

Spring-Boot 加载Bean的几种方式

01、课程内容

- 1、Spring-Boot 加载Bean的几种方式
- 2、自定义starter，顺便理解springboot加载原理和机制
- 3、自加载启动项
- 4、建模工具pdman和postman汉化问题

02、探索springboot的加载bean几种方式

springboot启动类上的注解是一个复合注解：

- @SpringBootConfiguration
 - 配置类 + @Bean
 - 配置类作用：方便扩展、方便集成，不受包约束。
- @EnableAutoConfiguration
 - springboot内部的starter加载的原理
 - 内部其实：配置+@Bean
- @ComponentScan （主要是用加载当前项目自身开发的bean）
 - 扫包 + @Service、@Controller、@Component

三者的作用都是一致的：把相关的bean加载ioc容器中。

03、@Enablexxx开关类

01、分析

未来在程序中我们会大量的看到@Enablexxx开关注解。这个注解本身没有加载bean到ioc容器作用，只是起到一个标记作用，真正把相关配置类加载到ioc容器的机制是：@Import。

@Import 才是开关类中，指定的类，而这个指定类要么是：

- 配置类（被@Configuration标记的类）
- 实现ImportSelector接口的类。

02、格式

```
1 @Enablexxx + @Import(配置类 || 实现ImportSelector接口的类。)
```

03、import机制的作用

如果你要去加载一个外部的配置类，你可以使用@Import机制。

springboot的starter的中一系列的配置类+@Bean是怎么加载进去：就是通过@Import机制

什么是外部的配置类？

当前默认包（启动类包）以外的配置类，都属于：外部配置类。

- 当前项目启动类包以外
- 自定义starter的配置类
- springboot自身提供的starter配置类

04、@Import + @Configuration配置类

具体实现如下：

加载的bean

```
1 package com.selector.service;
2
3 /**
4  * @author 飞哥
5  * @Title: 学相伴出品
6  * @Description: 飞哥B站地址:
7  * https://space.bilibili.com/490711252
8  * 记得关注和三连哦！
9  * @Description: 我们有一个学习网站: https://www.kuangstudy.com
10  * @date 2021/12/26 20:23
11  */
12 public class UserService {
13     public void sayHello(){
14         System.out.println("say hello !!!");
15     }
16 }
```

开关注解

```
1 package com.selector.anno;
2
3 import com.config.UserServiceConfiguration;
4 import com.selector.UserServiceImportSelector;
5 import org.springframework.context.annotation.Import;
6
7 import java.lang.annotation.*;
8
9 /**
10  * @author 飞哥
11  * @Title: 学相伴出品
12  * @Description: 我们有一个学习网站: https://www.kuangstudy.com
13  * @date 2021/12/26 20:32
14  */
15 @Target(ElementType.TYPE)
16 @Retention(RetentionPolicy.RUNTIME)
17 @Documented
18 @Inherited
19 @Import(UserServiceImportSelector.class)
20 public @interface EnableUserService2 {
21 }
22
```

配置类实现

```
1 package com.config;
2
3 import com.config.service.UserService;
4 import org.springframework.context.annotation.Bean;
5 import org.springframework.context.annotation.Configuration;
6
7 /**
8  * @author 飞哥
9  * @Title: 学相伴出品

```

```

10  * @Description: 飞哥B站地址:
    https://space.bilibili.com/490711252
11  * 记得关注和三连哦!
12  * @Description: 我们有一个学习网站: https://www.kuangstudy.com
13  * @date 2021/12/26 20:24
14  */
15  @Configuration
16  public class UserServiceConfiguration {
17
18      @Bean
19      public UserService userService(){
20          return new UserService();
21      }
22  }
23

```

 在需要的启动类上声明注解即可

```

1  package com.kuangstudy;
2
3  import com.config.UserServiceConfiguration;
4  import com.config.anno.EnableUserService;
5  import org.springframework.boot.SpringApplication;
6  import org.springframework.boot.SpringBootConfiguration;
7  import
    org.springframework.boot.autoconfigure.EnableAutoConfiguratio
    n;
8  import
    org.springframework.boot.autoconfigure.SpringBootApplication;
9  import org.springframework.context.annotation.ComponentScan;
10 import org.springframework.context.annotation.Import;
11 import
    org.springframework.scheduling.annotation.EnableScheduling;
12 import
    org.springframework.web.servlet.config.annotation.EnableWebMv
    c;

```

```
13
14 @SpringBootApplication
15 @EnableUserService
16 public class SpringbootEnableImportApplication {
17
18     public static void main(String[] args) {
19
20         SpringApplication.run(SpringbootEnableImportApplication.class
21             s, args);
22     }
23 }
```

测试

```
1 package com.kuangstudy;
2
3 import com.config.service.UserService;
4 import com.selector.service.UserService2;
5 import org.junit.jupiter.api.Test;
6 import
7     org.springframework.beans.factory.annotation.Autowired;
8
9 import org.springframework.boot.test.context.SpringBootTest;
10
11 @SpringBootTest
12 class SpringbootEnableImportApplicationTests {
13
14     @Autowired
15     private UserService userService;
16
17     @Test
18     void contextLoads() {
19         userService.sayHello();
20     }
21 }
```

```
20
21 }
22
```

运行结果：say hello !!! 说明成功

05、@Import + importselector接口

无论你是用配置类还是importselector接口的方式，它们的目的都是一致的：

- 要么用配置类+@Bean，加载到ioc容器中
- 要么用ImportSelector接口的数组的方式指定加载到ioc容器中

具体实现如下：

加载的bean

```
1 package com.selector.service;
2
3 /**
4  * @author 飞哥
5  * @Title: 学相伴出品
6  * @Description: 飞哥B站地址:
7  * https://space.bilibili.com/490711252
8  * 记得关注和三连哦！
9  * @Description: 我们有一个学习网站: https://www.kuangstudy.com
10  * @date 2021/12/26 20:23
11  */
12 public class UserService2 {
13     public void sayHello(){
```

```
13         System.out.println("say hello !!!");
14     }
15 }
16
```

开关注解

```
1 package com.selector.anno;
2
3 import com.config.UserServiceConfiguration;
4 import com.selector.UserServiceImportSelector;
5 import org.springframework.context.annotation.Import;
6
7 import java.lang.annotation.*;
8
9 /**
10  * @author 飞哥
11  * @Title: 学相伴出品
12  * @Description: 我们有一个学习网站: https://www.kuangstudy.com
13  * @date 2021/12/26 20:32
14  */
15 @Target(ElementType.TYPE)
16 @Retention(RetentionPolicy.RUNTIME)
17 @Documented
18 @Inherited
19 @Import(UserServiceImportSelector.class)
20 public @interface EnableUserService2 {
21 }
22
```

importselector实现

```
1 package com.selector;
2
3 import org.springframework.context.annotation.ImportSelector;
```



```

4  import org.springframework.core.type.AnnotationMetadata;
5
6  /**
7   * @author 飞哥
8   * @Title: 学相伴出品
9   * @Description: 飞哥B站地址:
10   https://space.bilibili.com/490711252
11   * 记得关注和三连哦!
12   * @Description: 我们有一个学习网站: https://www.kuangstudy.com
13   * @date 2021/12/26 20:39
14   */
15
16  public class UserServiceImportSelector implements
17      ImportSelector {
18
19      @Override
20      public String[] selectImports(AnnotationMetadata
21      importingClassMetadata) {
22          return new String[]
23          {"com.selector.service.UserService2"};
24      }
25  }
26
27

```

 在需要的启动类上声明注解即可

```

1  package com.kuangstudy;
2
3  import com.config.UserServiceConfiguration;
4  import com.config.anno.EnableUserService;
5  import com.selector.anno.EnableUserService2;
6  import org.springframework.boot.SpringApplication;
7  import org.springframework.boot.SpringBootConfiguration;
8  import
9      org.springframework.boot.autoconfigure.EnableAutoConfiguratio
10      n;
11

```

```

9  import
   org.springframework.boot.autoconfigure.SpringBootApplication;
10 import org.springframework.context.annotation.ComponentScan;
11 import org.springframework.context.annotation.Import;
12 import
   org.springframework.scheduling.annotation.EnableScheduling;
13 import
   org.springframework.web.servlet.config.annotation.EnableWebMvc;
14
15 @SpringBootApplication
16 @EnableUserService2
17 public class SpringbootEnableImportApplication {
18
19     public static void main(String[] args) {
20
21         SpringApplication.run(SpringbootEnableImportApplication.class, args);
22     }
23 }
24

```

测试

```

1  package com.kuangstudy;
2
3  import com.config.service.UserService;
4  import com.selector.service.UserService2;
5  import org.junit.jupiter.api.Test;
6  import
   org.springframework.beans.factory.annotation.Autowired;
7  import org.springframework.boot.test.context.SpringBootTest;
8
9  @SpringBootTest
10 class SpringbootEnableImportApplicationTests {

```

```
11
12     @Autowired
13     private UserService2 userService2;
14
15
16     @Test
17     void contextLoads() {
18         userService2.sayHello();
19     }
20
21 }
22
```

运行结果：say hello !!! 说明成功

05、importselector接口存在的意义是什么？

就专门去加载springboot提供的starter和自定义的starter的一种机制。

- spring-boot-starter-web
 - 配置类 + @Bean
- mybatis-boot-starter
 - 配置类 + @Bean
- kaptcha-boot-starter
 - 配置类 + @Bean

官方starter

- 因为springboot的autoconfiguration的目录META-INF提供专门的配置文件spring.factories。这个文件中就定义springboot中starter所需要加载的所有的配置类。

spring-boot-autoconfigure-2.6.1.jar 包下：

```
1
2 # Auto Configure
3 org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
4 org.springframework.boot.autoconfigure.admin.SpringApplicationAdminJmxAutoConfiguration,\
5 org.springframework.boot.autoconfigure.aop.AopAutoConfiguration,\
6 org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,\
7 org.springframework.boot.autoconfigure.batch.BatchAutoConfiguration,\
8 org.springframework.boot.autoconfigure.cache.CacheAutoConfiguration,\
9 org.springframework.boot.autoconfigure.cassandra.CassandraAutoConfiguration,\
10 org.springframework.boot.autoconfigure.context.ConfigurationPropertiesAutoConfiguration,\
11 org.springframework.boot.autoconfigure.context.LifecycleAutoConfiguration,\
12 org.springframework.boot.autoconfigure.context.MessageSourceAutoConfiguration,\
13 org.springframework.boot.autoconfigure.context.PropertyPlaceholderAutoConfiguration,\
14 org.springframework.boot.autoconfigure.couchbase.CouchbaseAutoConfiguration,\
15 org.springframework.boot.autoconfigure.dao.PersistenceExceptionTranslationAutoConfiguration,\
16 org.springframework.boot.autoconfigure.data.cassandra.CassandraDataAutoConfiguration,\
17 org.springframework.boot.autoconfigure.data.cassandra.CassandraReactiveDataAutoConfiguration,\
18 org.springframework.boot.autoconfigure.data.cassandra.CassandraReactiveRepositoriesAutoConfiguration,\
19 org.springframework.boot.autoconfigure.data.cassandra.CassandraRepositoriesAutoConfiguration,\
20 org.springframework.boot.autoconfigure.data.couchbase.CouchbaseDataAutoConfiguration,\
```

```
21 org.springframework.boot.autoconfigure.data.couchbase.C
   ouchbaseReactiveDataAutoConfiguration,\
22 org.springframework.boot.autoconfigure.data.couchbase.C
   ouchbaseReactiveRepositoriesAutoConfiguration,\
23 org.springframework.boot.autoconfigure.data.couchbase.C
   ouchbaseRepositoriesAutoConfiguration,\
24 org.springframework.boot.autoconfigure.data.elasticsear
   ch.ElasticsearchDataAutoConfiguration,\
25 org.springframework.boot.autoconfigure.data.elasticsear
   ch.ElasticsearchRepositoriesAutoConfiguration,\
26 org.springframework.boot.autoconfigure.data.elasticsear
   ch.ReactiveElasticsearchRepositoriesAutoConfiguration,\
27 org.springframework.boot.autoconfigure.data.elasticsear
   ch.ReactiveElasticsearchRestClientAutoConfiguration,\
28 org.springframework.boot.autoconfigure.data.jdbc.JdbcRe
   positoriesAutoConfiguration,\
29 org.springframework.boot.autoconfigure.data.jpa.JpaRepo
   sitoriesAutoConfiguration,\
30 org.springframework.boot.autoconfigure.data ldap.LdapRe
   positoriesAutoConfiguration,\
31 org.springframework.boot.autoconfigure.data.mongo.Mongo
   DataAutoConfiguration,\
32 org.springframework.boot.autoconfigure.data.mongo.Mongo
   ReactiveDataAutoConfiguration,\
33 org.springframework.boot.autoconfigure.data.mongo.Mongo
   ReactiveRepositoriesAutoConfiguration,\
34 org.springframework.boot.autoconfigure.data.mongo.Mongo
   RepositoriesAutoConfiguration,\
35 org.springframework.boot.autoconfigure.data.neo4j.Neo4j
   DataAutoConfiguration,\
36 org.springframework.boot.autoconfigure.data.neo4j.Neo4j
   ReactiveDataAutoConfiguration,\
37 org.springframework.boot.autoconfigure.data.neo4j.Neo4j
   ReactiveRepositoriesAutoConfiguration,\
38 org.springframework.boot.autoconfigure.data.neo4j.Neo4j
   RepositoriesAutoConfiguration,\
39 org.springframework.boot.autoconfigure.data.r2dbc.R2dbc
   DataAutoConfiguration,\
```

```
40 org.springframework.boot.autoconfigure.data.r2dbc.R2dbc
RepositoriesAutoConfiguration,\
41 org.springframework.boot.autoconfigure.data.redis.Redis
AutoConfiguration,\
42 org.springframework.boot.autoconfigure.data.redis.Redis
ReactiveAutoConfiguration,\
43 org.springframework.boot.autoconfigure.data.redis.Redis
RepositoriesAutoConfiguration,\
44 org.springframework.boot.autoconfigure.data.rest.Reposi
toryRestMvcAutoConfiguration,\
45 org.springframework.boot.autoconfigure.data.web.SpringD
ataWebAutoConfiguration,\
46 org.springframework.boot.autoconfigure.elasticsearch.El
asticsearchRestClientAutoConfiguration,\
47 org.springframework.boot.autoconfigure.flyway.FlywayAut
oConfiguration,\
48 org.springframework.boot.autoconfigure.freemarker.FreeM
arkerAutoConfiguration,\
49 org.springframework.boot.autoconfigure.groovy.template.
GroovyTemplateAutoConfiguration,\
50 org.springframework.boot.autoconfigure.gson.GsonAutoCon
figuration,\
51 org.springframework.boot.autoconfigure.h2.H2ConsoleAuto
Configuration,\
52 org.springframework.boot.autoconfigure.hateoas.Hypermed
iaAutoConfiguration,\
53 org.springframework.boot.autoconfigure.hazelcast.Hazelc
astAutoConfiguration,\
54 org.springframework.boot.autoconfigure.hazelcast.Hazelc
astJpaDependencyAutoConfiguration,\
55 org.springframework.boot.autoconfigure.http.HttpMessage
ConvertersAutoConfiguration,\
56 org.springframework.boot.autoconfigure.http.codec.Codec
sAutoConfiguration,\
57 org.springframework.boot.autoconfigure.influx.InfluxDbA
utoConfiguration,\
58 org.springframework.boot.autoconfigure.info.ProjectInfo
AutoConfiguration,\
```

```
59 org.springframework.boot.autoconfigure.integration.Inte
   grationAutoConfiguration,\
60 org.springframework.boot.autoconfigure.jackson.JacksonA
   utoConfiguration,\
61 org.springframework.boot.autoconfigure.jdbc.DataSourceA
   utoConfiguration,\
62 org.springframework.boot.autoconfigure.jdbc.JdbcTemplate
   AutoConfiguration,\
63 org.springframework.boot.autoconfigure.jdbc.IndiDataSou
   rceAutoConfiguration,\
64 org.springframework.boot.autoconfigure.jdbc.XADataSourc
   eAutoConfiguration,\
65 org.springframework.boot.autoconfigure.jdbc.DataSourceT
   ransactionManagerAutoConfiguration,\
66 org.springframework.boot.autoconfigure.jms.JmsAutoConfi
   guration,\
67 org.springframework.boot.autoconfigure.jmx.JmxAutoConfi
   guration,\
68 org.springframework.boot.autoconfigure.jms.IndiConnecti
   onFactoryAutoConfiguration,\
69 org.springframework.boot.autoconfigure.jms.activemq.Act
   iveMQAutoConfiguration,\
70 org.springframework.boot.autoconfigure.jms.artemis.Arte
   misAutoConfiguration,\
71 org.springframework.boot.autoconfigure.jersey.JerseyAut
   oConfiguration,\
72 org.springframework.boot.autoconfigure.jooq.JooqAutoCon
   figuration,\
73 org.springframework.boot.autoconfigure.jsonb.JsonbAutoC
   onfiguration,\
74 org.springframework.boot.autoconfigure.kafka.KafkaAutoC
   onfiguration,\
75 org.springframework.boot.autoconfigure.availability.App
   licationAvailabilityAutoConfiguration,\
76 org.springframework.boot.autoconfigure ldap.embedded.Em
   beddedLdapAutoConfiguration,\
77 org.springframework.boot.autoconfigure ldap.LdapAutoCon
   figuration,\
```

```
78 org.springframework.boot.autoconfigure.liquibase.Liquib
aseAutoConfiguration,\
79 org.springframework.boot.autoconfigure.mail.MailSenderA
utoConfiguration,\
80 org.springframework.boot.autoconfigure.mail.MailSenderV
alidatorAutoConfiguration,\
81 org.springframework.boot.autoconfigure.mongo.embedded.E
mbeddedMongoAutoConfiguration,\
82 org.springframework.boot.autoconfigure.mongo.MongoAutoC
onfiguration,\
83 org.springframework.boot.autoconfigure.mongo.MongoReact
iveAutoConfiguration,\
84 org.springframework.boot.autoconfigure.mustache.Mustach
eAutoConfiguration,\
85 org.springframework.boot.autoconfigure.neo4j.Neo4jAutoC
onfiguration,\
86 org.springframework.boot.autoconfigure.netty.NettyAutoC
onfiguration,\
87 org.springframework.boot.autoconfigure.orm.jpa.Hibernat
eJpaAutoConfiguration,\
88 org.springframework.boot.autoconfigure.quartz.QuartzAut
oConfiguration,\
89 org.springframework.boot.autoconfigure.r2dbc.R2dbcAutoC
onfiguration,\
90 org.springframework.boot.autoconfigure.r2dbc.R2dbcTrans
actionManagerAutoConfiguration,\
91 org.springframework.boot.autoconfigure.rsocket.RSocketM
essagingAutoConfiguration,\
92 org.springframework.boot.autoconfigure.rsocket.RSocketR
equesterAutoConfiguration,\
93 org.springframework.boot.autoconfigure.rsocket.RSocketS
erverAutoConfiguration,\
94 org.springframework.boot.autoconfigure.rsocket.RSocketS
trategiesAutoConfiguration,\
95 org.springframework.boot.autoconfigure.security.servlet
.SecurityAutoConfiguration,\
96 org.springframework.boot.autoconfigure.security.servlet
.UserDetailsServiceAutoConfiguration,\
```



```
97  org.springframework.boot.autoconfigure.security.servlet
    .SecurityFilterAutoConfiguration,\
98  org.springframework.boot.autoconfigure.security.reactiv
    e.ReactiveSecurityAutoConfiguration,\
99  org.springframework.boot.autoconfigure.security.reactiv
    e.ReactiveUserDetailsServiceAutoConfiguration,\
100 org.springframework.boot.autoconfigure.security.rsocket
    .RSocketSecurityAutoConfiguration,\
101 org.springframework.boot.autoconfigure.security.saml2.S
    aml2RelyingPartyAutoConfiguration,\
102 org.springframework.boot.autoconfigure.sendgrid.SendGri
    dAutoConfiguration,\
103 org.springframework.boot.autoconfigure.session.SessionA
    utoConfiguration,\
104 org.springframework.boot.autoconfigure.security.oauth2.
    client.servlet.OAuth2ClientAutoConfiguration,\
105 org.springframework.boot.autoconfigure.security.oauth2.
    client.reactive.ReactiveOAuth2ClientAutoConfiguration,\
106 org.springframework.boot.autoconfigure.security.oauth2.
    resource.servlet.OAuth2ResourceServerAutoConfiguration,\
    \
107 org.springframework.boot.autoconfigure.security.oauth2.
    resource.reactive.ReactiveOAuth2ResourceServerAutoConfi
    guration,\
108 org.springframework.boot.autoconfigure.solr.SolrAutoCon
    figuration,\
109 org.springframework.boot.autoconfigure.sql.init.SqlInit
    ializationAutoConfiguration,\
110 org.springframework.boot.autoconfigure.task.TaskExecuti
    onAutoConfiguration,\
111 org.springframework.boot.autoconfigure.task.TaskSchedul
    ingAutoConfiguration,\
112 org.springframework.boot.autoconfigure.thymeleaf.Thymel
    eafAutoConfiguration,\
113 org.springframework.boot.autoconfigure.transaction.Tran
    sactionAutoConfiguration,\
114 org.springframework.boot.autoconfigure.transaction.jta.
    JtaAutoConfiguration,\
```

```
115 org.springframework.boot.autoconfigure.validation.Valid
    ationAutoConfiguration,\
116 org.springframework.boot.autoconfigure.web.client.RestT
    emplateAutoConfiguration,\
117 org.springframework.boot.autoconfigure.web.embedded.Emb
    eddedWebServerFactoryCustomizerAutoConfiguration,\
118 org.springframework.boot.autoconfigure.web.reactive.Http
    pHandlerAutoConfiguration,\
119 org.springframework.boot.autoconfigure.web.reactive.Rea
    ctiveMultipartAutoConfiguration,\
120 org.springframework.boot.autoconfigure.web.reactive.Rea
    ctiveWebServerFactoryAutoConfiguration,\
121 org.springframework.boot.autoconfigure.web.reactive.Web
    FluxAutoConfiguration,\
122 org.springframework.boot.autoconfigure.web.reactive.Web
    SessionIdResolverAutoConfiguration,\
123 org.springframework.boot.autoconfigure.web.reactive.err
    or.ErrorWebFluxAutoConfiguration,\
124 org.springframework.boot.autoconfigure.web.reactive.fun
    ction.client.ClientHttpConnectorAutoConfiguration,\
125 org.springframework.boot.autoconfigure.web.reactive.fun
    ction.client.WebClientAutoConfiguration,\
126 org.springframework.boot.autoconfigure.web.servlet.Disp
    atcherServletAutoConfiguration,\
127 org.springframework.boot.autoconfigure.web.servlet.Serv
    letWebServerFactoryAutoConfiguration,\
128 org.springframework.boot.autoconfigure.web.servlet.erro
    r.ErrorMvcAutoConfiguration,\
129 org.springframework.boot.autoconfigure.web.servlet.Http
    EncodingAutoConfiguration,\
130 org.springframework.boot.autoconfigure.web.servlet.Mult
    ipartAutoConfiguration,\
131 org.springframework.boot.autoconfigure.web.servlet.webM
    vcAutoConfiguration,\
132 org.springframework.boot.autoconfigure.websocket.reacti
    ve.WebSocketReactiveAutoConfiguration,\
133 org.springframework.boot.autoconfigure.websocket.servle
    t.WebSocketServletAutoConfiguration,\
```

```
134 org.springframework.boot.autoconfigure.websocket.servle  
t.WebSocketMessagingAutoConfiguration,\n135 org.springframework.boot.autoconfigure.webservices.WebS  
ervicesAutoConfiguration,\n136 org.springframework.boot.autoconfigure.webservices.clie  
nt.WebServiceTemplateAutoConfiguration
```

自定义starter

- 如果你要自定义starter，你就必须在你maven工程中的resources目录下新建一个META-INF/spring.factories。把你项目中自己的配置类在这个文件中进行指定即可。

mybatis-plus-boot-starter-3.4.2.jar 如下：

```
1 # Auto Configure\n2 org.springframework.boot.env.EnvironmentPostProcessor=\n3\n4 com.baomidou.mybatisplus.autoconfigure.SafetyEncryptProcessor\n5 org.springframework.boot.autoconfigure.EnableAutoConfiguration\n6 =\n7\n8 com.baomidou.mybatisplus.autoconfigure.MybatisPlusLanguageDriv  
erAutoConfiguration,\n9\n10 com.baomidou.mybatisplus.autoconfigure.MybatisPlusAutoConfigur  
ation
```

如何加载这些上面的这些**starter**的配置类的呢？

原理是：importselector机制，处理者：AutoConfigurationImportSelector

- 第一步：这个接口的实现类 AutoConfigurationImportSelector 中，它会读取项目中所有的META-INF/spring.factories这个文件
- 第二步：把这个文件中所有关于这个注解EnableAutoConfiguration的配置类全部找出来。
- 第三步：把找出来的配置类根据条件注解@Condition进行过滤，过滤放入到集合set
- 第四步：把set集合中的已经明确加载的配置类，放入到数组中。
- 第五步：结束

参考

- Spring boot Bean加载源码分析
<http://blog.csdn.net/liaokailin/article/details/49107209>
- Spring boot 自动配置原理分析
<http://blog.csdn.net/liaokailin/article/details/49559951>
- Spring Boot 使用Java代码创建Bean并注册到Spring中
<http://blog.csdn.net/catoop/article/details/50558333>