springboot 内嵌tomcat，及如何将DispatcherServlet注入到内嵌Tomcat流程

1：项目启动步骤

SpringApplication.java

295行：SpringApplication.run 整个项目启动入口

585行：SpringApplication.createApplicationContext 创建SpringIoC容器，默认使用**AnnotationConfigServletWebServerApplicationContext**

314行：SpringApplication.prepareContext 容器前置处理

316行：SpringApplication.refreshContext 开始刷新容器

324行：SpringApplication.callRunners 开始调用实现了ApplicationRunner

接口和CommandLineRunner接口的实现类

2：容器刷新步骤

ServletWebServerApplicationContext.java

140行：ServletWebServerApplicationContext.refresh 调用父类refresh方法开始刷新容器，导入ServletWebServerFactoryConfiguration，该配置类会往springioc 容器中注入

TomcatServletWebServerFactory，JettyServletWebServerFactory，UndertowServletWebServerFactory 这个三个工厂类，默认是TomcatServletWebServerFactory

151行：ServletWebServerApplicationContext.onRefresh 开始创建webServer也就是web容器

201行：ServletWebServerApplicationContext.getWebServerFactory 获取webserver工厂，具体包括3个实现TomcatServletWebServerFactory，JettyServletWebServerFactory，UndertowServletWebServerFactory，默认使用tomcat，如果要切换，需要在pom文件去引入相关依赖

3：Tomcat容器创建步骤

TomcatServletWebServerFactory.java

159行：TomcatServletWebServerFactory.getWebServer 创建Tomcat容器，并启动Tomcat容器，

328行：TomcatServletWebServerFactory.configureContext 创建ServletContainerInitializer接口实例TomcatStarter并将其加入到Tomcat容器中，该类负责将DispatcherServlet注入到Tomcat容器中。

4：TomcatStarter 往容器中注册servlet,filter需要通过RegistrationBean的实现类进行

RegistrationBean又是实现了ServletContextInitializer

接口，ServletContextInitializer实现类为ServletWebServerApplicationContext

.selfInitialize，在TomcatStarter执行onStartup

方法时调用

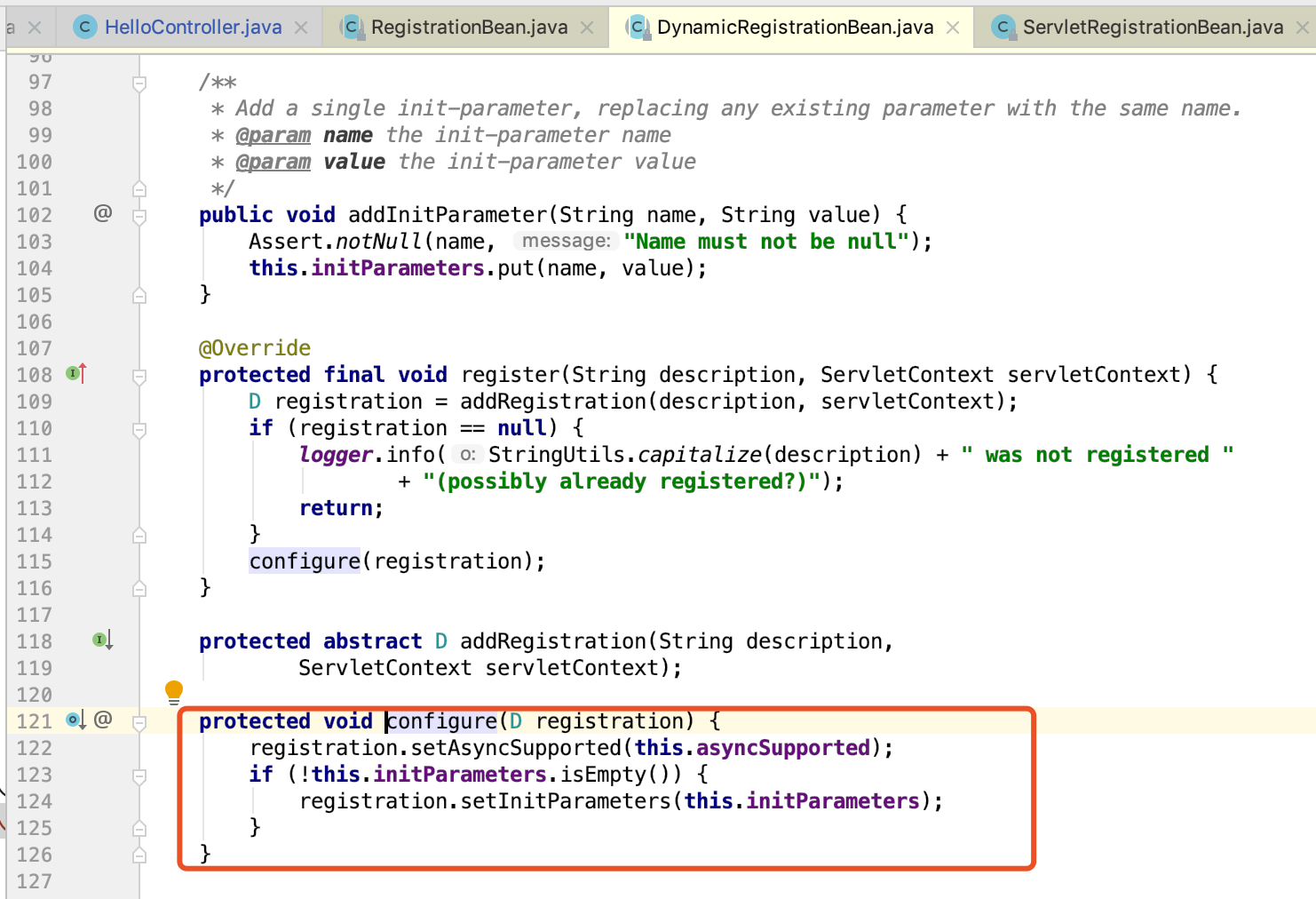
5：servlet,filter 注册步骤

260行：ServletWebServerApplicationContext.getServletContextInitializerBeans 从容器中获取所有实现了RegistrationBean接口的类，DispatcherServlet的为DispatcherServletRegistrationBean（由自动配置加载），还包括我们自己定义的实现类，然后调用每个RegistrationBean实现类的onStartup方法往容器中注册servlet,filter,这样我们的springmvc就和Tomcat容器关联起来了

6：DispatcherServlet默认配置在web容器启动时不进行初始化，第一次调用才进行初始化，

且默认开启异步

异步设置：



初始化设置：

