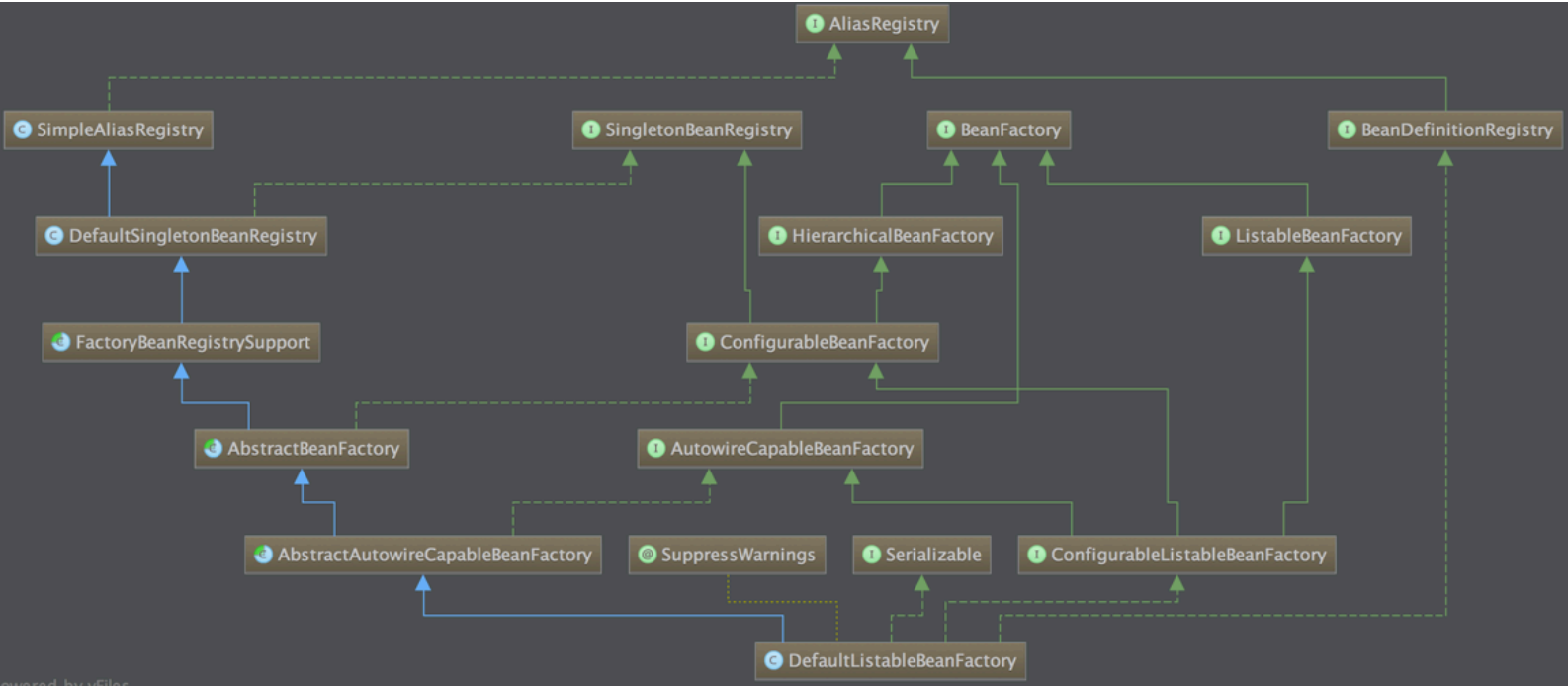


《Spring源码学习一：源码分析概述》中提到在Spring容器启动的过程中，会将类解析成Spring内部的BeanDefinition结构，并将BeanDefinition存储到一个叫DefaultListableBeanFactory中，本篇将深入分析BeanFactory。

首先，看一下DefaultListableBeanFactory的类图，



可以看到DefaultListableBeanFactory间接实现了BeanFactory接口， 先来分析一下这个BeanFactory， 直接查看BeanFactory源代码，

```
public interface BeanFactory {
    String FACTORY_BEAN_PREFIX = "&";

    Object getBean(String name) throws BeansException;

    <T> T getBean(String name, Class<T> requiredType) throws BeansException;

    <T> T getBean(Class<T> requiredType) throws BeansException;

    Object getBean(String name, Object... args) throws BeansException;

    <T> T getBean(Class<T> requiredType, Object... args) throws BeansException;

    boolean containsBean(String name);

    boolean isSingleton(String name) throws NoSuchBeanDefinitionException;

    boolean isPrototype(String name) throws NoSuchBeanDefinitionException;

    boolean isTypeMatch(String name, ResolvableType typeToMatch) throws
NoSuchBeanDefinitionException;

    boolean isTypeMatch(String name, Class<?> typeToMatch) throws
NoSuchBeanDefinitionException;

    Class<?> getType(String name) throws NoSuchBeanDefinitionException;

    String[] getAliases(String name);
}
```

从接口中的方法名可以很容易的理解每个方法的意图， 最常用或者最常见的就是getBean方法，用于获取Bean的实例。BeanFactory是用于访问Spring Bean容器的根接口，是一个单纯的Bean工厂，也就是常说的IOC容器的顶层定义，各种IOC容器是在其基础上为了满足不同需求而扩展的，包括经常使用的ApplicationContext。

DefaultListableBeanFactory是整个Bean加载的核心部分，是Spring注册及加载Bean的默认实现。下面列出了DefaultListableBeanFactory源代码中两个重要的属性，

```
/** Map of bean definition objects, keyed by bean name */
private final Map<String, BeanDefinition> beanDefinitionMap = new
ConcurrentHashMap<String, BeanDefinition>(256);
```

```
/** List of bean definition names, in registration order */
private volatile List<String> beanDefinitionNames = new ArrayList<String>(256);
```

其中，bean的定义被解析成BeanDefinition，同时解析得到beanName，将beanName和BeanDefinition存储到beanDefinitionMap中，同时会将beanName存储到beanDefinitionNames中。DefaultListableBeanFactory间接继承DefaultSingletonBeanRegistry，DefaultSingletonBeanRegistry中有如下属性，

```
/** Cache of singleton objects: bean name --> bean instance */
private final Map<String, Object> singletonObjects = new ConcurrentHashMap<String,
Object>(256);
```

singletonObjects用于存储单例bean的实例，getBean方法就是从这个Map里取实例对象的。