springMvc到springboot的演变过程

传统的springmvc的基本配置

1、下载springboot的源码 自行构建源码

2、首先分析传统的springmvc搭建步骤

project web.xml

```
<context-param>
 <param-name>contextConfigLocation</param-name>
 <param-value>classpath:applicationContext.xml</param-value>
</context-param>

prase applicationContext.xml为什么需要parse和这个xml，这个xml做了什么事？看下文会解释

<listener>
 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

ContextLoaderListener--------------初始化springcontext，为什么需要在这里初始化呢？
```

```
<servlet>
 <servlet-name>dispatcherServlet</servlet-name>
 <servlet-class>DispatcherServlet</servlet-class>
 <init-param>
 <param-name>contextConfigLocation</param-name>
 <param-value>classpath:spring-mvc.xml</param-value>
 </init-param>
</servlet>

servlet注册给给容器（tomcat、jetty注册一个servelt,会拦截所有的请求）?springboot如果注册的这个servlet
其实这并不是springboot完成----传统的spring当中已经提供了技术来完成？
parse spring-mvc.xml？为什么需要parse？这个xml当中做了什么事？
```

applicationContext.xml

```
<context:component-scan base-package="com.xxxx"/> 扫描业务类（dao）
```

springmvc.xml

```
<context:component-scan base-package="com.xxxxr" /> 扫描controller
```

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/view/"></property>
  <property name="suffix" value=".jsp"></property>
</bean>
视图解析，这个是不必须的，因为假设我们做个简单的demo，不需要返回页面，只需要访问到controller
```

- [ ] **至此传统的springmvc的搭建方式我们已经分析完了？接下来我们来分析springbott为什么可以替代传统的mvc方式来构建spring web应用？原因很简单，假设传统springmvc上述配置文件完成的事情能够以其他方式完成，那就完全可以替代了？那么springboot到底怎么完成的我先放一边，先以我们自己的思路来完成替代，然后再来看springboot的源码看看自己想的是不是和springboot的一样，就能清晰的对比出来了**

3、怎么利用spring5的新特性来完成对传统springmvc的改进----0配置

1. 去掉web.xml

利用servlet3.0提供的spi来完成servlet的注册

```
DispatcherServlet servlet = new DispatcherServlet(ac);
ServletRegistration.Dynamic registration = servletContext.addServlet("app", servlet);
```

利用spring javaconfig技术来完成对spring环境的初始化

```
AnnotationConfigWebApplicationContext ac = new AnnotationConfigWebApplicationContext();
```

2. 去掉applicationContext.xml

利用spring javaconfig技术来完成对spring bean的扫描

```
ac.register(AppLuban.class);
@ComponentScan("org.luban")
```

3. 去掉springmvc.xml

利用spring javaconfig技术来完成对spring controller的扫描

```
ac.register(AppLuban.class);
@ComponentScan("org.luban")
```

4、怎么来内嵌容器（tomcat）

利用依赖直接new Tomcat start

```
<!-- https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-core -->
<dependency>
 <groupId>org.apache.tomcat.embed</groupId>
 <artifactId>tomcat-embed-core</artifactId>
 <version>9.0.14</version>
</dependency>
```

```
Tomcat tomcat1 = new Tomcat();
 tomcat1.setPort(9876);
 try {
 tomcat1.start();
 tomcat1.getServer().await();
 } catch (LifecycleException e) {
 e.printStackTrace();
 }
```

onStartup 没有被调用到

```
除非tomcat源码当中对WebApplicationInitializer？那么tomcat会不会这么做呢？因为必须引入spring依赖
```