第十一章：Spring 资源管理

小马哥 • mercyblitz

Spring 资源管理

1. 引入动机
2. Java 标准资源管理
3. Spring 资源接口
4. Spring 内建 Resource 实现
5. Spring Resource 接口扩展
6. Spring 资源加载器
7. Spring 通配路径资源加载器
8. Spring 通配路径资源扩展
9. 依赖注入Spring Resource
10. 依赖注入 ResourceLoader
11. 面试题精选

引入动机

- 为什么 Spring 不使用 Java 标准资源管理，而选择重新发明轮子？
- Java 标准资源管理强大，然而扩展复杂，资源存储方式并不统一
- Spring 要自立门户（重要的话，要讲三遍）
- Spring “抄”、“超” 和 “潮”

Java 标准资源管理

- Java 标准资源定位

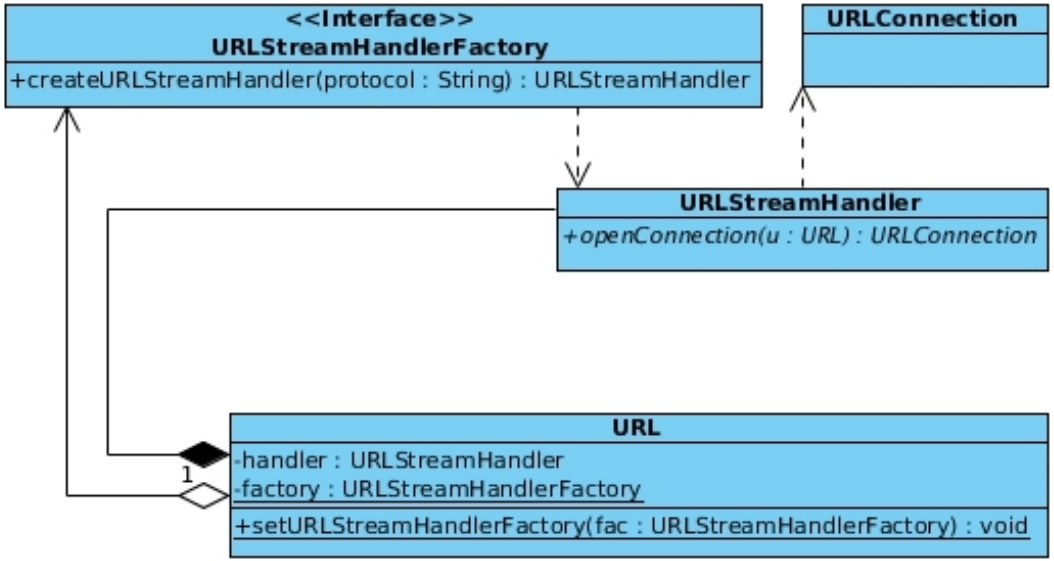
职责	说明
面向资源	文件系统、artifact（jar、war、ear 文件）以及远程资源（HTTP、FTP 等）
API 整合	java.lang.ClassLoader#getResource、java.io.File 或 java.net.URL
资源定位	java.net.URL 或 java.net.URI
面向流式存储	java.net.URLConnection
协议扩展	java.net.URLStreamHandler 或 java.net.URLStreamHandlerFactory

Java 标准资源管理

- Java URL 协议扩展
  - 基于 java.net.URLStreamHandlerFactory
  - 基于 java.net.URLStreamHandler

Java 标准资源管理

- 基于 java.net.URLStreamHandlerFactory 扩展协议



Java 标准资源管理

- 基于 java.net.URLStreamHandler 扩展协议
  - JDK 1.8 内建协议实现

协议	实现类
file	sun.net.www.protocol.file.Handler
ftp	sun.net.www.protocol.ftp.Handler
http	sun.net.www.protocol.http.Handler
https	sun.net.www.protocol.https.Handler
jar	sun.net.www.protocol.jar.Handler
mailto	sun.net.www.protocol.mailto.Handler
netdoc	sun.net.www.protocol.netdoc.Handler

Java 标准资源管理

- 基于 java.net.URLStreamHandler 扩展协议
  - 实现类名必须为 “Handler”

实现类命名规则	说明
默认	sun.net.www.protocol.${protocol}.Handler
自定义	通过 Java Properties java.protocol.handler.pkgs 指定实现类包名，实现类名必须为“Handler”。如果存在多包名指定，通过分隔符 “|”

Spring 资源接口

- 资源接口

类型	接口
输入流	org.springframework.core.io.InputStreamSource
只读资源	org.springframework.core.io.Resource
可写资源	org.springframework.core.io.WritableResource
编码资源	org.springframework.core.io.support.EncodedResource
上下文资源	org.springframework.core.io.ContextResource

Spring 内建 Resource 实现

- 内建实现

资源来源	资源协议	实现类
Bean 定义	无	org.springframework.beans.factory.support.BeanDefinitionResource
数组	无	org.springframework.core.io.ByteArrayResource
类路径	classpath:/	org.springframework.core.io.ClassPathResource
文件系统	file:/	org.springframework.core.io.FileSystemResource
URL	URL 支持的协议	org.springframework.core.io.UrlResource
ServletContext	无	org.springframework.web.context.support.ServletContextResource

Spring Resource 接口扩展

- 可写资源接口
  - org.springframework.core.io.WritableResource
    - org.springframework.core.io.FileSystemResource
    - org.springframework.core.io.FileUrlResource（@since 5.0.2）
    - org.springframework.core.io.PathResource（@since 4.0 & @Deprecated）

- 编码资源接口
  - org.springframework.core.io.support.EncodedResource

Spring 资源加载器

- Resource 加载器
  - org.springframework.core.io.ResourceLoader
    - org.springframework.core.io.DefaultResourceLoader
      - org.springframework.core.io.FileSystemResourceLoader
      - org.springframework.core.io.ClassRelativeResourceLoader
      - org.springframework.context.support.AbstractApplicationContext

Spring 通配路径资源加载器

- 通配路径 ResourceLoader
  - org.springframework.core.io.support.ResourcePatternResolver
    - org.springframework.core.io.support.PathMatchingResourcePatternResolver

- 路径匹配器
  - org.springframework.util.PathMatcher
    - Ant 模式匹配实现 - org.springframework.util.AntPathMatcher

Spring 通配路径资源扩展

- 实现 org.springframework.util.PathMatcher

- 重置 PathMatcher
  - PathMatchingResourcePatternResolver#setPathMatcher

依赖注入Spring Resource

- 基于 @Value 实现
  - 如：

```
@Value(“classpath:/...”)
private Resource resource;
```

依赖注入 ResourceLoader

- 方法一：实现 ResourceLoaderAware 回调
- 方法二：@Autowired 注入 ResourceLoader
- 方法三：注入 ApplicationContext 作为 ResourceLoader

面试题

沙雕面试题 - Spring 配置资源中有哪些常见类型？

答：

- XML 资源
- Properties 资源
- YAML 资源

面试题

996 面试题 - 请例举不同类型 Spring 配置资源？

答：

- XML 资源
  - 普通 Bean Definition XML 配置资源 - *.xml
  - Spring Schema 资源 - *.xsd
- Properties 资源
  - 普通 Properties 格式资源 - *.properties
  - Spring Handler 实现类映射文件 - META-INF/spring.handlers
  - Spring Schema 资源映射文件 - META-INF/spring.schemas
- YAML 资源
  - 普通 YAML 配置资源 - *.yaml 或 *.yml

面试题

劝退面试题 - Java 标准资源管理扩展的步骤？

答：

- 简易实现
  - 实现 URLStreamHandler 并放置在 sun.net.www.protocol.${protocol}.Handler 包下
- 自定义实现
  - 实现 URLStreamHandler
  - 添加 -Djava.protocol.handler.pkgs 启动参数，指向 URLStreamHandler 实现类的包下
- 高级实现
  - 实现 URLStreamHandlerFactory 并传递到 URL 之中