1.spring.factories加载：

ApplicationContextInitializer

ApplicationListener

getSpringFactoriesInstances(ApplicationContextInitializer.**class**)

getSpringFactoriesInstances(ApplicationListener.**class**)

该函数加载SpringFactories的配置类，通过setInitializers和setListeners将配置类对象保存在list中。

setInitializers(**this**.getSpringFactoriesInstances(ApplicationContextInitializer.**class**));  
setListeners(**this**.getSpringFactoriesInstances(ApplicationListener.**class**));

C:\Users\63474\.m2\repository\org\springframework\boot\spring-boot\2.2.5.RELEASE\spring-boot-2.2.5.RELEASE.jar!\META-INF\spring.factories

*# Application Context Initializers***org.springframework.context.ApplicationContextInitializer**=**\  
org.springframework.boot.context.ConfigurationWarningsApplicationContextInitializer,\  
org.springframework.boot.context.ContextIdApplicationContextInitializer,\  
org.springframework.boot.context.config.DelegatingApplicationContextInitializer,\  
org.springframework.boot.rsocket.context.RSocketPortInfoApplicationContextInitializer,\  
org.springframework.boot.web.context.ServerPortInfoApplicationContextInitializer**

C:\Users\63474\.m2\repository\org\springframework\boot\spring-boot-autoconfigure\2.2.5.RELEASE\spring-boot-autoconfigure-2.2.5.RELEASE.jar!\META-INF\spring.factories

# Initializers**org.springframework.context.ApplicationContextInitializer**=**\  
org.springframework.boot.autoconfigure.SharedMetadataReaderFactoryContextInitializer,\  
org.springframework.boot.autoconfigure.logging.ConditionEvaluationReportLoggingListener**

2.从堆栈中找到有main函数的clas，保存到mainApplicationClass

**this.mainApplicationClass = this.deduceMainApplicationClass();**

3.run

StopWatch记录开始的时间和taskname

**StopWatch stopWatch = new StopWatch();  
stopWatch.start();**

java.awt.headless属性设置为true

**this**.configureHeadlessProperty();

从SpringFactories加载SpringApplicationRunListener

*# Run Listeners***org.springframework.boot.SpringApplicationRunListener**=**\  
org.springframework.boot.context.event.EventPublishingRunListener**

**private** SpringApplicationRunListeners getRunListeners(String[] args) {  
 Class<?>[] types = **new** Class[]{SpringApplication.**class**, String[].**class**};  
 **return new** SpringApplicationRunListeners(logger, **this**.getSpringFactoriesInstances(SpringApplicationRunListener.**class**, types, **this**, args));  
}

EventPublishingRunListener starting

SpringApplicationRunListeners listeners = this.getRunListeners(args);  
listeners.starting();

EventPublishingRunListener.java

this.initialMulticaster = new SimpleApplicationEventMulticaster();

public void starting() {  
 this.initialMulticaster.multicastEvent(new ApplicationStartingEvent(this.application, this.args));  
}

创建 context实例：

org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext

context = createApplicationContext();

prepareContext(context, environment, listeners, applicationArguments, printedBanner);  
refreshContext(context);  
afterRefresh(context, applicationArguments);

prepareContext

使用context为Initializers和listeners初始化.

将DefaultApplicationArguments对象注入到context的DefaultSingletonBeanRegistry容器中。

Load the sources。