



链滴

# 如何实现一个 Spring Boot Starter

作者: [xjlnjut730](#)

原文链接: <https://ld246.com/article/1576109577310>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 1. Starter 是什么

在实现 Spring Boot Starter 之前，首先要明确实现 Spring Boot Starter 是干嘛用的？看一下方的描述：

<blockquote>

Starters are a set of convenient dependency descriptors that you can include in your application. You get a one-stop shop for all the Spring and related technologies that you need without having to hunt through sample code and copy-paste loads of dependency descriptors. For example, if you want to get started using Spring and JPA for database access, include the spring-boot-starter-data-jpa dependency in your project.

</blockquote>

Starters 是一些便捷实用的依赖，可以在工程中引用。你可以通过 Starter，直接得到你所需要的 Spring 以及相关技术的一站式服务，而不再需要通过复制样例代码和依赖的方式。举个例子，如果需要使用 Spring 和 JPA 来进行数据库访问，只需要在工程中引入 spring-boot-starter-data-jpa 即可。

通过官方的描述，我们可以发现 Starter 是通过自动注入和依赖管理的方式来帮我们简化应用开发。

## 2. Starter 构成

一个完整的 Spring Boot Starter 包含以下组件：

<ol>

<li>一个 autoconfigure 模块，包含了自动注入的代码。

<li>一个 starter 模块，提供 autoconfigure 模块所需要的依赖。简而言之，这个 starter 需要包含该 starter 所需要的所有依赖。

</ol>

以 github 上一个 spring-boot-custom-starter 库为例(见参考资料 4)，我们看下一个 Starter 程序的样例：

```
<code>$ tree -L 2
```

```
</code>
```

```
<code>├── README.md
```

```
<code>├── greeter
```

```
<code>├── pom.xml
```

```
<code>├── greeter-library
```

```
<code>├── pom.xml
```

```
<code>├── pom.xml
```

```
<code>├── src
```

```
<code>├── greeter-spring-boot-autoconfigure
```

```
<code>├── pom.xml
```

```
<code>├── src
```

```
<code>├── greeter-spring-boot-sample-app
```

```
<code>├── pom.xml
```

```
<code>├── src
```

```
<code>├── greeter-spring-boot-starter
```





```

</span></span><span class="highlight-line"><span class="highlight-cl"> @Bean
</span></span><span class="highlight-line"><span class="highlight-cl"> @ConditionalO
MissingBean
</span></span><span class="highlight-line"><span class="highlight-cl"> public Greeting
onfig greeterConfig() {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> String userN
me = greeterProperties.getUserName() == null ? System.getProperty("user.name") : greeterP
operties.getUserName();
</span></span><span class="highlight-line"><span class="highlight-cl"> // ...略去类似
逻辑
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> GreetingConf
g greetingConfig = new GreetingConfig();
</span></span><span class="highlight-line"><span class="highlight-cl"> greetingConf
g.put(USER_NAME, userName);
</span></span><span class="highlight-line"><span class="highlight-cl"> // ...略去类似
逻辑
</span></span><span class="highlight-line"><span class="highlight-cl"> return greeti
gConfig;
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Bean
</span></span><span class="highlight-line"><span class="highlight-cl"> @ConditionalO
MissingBean
</span></span><span class="highlight-line"><span class="highlight-cl"> public Greeter
reeter(GreetingConfig greetingConfig) {
</span></span><span class="highlight-line"><span class="highlight-cl"> return new G
eeter(greetingConfig);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>

```

<p>对应的 GreeterProperties 这里就不贴了。 </p>

<p>@ConditionalOnClass(Greeter.class)表示：当应用启动时，如果 Spring Boot 发现 Greeter 在于 classpath 中，那么 GreeterAutoConfiguration 就会被执行。 </p>

<p>当 GreeterAutoConfiguration 执行时，Spring 容器如果发现工程中没有定义 GreetingConfig 这个 Bean，则会默认通过 greeterConfig()方法创建一个，并放到 Spring 容器中。类似的，如果 pri g 容器如果发现工程中没有定义 Greeter 这个 Bean，则会默认根据 greeter(...)方法创建一个，并放到 Spring 容器中。 </p>

<p>@ConditionalOnMissingBean 表示如果容器中不存在对应的 Bean 才会去执行并创建对应的 B an。所以，在应用可以通过 @Configuration 等方式自定义创建对应的 Bean，以替换默认实现。 </p>

### 2.2 starter</h3>

<p>我们会看到 starter 项目中，仅存在一个 pom.xml，没有其它内容。 </p>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">&lt;project ...&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> &lt;modelVersi
n&gt;4.0.0&lt;/modelVersion&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> &lt;groupId&gt;
com.baeldung&lt;/groupId&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> &lt;artifactId&g
;greeter-spring-boot-starter&lt;/artifactId&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> &lt;version&gt;

```

```

0.1-SNAPSHOT</version>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;properties
gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;project.bu
ld.sourceEncoding>UTF-8</project.build.sourceEncoding>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;greeter.v
rsion>0.0.1-SNAPSHOT</greeter.version>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;spring-b
oot.version>1.5.2.RELEASE</spring-boot.version>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/properties
gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;dependenci
s>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;depende
cy>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;groupI
d>org.springframework.boot</groupId>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;artifact
Id>spring-boot-starter</artifactId>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;versio
n>${spring-boot.version}</version>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/depend
ncy>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;depende
cy>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;groupI
d>com.baeldung</groupId>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;artifact
Id>greeter-spring-boot-autoconfigure</artifactId>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;versio
n>${project.version}</version>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/depend
ncy>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;depende
cy>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;groupI
d>com.baeldung</groupId>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;artifact
Id>greeter</artifactId>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;versio
n>${greeter.version}</version>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/depend
ncy>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/dependenc
es>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/project>
</span> </span> </code> </pre>

```

该 pom.xml 规定了 starter 的名称叫：greeter-spring-boot-starter，以及使用的 greeter 库与 spring-boot 相关依赖的版本。

这里需要注意的是，我们平时使用的是 spring-boot-starter-xxx，是因为我们使用的 starter 大

分是 spring-boot 官方提供的 starter。而在实际工程中内部创建的 starter，官方是不建议按照这种方式进行命名的，以便为后续 Spring Boot 官方支持提供可能性。所以这里的命名方式是 xxx-spring-boot-starter。</p></div>
<div data-bbox="77 101 434 118" data-label="Section-Header"><h2 id="3--参考资料">3. 参考资料</h2></div>
<div data-bbox="77 117 126 132" data-label="Text"><ol></ol></div>
<div data-bbox="77 132 917 195" data-label="Text"><li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fdocs.spring.io%2Fspring-boot%2Fdocs%2F2.2.2.RELEASE%2Freference%2Fhtml%2Fusing-spring-boot.html%23using-boot-starter" target="\_blank" rel="nofollow ugc">https://docs.spring.io/spring-boot/docs/2.2.2.RELEASE/reference/html/using-spring-boot.html#using-boot-starter</a></li></div>
<div data-bbox="77 194 916 271" data-label="Text"><li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fdocs.spring.io%2Fspring-boot%2Fdocs%2F2.2.2.RELEASE%2Freference%2Fhtml%2Fspring-boot-features.html%23boot-features-custom-starter" target="\_blank" rel="nofollow ugc">https://docs.spring.io/spring-boot/docs/2.2.2.RELEASE/reference/html/spring-boot-features.html#boot-features-custom-starter/a></li></div>
<div data-bbox="77 270 917 317" data-label="Text"><li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.baeldung.com%2Fspring-boot-custom-starter" target="\_blank" rel="nofollow ugc">https://www.baeldung.com/spring-boot-custom-starter</a></li></div>
<div data-bbox="77 316 916 379" data-label="Text"><li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Feugenp%2Ftutorials%2Ftree%2Fmaster%2Fspring-boot-custom-starter" target="\_blank" rel="nofollow ugc">https://github.com/eugenp/tutorials/tree/master/spring-boot-custom-starter</a></li></ol></div>
<div data-bbox="635 936 930 952" data-label="Page-Footer"><p>原文链接: <a href="https://www.baeldung.com/spring-boot-custom-starter">如何实现一个 Spring Boot Starter</a></p></div>