Spring boot web容器实例化

目录

[1. 介绍](#_Toc26063_WPSOffice_Level1) [2](#_Toc26063_WPSOffice_Level1)

[1.1 默认配置：tomcat](#_Toc24807_WPSOffice_Level2) [2](#_Toc24807_WPSOffice_Level2)

[1.2 其他配置：undertow和jetty](#_Toc17514_WPSOffice_Level2) [2](#_Toc17514_WPSOffice_Level2)

[1.3 Web服务器的配置：application.properties](#_Toc489_WPSOffice_Level2) [3](#_Toc489_WPSOffice_Level2)

[2. Web 服务器相关的结构信息](#_Toc24807_WPSOffice_Level1) [4](#_Toc24807_WPSOffice_Level1)

[2.1 WebServer 结构](#_Toc25991_WPSOffice_Level2) [4](#_Toc25991_WPSOffice_Level2)

[2.1.1 接口定义](#_Toc24807_WPSOffice_Level3) [4](#_Toc24807_WPSOffice_Level3)

[2.1.2 实现类](#_Toc17514_WPSOffice_Level3) [4](#_Toc17514_WPSOffice_Level3)

[2.2 WebServerFactory 结构](#_Toc10702_WPSOffice_Level2) [4](#_Toc10702_WPSOffice_Level2)

[2.2.1 介绍](#_Toc489_WPSOffice_Level3) [4](#_Toc489_WPSOffice_Level3)

[2.2.2 TomcatServletWebServerFactory](#_Toc25991_WPSOffice_Level3) [4](#_Toc25991_WPSOffice_Level3)

[3. 启动WebServer的流程](#_Toc17514_WPSOffice_Level1) [6](#_Toc17514_WPSOffice_Level1)

[3.1 创建WebServer](#_Toc1347_WPSOffice_Level2) [6](#_Toc1347_WPSOffice_Level2)

[3.1.1 调用链](#_Toc10702_WPSOffice_Level3) [6](#_Toc10702_WPSOffice_Level3)

[3.1.2 创建过程详解](#_Toc1347_WPSOffice_Level3) [6](#_Toc1347_WPSOffice_Level3)

[3.2 启动WebServer](#_Toc22771_WPSOffice_Level2) [6](#_Toc22771_WPSOffice_Level2)

[3.2.1 调用链](#_Toc22771_WPSOffice_Level3) [6](#_Toc22771_WPSOffice_Level3)

[3.2.2 启动过程详解](#_Toc20200_WPSOffice_Level3) [6](#_Toc20200_WPSOffice_Level3)

1. 介绍

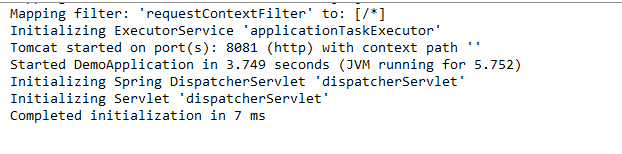
使用spring boot创建web项目不需要手动部署到诸如tomcat的服务容器上并启动，当项目运行起来的时候，spring boot会自动为创建、部署并启动web容器。Spring boot目前内嵌三种web容器，tomcat，undertow和jetty，其中tomcat是默认的web容器。

1.1 默认配置：tomcat

Pom.xml文件配置：引入web starter配置即可，starter-web的pom文件中引入了tomcat的starter

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
</**dependency**>

运行截图

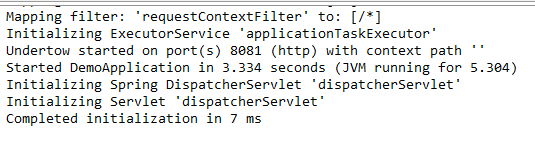


1.2 其他配置：undertow和jetty

Pom.xml文件设置：去掉默认的tomcat的starter配置并引入新容器的starter配置

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 <**exclusions**>  
 <**exclusion**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-tomcat</**artifactId**>  
 </**exclusion**>  
 </**exclusions**>  
</**dependency**>  
  
<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-undertow</**artifactId**>  
</**dependency**>

运行截图



1.3 Web服务器的配置：application.properties

server 配置通用属性

server.tomcat 配置tomcat的属性

server.port=# 配置程序端口，默认8080

server.serrsion-timeout= #用户会话session过期时间，以秒为单位

server.context-path= #配置访问路径。默认为/

server.tomcat.uri-encoding = #配置Tomcat编码，默认UTF-8

server.tomcat.compression = # Tomcat 是否开启压缩，默认为关闭off

1. Web 服务器相关的结构信息
   1. WebServer 结构

2.1.1 接口定义

**public interface** WebServer {**void** start() **throws** WebServerException;**void** stop() **throws** WebServerException;**int** getPort();  
}

2.1.2 实现类

以tomcat的实现为例。

**public class** TomcatWebServer **implements** WebServer {

用来存储服务和与之关联的连接器

**private final** Map<Service, Connector[]> **serviceConnectors**;  
tomcat实例  
**private final** Tomcat **tomcat**;

构造函数，传入一个tomcat实例和一个是否自动启动的标志。

**public** TomcatWebServer(Tomcat tomcat, **boolean** autoStart);

初始化方法，在构造函数内被调用

**private void** initialize() **throws** WebServerException;

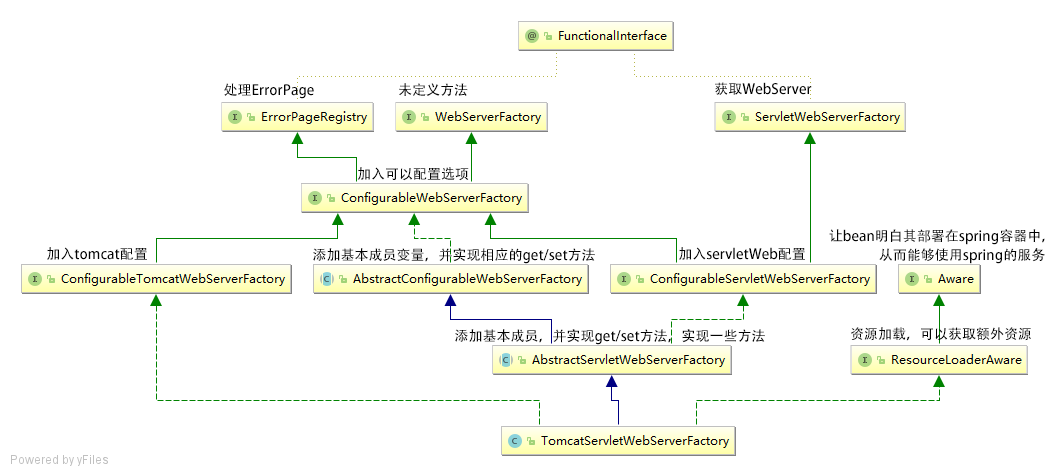
}

* 1. WebServerFactory 结构

2.2.1 介绍

依然以tomcat服务器为例。虽然TomcatWebServer的构造函数是公开的，但是在spring boot中并不推荐直接创建一个TomcatWebServer的实例，而是推荐使用相应的Factory来创建WebServer的实例。Springboot里实现了两个关于tomcat服务器的Factory，分别是TomcatReactiveWebServerFactory（对应于reactive web项目） 和TomcatServletWebServerFactory（对应于普通web项目）。

2.2.2 TomcatServletWebServerFactory



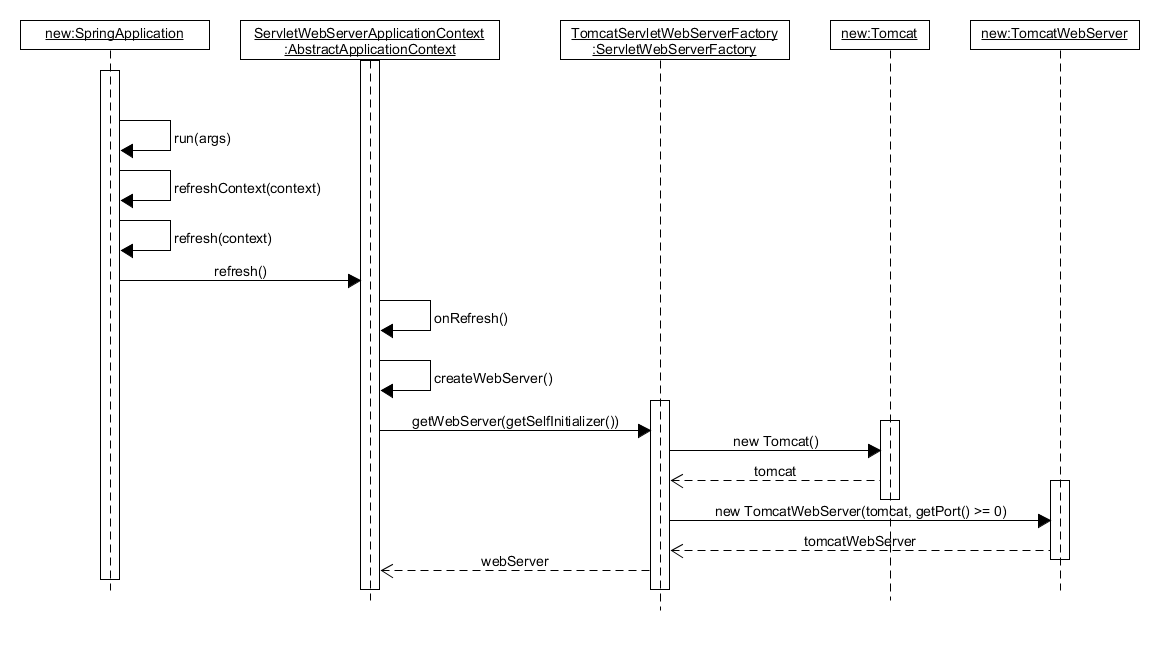
接口ServletWebServerFactory定义了获取WebServer的方法，getWebServer

**public interface** ServletWebServerFactory {WebServer getWebServer(ServletContextInitializer... initializers);

}

类TomcatServletWebServerFactory实现了getWebServer方法并返回一个TomcatWebServer对象。

1. 启动WebServer的流程
   1. 创建WebServer
      1. 调用链



* + 1. 创建过程详解
  1. 启动WebServer
     1. 调用链
     2. 启动过程详解