Spring自动配置原理分析

- @SpringBootApplication
  - @SpringBootConfiguration：SpringBoot配置注解
  - @EnableAutoConfiguration：开启自动配置功能
    - 这个注解可以帮助我们自动载入应用程序所需要的所有默认配置。
    - @AutoConfigurationPackage：自动配置包
      - @Import(AutoConfigurationPackages.Registrar.class)
        - 在默认的情况下就是将：主配置类(@SpringBootApplication)的所在包及其子包里边的组件扫描到Spring容器中。
        - 和@ComponentScan不同之处：比如说，你用了Spring Data JPA，可能会在实体类上写@Entity注解。这个@Entity注解由@AutoConfigurationPackage扫描并加载，而我们平时开发用的@Controller/@Service/@Component/@Repository这些注解是由ComponentScan来扫描并加载的。
    - @Import(AutoConfigurationImportSelector.class)：导入自动配置选择器，自动配置的核心

      ```
      protected AutoConfigurationEntry getAutoConfigurationEntry(AnnotationMetadata annotationMetadata) {
      if (!isEnabled(annotationMetadata)) {
      return EMPTY_ENTRY;
      }
      AnnotationAttributes attributes = getAttributes(annotationMetadata);
      List<String> configurations = getCandidateConfigurations(annotationMetadata, attributes);

      ......
      ```

      ```
      protected List<String> getCandidateConfigurations(AnnotationMetadata metadata, AnnotationAttributes attributes) {
      List<String> configurations = SpringFactoriesLoader.loadFactoryNames(getSpringFactoriesLoaderFactoryClass(),

      ........
      ```

      ```
      //使用给定的类加载器从"META-INF/spring.factories"加载给定类型的工厂实现的完全限定类名。
      public static List<String> loadFactoryNames(Class<?> factoryType, @Nullable ClassLoader classLoader) {
      ClassLoader classLoaderToUse = classLoader;
      if (classLoaderToUse == null) {
      classLoaderToUse = SpringFactoriesLoader.class.getClassLoader();
      }
      String factoryTypeName = factoryType.getName();
      return loadSpringFactories(classLoaderToUse).getOrDefault(factoryTypeName, Collections.emptyList());
      }
      ```

      ```
      public static final String FACTORIES_RESOURCE_LOCATION = "META-INF/spring.factories";
      ```

      ```
      private static Map<String, List<String>> loadSpringFactories(ClassLoader classLoader) {

        Enumeration<URL> urls = classLoader.getResources(FACTORIES_RESOURCE_LOCATION);
      ```

      FACTORIES_RESOURCE_LOCATION的值是META-INF/spring.factories，Spring启动的时候会扫描所有jar路径下的META-INF/spring.factories，将其文件包装成Properties对象从Properties对象获取到key值为EnableAutoConfiguration的数据，然后添加到容器里边。
  - @ComponentScan
    - 扫描注解，默认扫描当前类下的Package，将@Controller/@Service/@Component/@Repository等注解加载到IOC容器中。
- 总结
  - @SpringBootApplication等同于下面三个注解：
    - @SpringBootConfiguration
    - @EnableAutoConfiguration
    - @ComponentScan

    其中@EnableAutoConfiguration是关键(启用自动配置)，内部实际上就去加载META-INF/spring.factories文件的信息，然后筛选出以EnableAutoConfiguration为key的数据，加载到IOC容器中，实现自动配置功能！