



链滴

# TinySpring 构建过程 Step6-ApplicationContext

作者: [Arthur](#)

原文链接: <https://ld246.com/article/1519712774620>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

其实BeanFactory已经提供了完整的功能，但是使用起来比较麻烦。现在用ApplicationContext做一接口封装。

## 1. ApplicationContext接口

```
public interface ApplicationContext extends BeanFactory {  
}
```

## 2. AbstractApplicationContext抽象类

```
public abstract class AbstractApplicationContext implements ApplicationContext {  
    protected AbstractBeanFactory beanFactory;  
  
    public AbstractApplicationContext(AbstractBeanFactory beanFactory) {  
        this.beanFactory = beanFactory;  
    }  
  
    public void refresh() throws Exception{  
    }  
  
    @Override  
    public Object getBean(String name) throws Exception {  
        return beanFactory.getBean(name);  
    }  
}
```

## 3. ClassPathXmlApplicationContextxml的ApplicationContext, refresh()来初始化Bean

```
public class ClassPathXmlApplicationContext extends AbstractApplicationContext {  
  
    private String configLocation;  
  
    public ClassPathXmlApplicationContext(String configLocation) throws Exception {  
        this(configLocation, new AutowireCapableBeanFactory());  
    }  
  
    public ClassPathXmlApplicationContext(String configLocation, AbstractBeanFactory beanFactory) throws Exception {  
        super(beanFactory);  
        this.configLocation = configLocation;  
        refresh();  
    }  
  
    @Override  
    public void refresh() throws Exception {  
        XmlBeanDefinitionReader xmlBeanDefinitionReader = new XmlBeanDefinitionReader(new ResourceLoader());  
        xmlBeanDefinitionReader.loadBeanDefinitions(configLocation);  
    }  
}
```

```
        for (Map.Entry<String, BeanDefinition> beanDefinitionEntry : xmlBeanDefinitionReader.  
etRegistry().entrySet()) {  
            beanFactory.registerBeanDefinition(beanDefinitionEntry.getKey(), beanDefinitionEntry  
getValue());  
        }  
    }  
}
```

#### 4. 测试代码，比之前使用更佳简便，封装了实现细节。

```
@Test  
public void test() throws Exception {  
    ApplicationContext applicationContext = new ClassPathXmlApplicationContext("tinyioc.xml  
);  
    HelloWorldService helloWorldService = (HelloWorldService) applicationContext.getBean("h  
lloWorldService");  
    helloWorldService.helloWorld();  
}
```