5、训练营期间：把BeanDefinition保存的配置信息映射到Bean的属性中

2020/11/24 12:05

1、数据绑定

有了配置信息的载体——资源Resource之后，还需要把配置信息映射到Bean中，这就是数据绑定。（详见《Spring数据绑定》）

配置信息的本质就是键值对，从资源中读取后，保存在BeanDefinition中，对应代码中的PropertyValue对象。可以从BeanDefinition中获取PropertyValue对象。

数据绑定不仅用于从 BeanDefinition 到 Bean 实例创建，还有以下的两个应用

- Spring 数据绑定（DataBinder）
- Spring Web 参数绑定（WebDataBinder）

1.1、绑定组件

- 标准组件
  - org.springframework.validation.DataBinder
- Web 组件
  - org.springframework.web.bind.WebDataBinder
  - org.springframework.web.bind.ServletRequestDataBinder
  - org.springframework.web.bind.support.WebRequestDataBinder
  - org.springframework.web.bind.support.WebExchangeDataBinder（since 5.0）

1.2、绑定过程

就是一个使用bind(PropertyValues)：将 PropertyValues Key-Value 内容映射到关联 Bean（target）中的属性上的过程。

假设 PropertyValues 中包含“name = 小马哥”的键值对，同时 Bean 对象 User 中存在name 属性，当 bind 方法执行时，User 对象中的 name 属性值将被绑定为“小马哥”。

1.3、DataBinder 元数据

特征	说明
数据来源	BeanDefinition，主要来源 XML 资源配置 BeanDefinition
数据结构	由一个或多个 PropertyValue 组成
成员结构	PropertyValue 包含属性名称，以及属性值（包括原始值、类型转换后的值）
常见实现	MutablePropertyValues
Web 扩展实现	ServletConfigPropertyValues、ServletRequestParameterPropertyValues
相关生命周期	InstantiationAwareBeanPostProcessor#postProcessProperties

InstantiationAwareBeanPostProcessor#postProcessProperties方法是在应用PropertyValues之前的一个扩展点。

可以从BeanDefinition获取到PropertyValues： PropertyValues#getPropertyValues->MutablePropertyValues。

MutablePropertyValues是用List<PropertyValue>实现的。

1.4、DataBinder的使用

DataBinder就是BEAN和属性之间的桥梁，创建DataBinder实例的时候先关联一个BEAN类型，然后从MAP创建一个属性源，最后根据这个属性源创建PropertyValue，再把PropertyValue这个数据源绑定到DataBinder上，从而完成“胶合”的功能。

```
// 创建空白对象
User user = new User();
// 1. 创建 DataBinder
DataBinder binder = new DataBinder(user, "");
// 2. 创建 PropertyValues
Map<String, Object> source = new HashMap<>();
source.put("id", 1);
source.put("name", "小马哥");
PropertyValues propertyValues = new MutablePropertyValues(source);
binder.bind(propertyValues);
// 3. 输出 User 内容
System.out.println(user);
```

- PropertyValues 存在 User 中不存在属性值时会被忽略；
- DataBinder 支持嵌套属性

```
// b. PropertyValues 存在一个嵌套属性，比如 company.name
source.put("company.name", "geekbang");
//相当于框架作了以下动作
// DataBinder 特性二：支持嵌套属性
// Company company = new Company();
// company.setName("geekbang");
// user.setCompany(compay)
```