



Spring Boot入门



机票-服务平台 王君明

目录



- —. Spring Boot简介
- 二. 启动依赖(Starter)
- 三. 智能装配(AutoConfiguration)
- 四. 应用监控(Actuator)
- 五. Q&A





Spring Boot简介



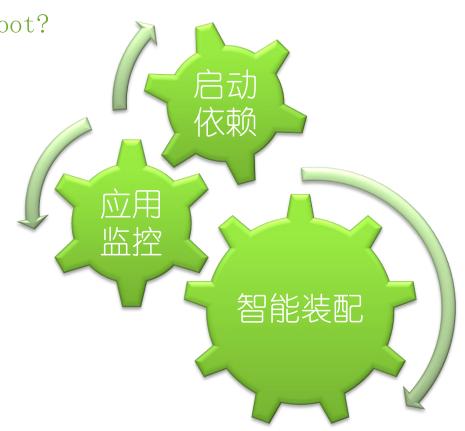
初识Spring Boot



⋄ Spring Boot是什么?

☆ 为什么学习Spring Boot?

◎ 如何下手





常规Spring MVC





配置pom通过starters



```
<packaging>jar</packaging>
           <parent>
               <groupId>org_springframework.boot</groupId>
11
               <artifactId>spring-boot-starter-parent</artifactId>
12
               <version>1.4.1.RELEASE
               <relativePath/>
15
           </parent>
           <dependencies>
17
               <dependency>
                   <groupId>org.springframework.boot
                   <artifactId>spring-boot-starter-web</artifactId>
               </dependency>
               <dependency>
21
                   <groupId>org.springframework.boot</groupId>
                   <artifactId>spring-boot-starter-test</artifactId>
                   <scope>test</scope>
               </dependency>
25
           </demendencies>
```

认识新的依赖 spring-boot-starter-xxx

9:注意 packaging 类型是 jar

12:官方推荐的parent设置

23:spring-boot-starter-web是个特殊的jar



Spring Boot Sample



```
@RestController
@SpringBootApplication
public class SampleApplication {

@RequestMapping("/home")
public String home() {
    return "Hello World!";
}

public static void main(String[] args) {

// SpringApplication.run(SampleApplication.class,args);
new SpringApplication(SampleApplication.class).run(args);
}

and the public static void main(String[] args) {

// SpringApplication(SampleApplication.class).run(args);
}
```

认识新的注解

10:@SpringBootApplication开启组件扫描和自动配置功能

13:定义mvc请求映射/home

20:main主程序启动的入口,启动内嵌tomcat默认8080端口



控制台输出



```
(v1.4.1.RELEASE)
                                                    main] com.boot.sample.SampleApplication
                                                                                                    : Starting SampleApplication on OF-junmingwang with PID 14024 (D:\WOM
2016-11-09 11:23:10.048
2016-11-09 11:23:10.053
                                                    main] com.boot.sample.SampleApplication
                                                                                                    : No active profile set, falling back to default profiles: default
2016-11-09 11:23:10.298
                                                    main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context mbedded.AnnotationConf
2016-11-09 11:23:13.543
                                                    main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat initialized with port(s): 8080 (b
2016-11-09 11:23:13.562
                                                    main] o.apache.catalina.core.StandardService
                                                                                                    : Starting service Tomcat
2016-11-09 11:23:13.564
                                                    main] org.apache.catalina.core.StandardEngine
                                                                                                   : Starting Servlet Engine:
                                                                                                                                                      集成
2016-11-09 11:23:13.707
                              14024 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/]
                                                                                                    : Initializing Spring embe
2016-11-09 11:23:13.707
                         INFO 14024 --- [ost-startStop-1] o.s.web.context.ContextLoader
                                                                                                    : Root WebApplicationConte
2016-11-09 11:23:13.902
                              14024 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean
                                                                                                   : Mapping servlet: 'dispat
2016-11-09 11:23:13.909
                             14024 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean
                                                                                                    : Mapping filter: 'charact
2016-11-09 11:23:13.910
                              14024 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean
                                                                                                    : Mapping filter: 'hidder'
2016-11-09 11:23:13.910
                              14024 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean
                                                                                                    : Mapping filter: 'http
2016-11-09 11:23:13.910
                              14024 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean
                                                                                                    : Mapping filter: 'reque
2016-11-09 11:23:14.373
                                                    main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerA
2016-11-09 11:23:14.472
                                                    main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/home]}" onto
2016-11-09 11:23:14.478
                                                    main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto
                                                    main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produc
2016-11-09 11:23:14.478
2016-11-09 11:23:14.515
                                                    main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjar
2016-11-09 11:23:14.515
                                                    main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] on
2016-11-09 11:23:14.568
                                                    main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favio
2016-11-09 11:23:14.773
                                                                                                   : Registering beans for JMX exposure on sta
2016-11-09 11:23:14.856
                                                    main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
                                                                                                    : Started SampleApplication in 6.087 seconds (JVM running for 7.003)
2016-11-09 11:23:14.864
```



Spring boot plugin和Runnable Jar



Spring boot插件运行

- 1:添加插件spring-boot-maven-plugin
- 2:直接运行 mvn spring-boot:run
- 3:打包Runnable Jar运行
- a)mvn clean package 生成Jar文件
- b)执行 java –jar xxxx.jar
- 4:curl http:/localhost:8080/home -s Hello World



常规war文件部署配置



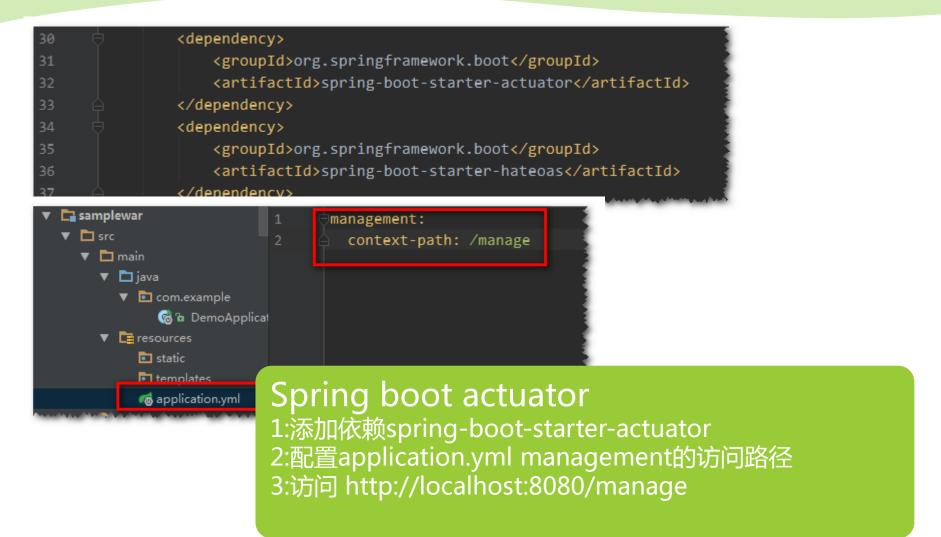
支持war方式运行

- 1:添加插件spring-boot-starter-tomcat 注意scope provided
- 2:可以添加 tomcat7-maven-plugin 使用 mvn tomcat7:run
- 3:可以添加 jetty-maven-plugin 使用 mvn jetty:run

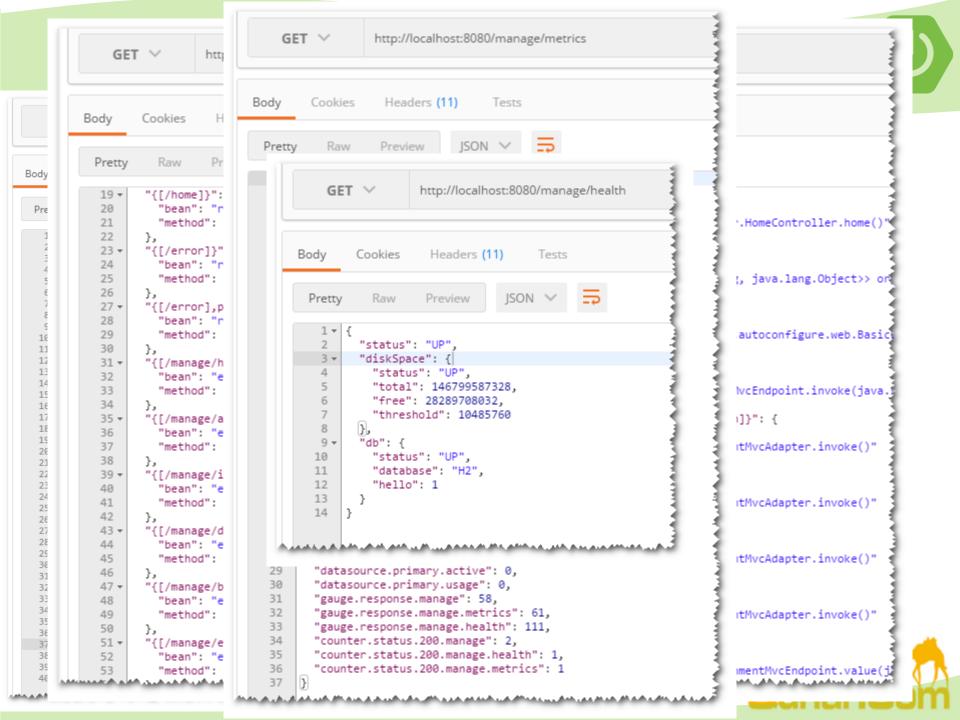


添加运维监控Starter Actuator









Spring Boot实例boot-sample



- ☆ 演示代码
 - git@gitlab.corp.qunar.com:junming.wang/spring4stepbystep.git
 - 分支boot
- ◎ 快速入门boot-sample
 - 体验Starter和AutoConfiguration
 - 了解可独立运行的Runnable Jar
- ☆ 兼容应用boot-samplewar
 - 兼容现在开发部署环境
 - 支持main方法快速启动测试
- 通用实例boot-showcase
 - Spring boot推荐的项目结构



Starters

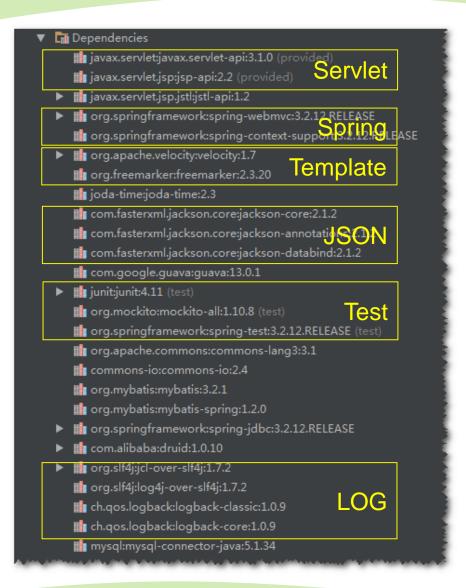


启动依赖



依赖管理的日常





添加依赖是个技术活

解决冲突是个体力活

拒绝囫囵拷贝

按需添加最小依赖



简洁无脑的Starter配置



```
<parent>
              <groupId>org.springframework.boot</groupId>
15
              <artifactId>spring-boot-starter-parent</artifactId>
              <version>1.4.1.RELEASE</version>
17
18
              <relativePath/>
19
          </parent>
                                                       简化
          <dependencies>
                                                                     功能
              <dependency>
21
          Spring WebowWoorg.springframework.boot</gr
22
                                                                     聚合
                                                       配置
                  <artifactId>spring-boot-starter-web
23
              </dependency>
25
              <dependency>
                  <groupId>org.springframework.boot
                                                       版本
                                                                     最小
         Test
27
                  <artifactId>spring-boot-starter-test
                  <scope>test</scope>
                                                                     依赖
29
              </dependency>
          </dependencies>
```



spring-boot-starter-web



```
spring-boot-starter-web:jar:1.4.1.RELEASE:compile
       spring-boot-starter:jar:1.4.1.RELEASE:compile
        +- spring-boot:jar:1.4.1.RELEASE:compile
 4
        +- spring-boot-autoconfigure:jar:1.4.1.RELEASE:compile
        +- spring-boot-starter-logging:jar:1.4.1.RELEASE:compile
 5
 6
        \- org.yaml:snakeyaml:jar:1.17:runtime
    +- spring-boot-starter-tomcat:jar:1.4.1.RELEASE:compile
    +- org.hibernate:hibernate-validator:jar:5.2.4.Final:compile
 8
    +- com.fasterxml.jackson.core:jackson-databind:jar:2.8.3:compile
       spring-web:jar:4.3.3.RELEASE:compile
10
        +- spring-aop:jar:4.3.3.RELEASE:compile
11
        +- spring-beans:jar:4.3.3.RELEASE:compile
12
        \- spring-context:jar:4.3.3.RELEASE:compile
13
     \- spring-webmvc:jar:4.3.3.RELEASE:compile
14
        \- spring-expression:jar:4.3.3.RELEASE:compile
15
```

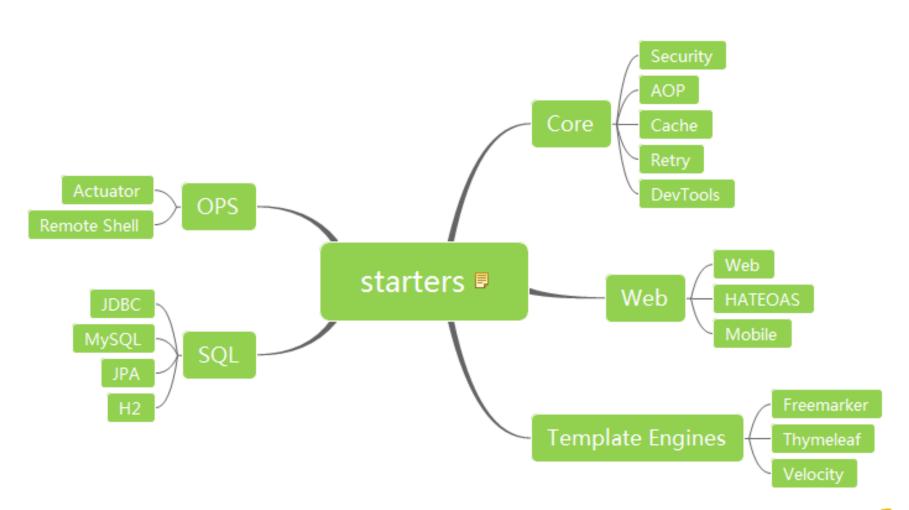
spring-boot-starter-web

利用maven的依赖管理 隐藏复杂的间接依赖 简化我们的配置



常用Starters







starter按需添加Maven仓库查询



GAV Search ▼ Group: org.springframework	c.boot Artifact: spring-boot-starter*	Version: 1.4.1.RELEASE Packaging
Group	Artifact 🛦	Version
org.springframework.boot	spring-boot-starter	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-activemq	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-actuator	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-amqp	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-aop	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-artemis	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-batch	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-cache	Latest: 1.4.1.RELEASE (Show All Versions)
org.springframework.boot	spring-boot-starter-cloud-connectors	Latest: 1.4.1.RELEASE (Show All Versions)
Displaying Top 54 records × Clear Result	ts	



AutoConfiguration







装配入口@SpringBootApplication



```
@RestController
            @SpringBootApplication
            public class SampleApplication {
12
                @RequestMapping("/home")
14
                public String home() {
                    return "Hello World!";
                                     @Target(ElementType.TYPE)
                                     @Retention(RetentionPolicy.RUNTIME)
                public stat:
                                     @Documented
                                     @Inherited
                                     @SpringBootConfiguration
                                                                        组合注解
                                     @EnableAutoConfiguration
                                     @ComponentScan(excludeFilters = @Filter(type = FilterType.CUSTO)
                                     public @interface SpringBootApplication {
```

启动注解一带三

10: @SpringBootApplication

开启组件扫描和自动配置功能

- @SpringBootConfiguration -> @Configuration
- @ComponentScan
- @EnableAutoConfiguration



智能装配@EnableAutoConfiguration



类似于众多EnableXXX注解

- EnableScheduling
- EnableAsync
- EnableCaching
- EnableAutoConfiguration

支持任务

开启异步

开启缓存

自动配置

加载智能配置xxxAutoCofniguration



```
# Auto Configure

org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
org.springframework.boot.autoconfigure.admin.SpringApplicationAdminJmxAutoConfiguration,\
org.springframework.boot.autoconfigure.aop.AopAutoConfiguration,\
org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,\
org.springframework.boot.autoconfigure.MessageSourceAutoConfiguration,\
org.springframework.boot.autoconfigure.PropertyPlaceholderAutoConfiguration,\
org.springframework.boot.autoconfigure.batch.BatchAutoConfiguration,\
org.springframework.boot.autoconfigure.cache.CacheAutoConfiguration,\
org.springframework.boot.autoconfigure.cassandra.CassandraAutoConfiguration,\
org.springframework.boot.autoconfigure.cloud.CloudAutoConfiguration,\
org.springframework.boot.autoconfigure.context.ConfigurationPropertiesAutoConfiguration,\
org.springframework.boot.autoconfigure.dao.PersistenceExceptionTranslationAutoConfiguration,\
org.springframework.boot.autoconfigure.data.cassandra.CassandraDataAutoConfiguration,\
org.springframework.boot.autoconfigure.data.cassandra.CassandraDataAutoConfiguration,\
org.springframework.boot.autoconfigure.data.cassandra.CassandraBataAutoConfiguration,\
org.springframework.boot.autoconfigure.data.cassandra.CassandraBataAuto
```

EnableAutoConfigurationImportSelector

加载/META-INF/spring.factories 众多的AutoConfiguration



列举装配AutoConfiguration



AutoConfiguration	装配作用
WebMvcAutoConfiguration	智能配置Spring Web MVC
DispatcherServletAutoConfiguration	智能配置控制器Servlet
HttpEncodingAutoConfiguration	智能字符编码过滤器
CacheAutoConfiguration	智能配置缓存支持
AopAutoConfiguration	智能配置AOP支持
JdbcTemplateAutoConfiguration	智能装配JdbcTemplate支持



JdbcTemplateAutoConfiguration



```
@Configuration
       @ConditionalOnClass({ DataSource.class, JdbcTemplate.class })
       @ConditionalOnSingleCandidate(DataSource.class)
       @AutoConfigureAfter(DataSourceAutoConfiguration.class)
       public class JdbcTemplateAutoConfiguration {
           private final DataSource dataSource;
           public JdbcTemplateAutoConfiguration(DataSource dataSource) { this.dataSource = dataSource; }
56 😘
           @Bean
           @Primary
           @ConditionalOnMissingBean(JdbcOperations.class)
           public JdbcTemplate jdbcTemplate() { return new JdbcTemplate(this.dataSource); }
63 🗞
           @Bean
           @Primary
           @ConditionalOnMissingBean(NamedParameterJdbcOperations.class)
           public NamedParameterJdbcTemplate namedParameterJdbcTemplate() {
               return new NamedParameterJdbcTemplate(this.dataSource);
```

智能装配的核心 @ConditionalXXX



智能装配Conditional



条件化注解	配置生效条件
@ConditionalOnBean	配置了某个特定Bean
@ConditionalOnMissingBean	没有配置特定的Bean
@ConditionalOnClass	Classpath里有指定的类
@ConditionalOnMissingClass	Classpath里缺少指定的类
@ConditionalOnExpression	给定的Spring Expression Language (SpEL) 表达式计算结果为true
@ConditionalOnJava	Java的版本匹配特定值或者一个范围值
@ConditionalOnProperty	指定的配置属性要有一个明确的值
@ConditionalOnResource	Classpath里有指定的资源
@ConditionalOnWebApplication	这是一个Web应用程序
@ConditionalOnNotWebApplication	这不是一个Web应用程序



Actuator







应用监控Actuator



Spring Boot Actuator

- 应用容器Bean和环境信息
- 展示自动装配情况
- 监控系统运行度量指标
- 暴露一些管理操作后门
- 支持REST, Shell, JMX访问



常用Actuator



应用信息

• / info 应用信息

• /beans 应用中所有的Bean

•/mappings @RequestMapping映射路径

• /autoconfig 自动装配信息

环境配置

• /env 环境信息

•/configprops 配置信息

度量指标

•/metrics 应用的各项动态指标

• /health 应用监控状态

• /heapdump dump堆内存信息

•/trace 追踪信息 最新的http请求



配置Actuator EndPoint



EndPoints Attributes

• id 标志 (字符数字_)

• sensitive 是否敏感信息 (true false)

• enable 是否可用 (true false)

```
security:
         user:
           name: admin
           password: admin
           role: ADMIN
11
      management:
12
         context-path: /manage
        security:
13
         roles: ADMIN
      endpoints:
15
        metrics:
17
           id: metrics
           enabled: true
           sensitive: false
```



深入BeansEndpoint



```
@ConfigurationProperties(prefix = "endpoints.beans")
        public class BeansEndpoint extends AbstractEndpoint<List<Object>>
                 implements ApplicationContextAware {
            private final LiveBeansView liveBeansView = new LiveBeansView();
            private final JsonParser parser = JsonParserFactory.getJsonParser();
            public BeansEndpoint() { super("beans"); }
            @Override
51 1 @
            public void setApplicationContext(ApplicationContext context) throws BeansException {
                 if (context.getEnvironment()
                         .getProperty(LiveBeansView.MBEAN DOMAIN PROPERTY NAME) == null) {
                     this.liveBeansView.setApplicationContext(context);
            @Override
59 🚮
            public List<Object> invoke() {
                 return this.parser.parseList(this.liveBeansView.getSnapshotAsJson());
```

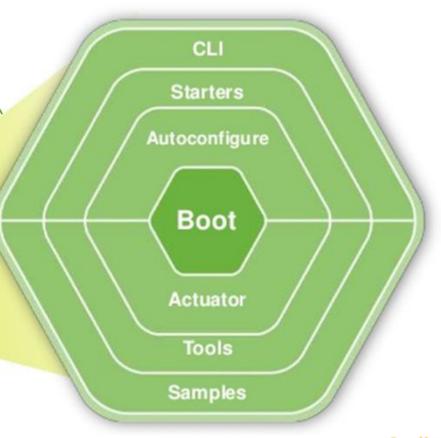


Spring Boot



- ☆ 提供丰富的Stater 简化POM配置
- ☆ 根据上下文智能装配
- ☆ 提供可视化的运行时信息
- ☆ 不要求xm1配置







Spring Boot



- ☆ 创建Stand-alone Spring应用
- ☆ 嵌入集成Tomcat、Jetty运行不需要WAR文件
- ☆ 有可能成为java微服务的标准

