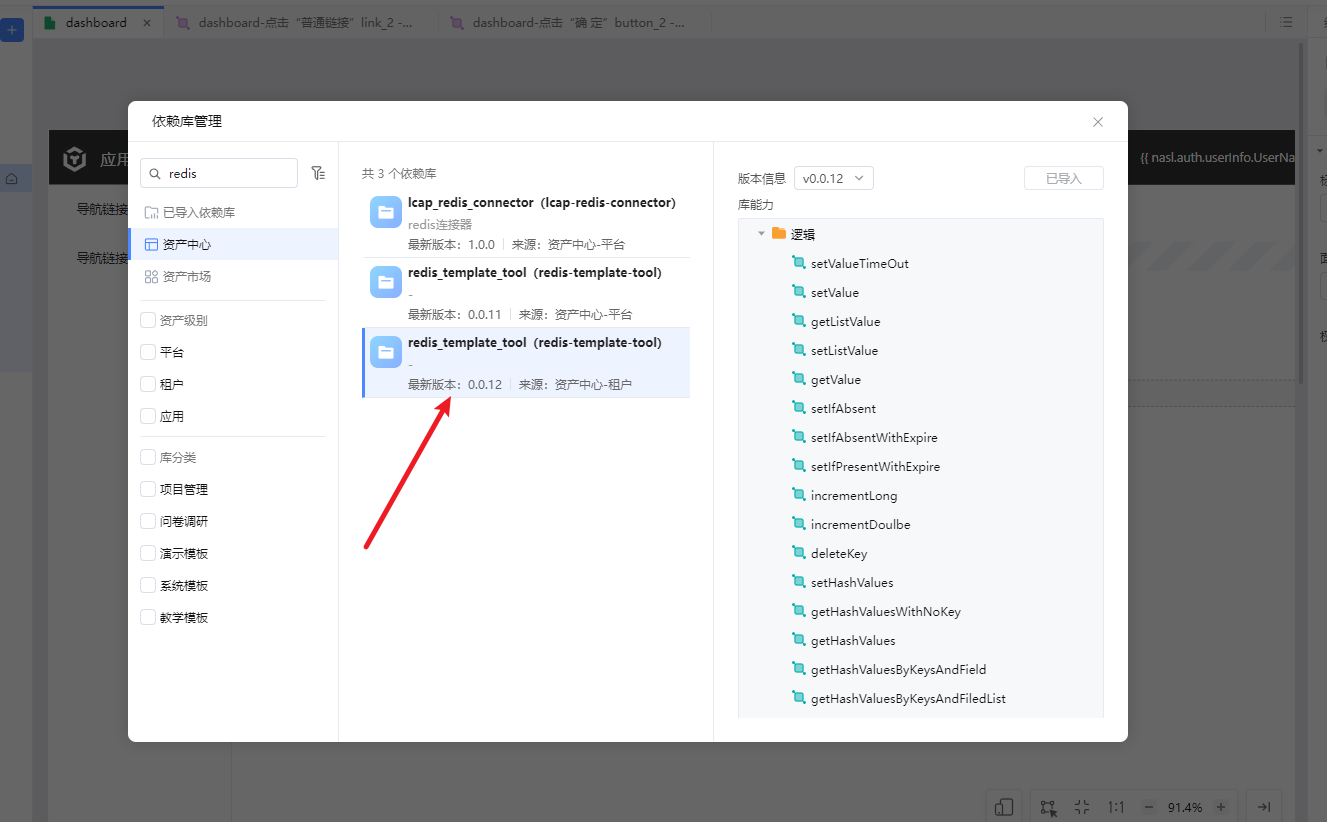
**2.常用术语**

|  |  |
| --- | --- |
| Key | redis中的键，数据是通过Key来进行存储和访问的，每个Key都是一个唯一的标识符。 |
| Value | 值，与每个Key相关联的数据 |
| 数据结构 | Redis支持多种数据结构，包括字符串（String）、哈希表（Hash）、列表（List）、集合（Set）、有序集合（Sorted Set）等 |
| TTL（Time To Live） | 表示Key的生存时间，可以通过设置TTL来实现Key的自动过期。 |

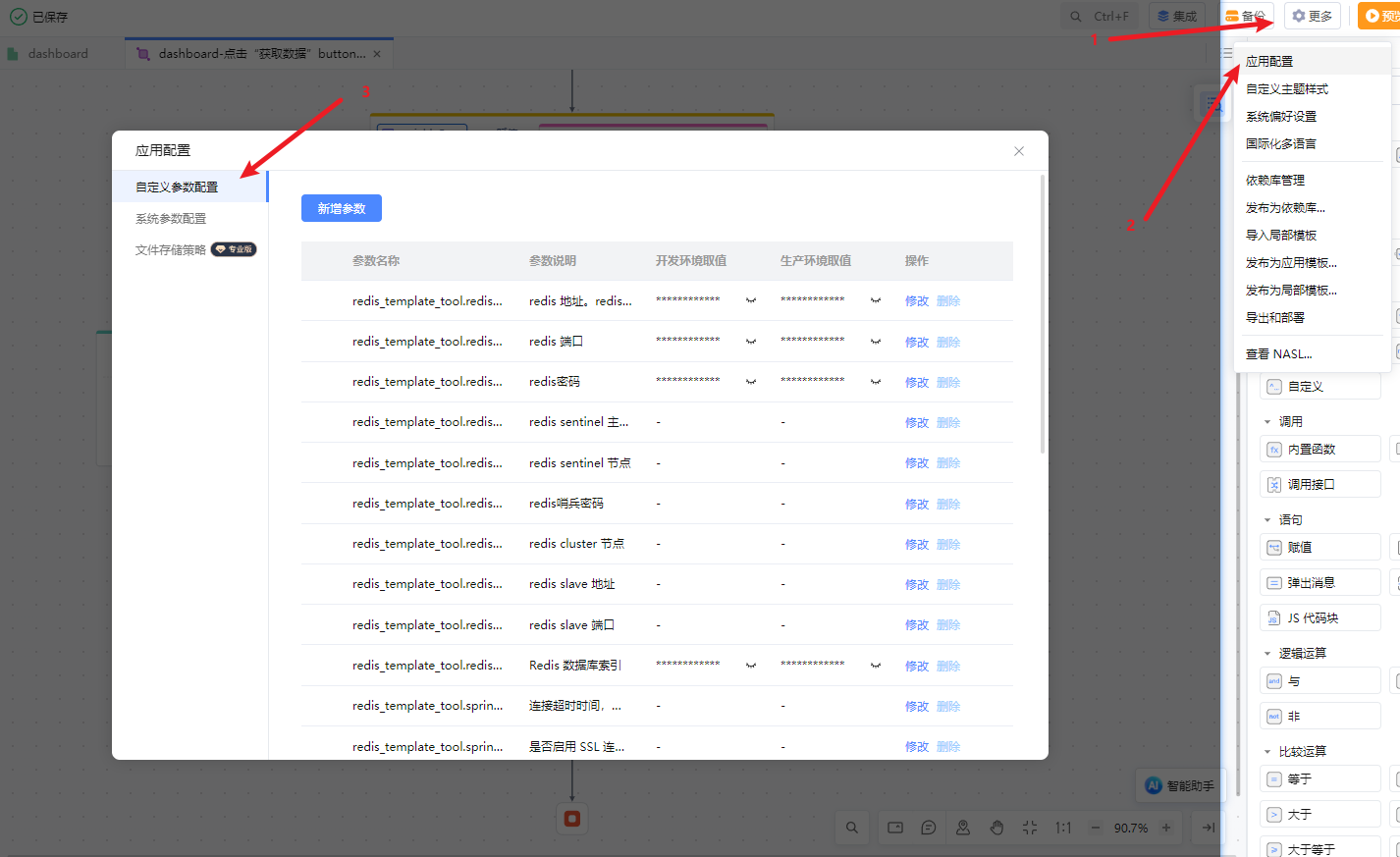
**3.示例**

1）引入依赖

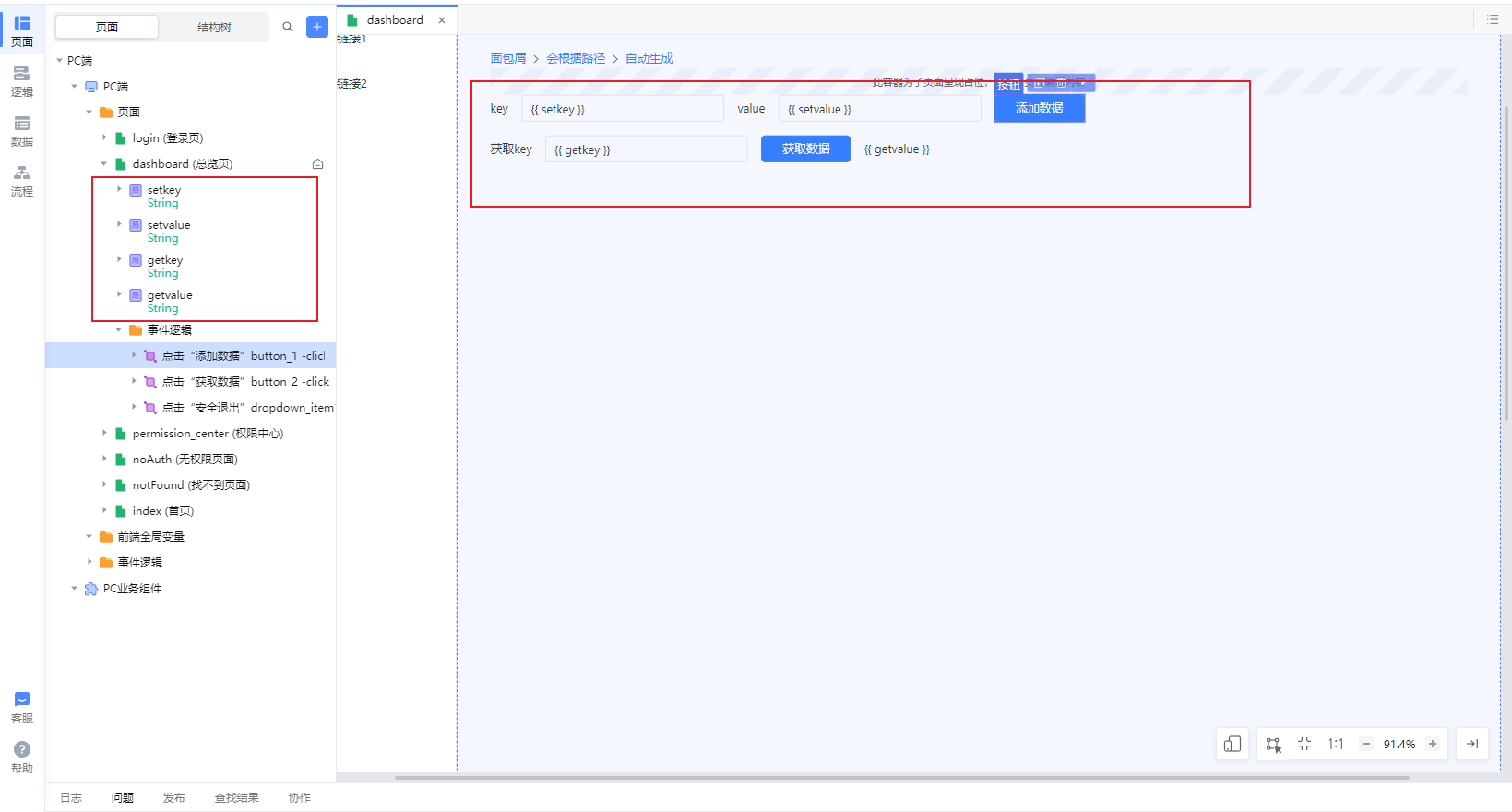
在依赖库管理导入redis依赖库。



2）在更多中选择应用配置，在自定义参数配置中将对应的redis信息填入

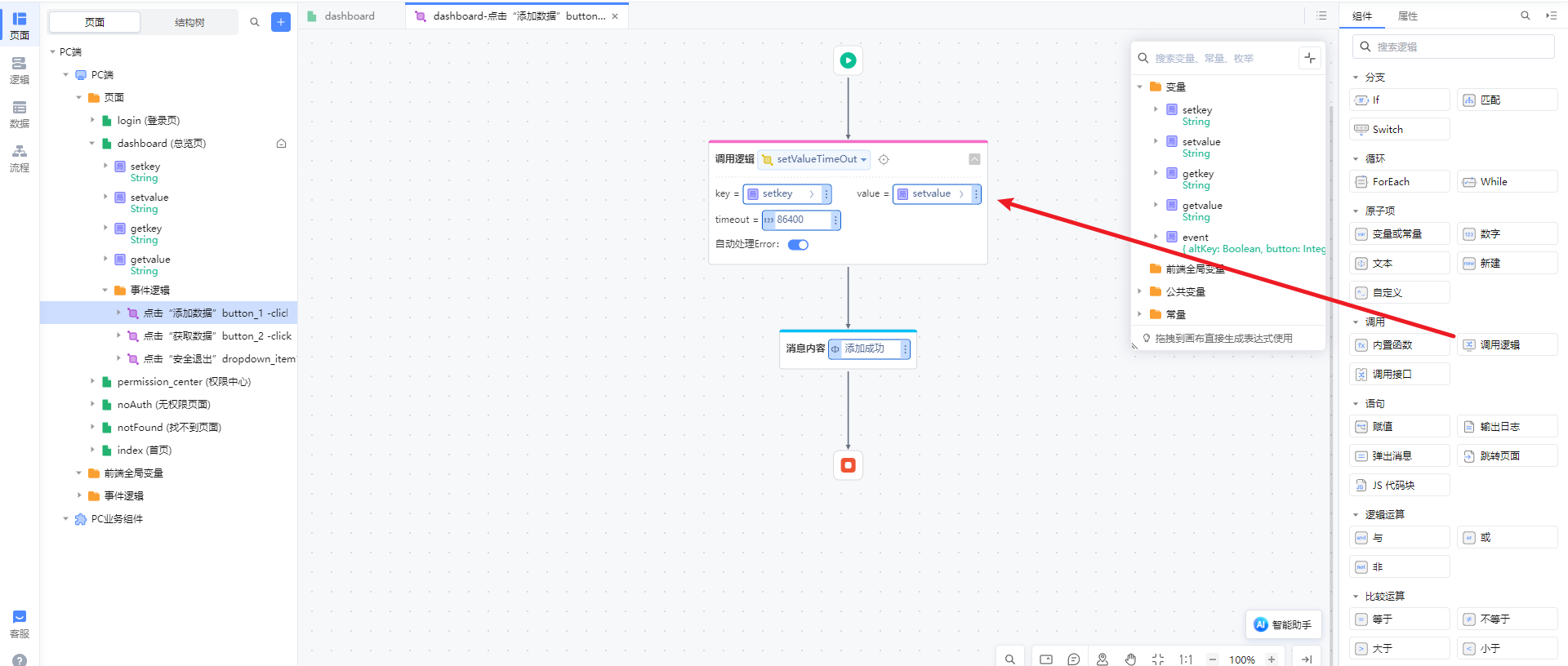


3）在页面下创建四个String类型变量，并搭建一个场景对redis进行操作

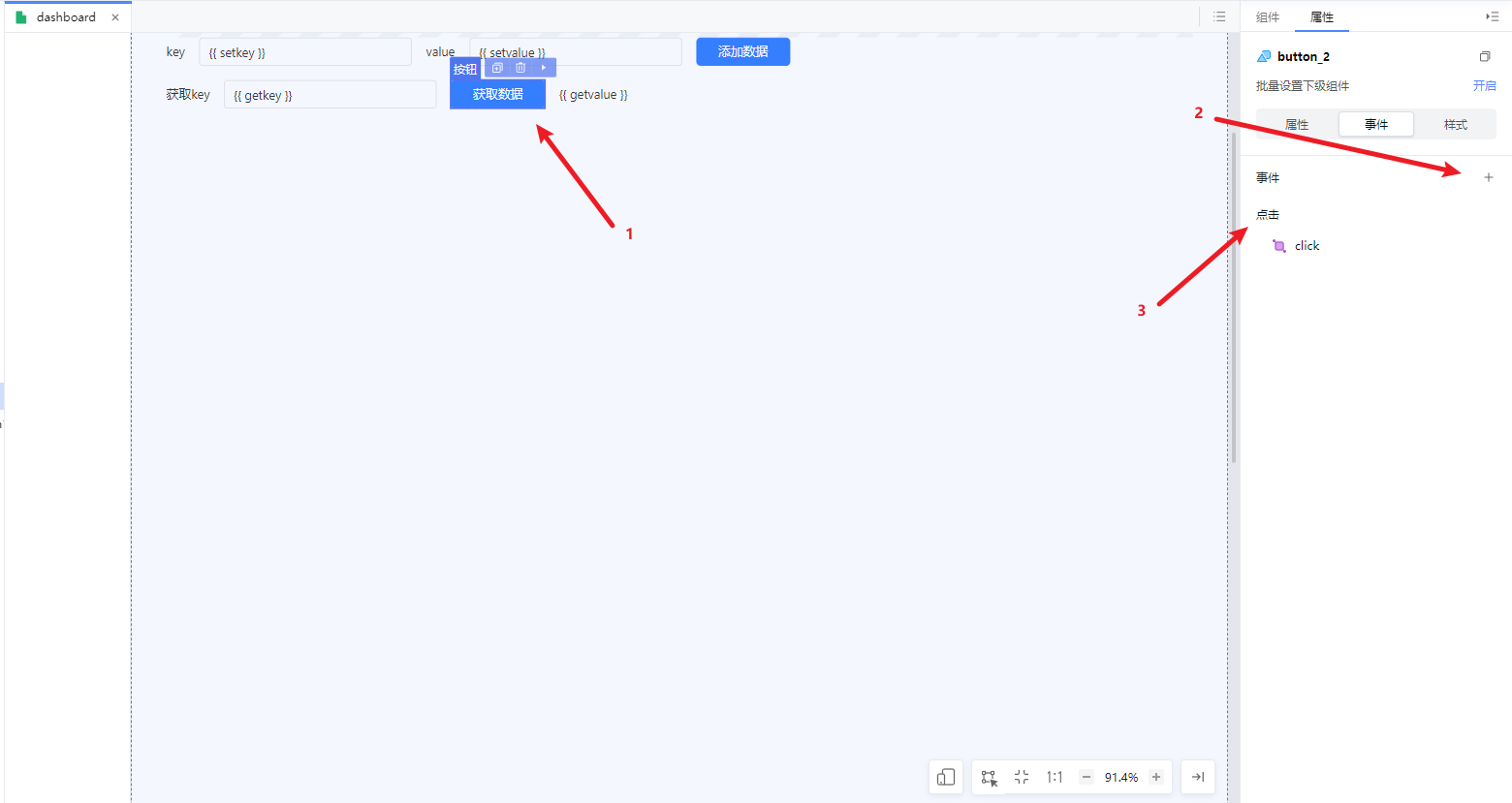


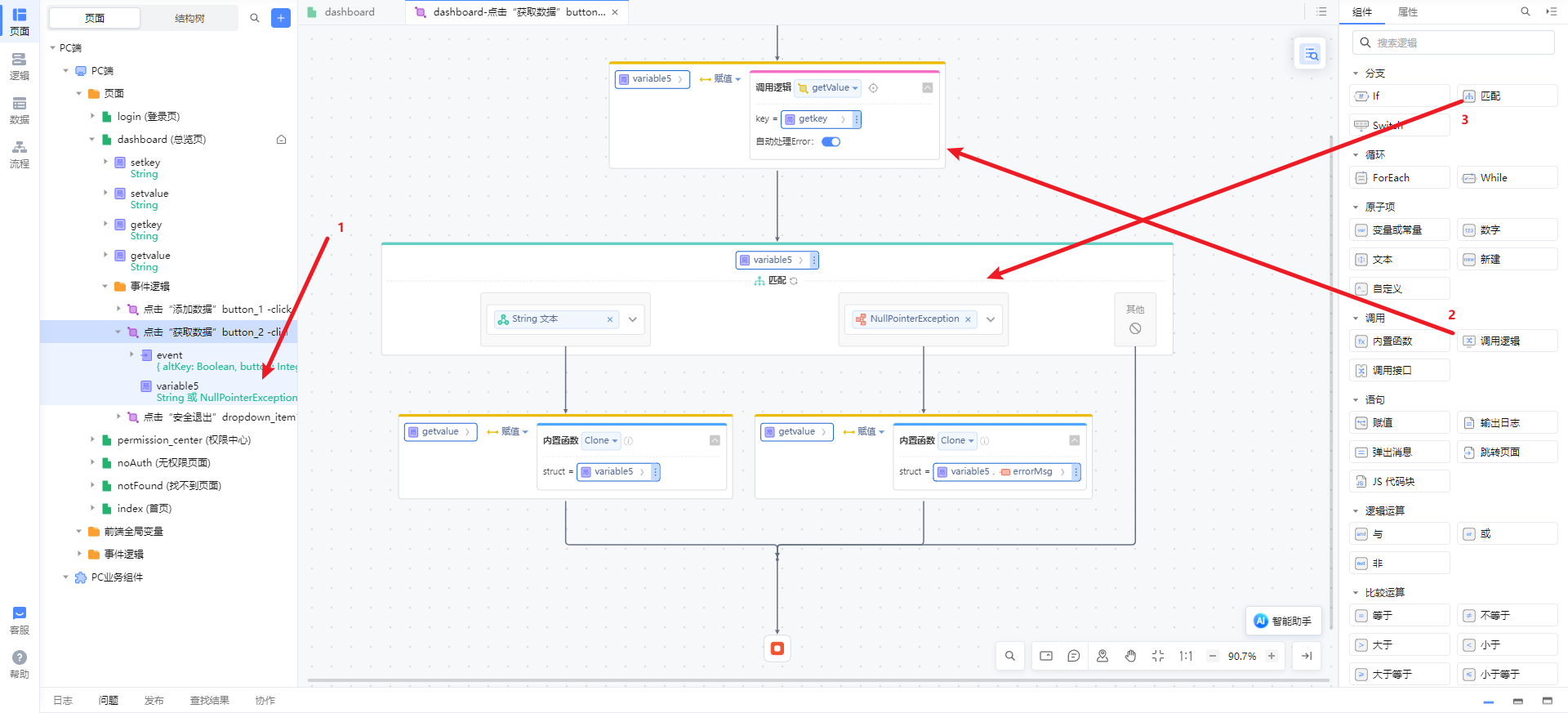
4）给添加数据按钮添加一个点击事件，在事件中调用依赖库中提供的添加逻辑，然后填充对应的属性





5）给获取数据按钮添加一个点击事件，在事件下添加一个变量，并调用依赖库中提供的查询逻辑，然后填入对应的key，并通过匹配对不同的返回类型进行处理，正常返回直接赋值即可，如类型不为String则可以将其中的错误信息进行赋值查看





6）将数据填好后添加并进行获取即可查看最终效果，如下图

