**Interface BeanFactoryPostProcessor**

Allows for custom modification of an application context's bean definitions, adapting the bean property values of the context's underlying bean factory.

Application contexts can auto-detect **BeanFactoryPostProcessor** beans in their bean definitions and apply them before any other beans get created.

void **postProcessBeanFactory**([ConfigurableListableBeanFactory](http://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/beans/factory/config/ConfigurableListableBeanFactory.html) beanFactory)

throws [BeansException](http://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/beans/BeansException.html)

**Parameters:**

**beanFactory** - the bean factory used by the application context

Modify the application context's internal bean factory after its standard initialization. All bean definitions will have been loaded, but no beans will have been instantiated yet. This allows for overriding or adding properties even to eager-initializing beans.

A BeanFactoryPostProcessor may interact with and modify bean definitions, but never bean instances. Doing so may cause premature bean instantiation, violating the container and causing unintended side-effects. If bean instance interaction is required, consider implementing BeanPostProcessor instead.

**BeanPostProcessor** instead, which **alters the bean instance** rather than **the bean definition**.

I'm using a BeanPostProcessor, and rather than replacing the original MessageSource in the application context, I just add my own as its parent:

@Bean

BeanPostProcessor messageSourceCustomExtender() {

return new BeanPostProcessor() {

@Override

public Object **postProcessBeforeInitialization**(Object bean, String beanName) throws BeansException {

return bean;

}

@Override

public Object **postProcessAfterInitialization**(Object bean, String beanName) throws BeansException {

if (bean instanceof HierarchicalMessageSource && beanName.equals("messageSource")) {

ResourceBundleMessageSource parent = new ResourceBundleMessageSource();

parent.setBasename("custom");

((HierarchicalMessageSource) bean).setParentMessageSource(parent);

}

return bean;

}

};

}