



链滴

Solo Markdown 渲染

作者: [88250](#)

原文链接: <https://ld246.com/article/1544880962218>

来源网站: [链滴](#)

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

本文是《Solo 从设计到实现》的一个章节，该系列文章将介绍 Solo 这款 Java 博客系统是如何从无有的，希望大家能通过它对 Solo 从设计到实现有个直观地了解、能为想参与贡献的人介绍清楚项目也希望能给重复发明——重新定义博客系统的人做个参考 ☺

Markdown 渲染指的是将 Markdown 文本转换成 HTML 的过程。

引擎选择

Solo 默认是通过 flexmark-java 进行 Markdown 渲染的。如果启用了独立进程 Lute HTTP 则优先用 Lute HTTP 引擎渲染。建议使用 Lute 引擎，因为 Lute 是针对中文语境优化的 Markdown 引擎。

```
if (LUTE_AVAILABLE) {
    try {
        html = toHtmlByLute(markdownText);
    } catch (final Exception e) {
        LOGGER.log(Level.WARN, "Failed to use Lute [" + LUTE_ENGINE_URL + "] for markdown
md=" + StringUtils.substring(markdownText, 0, 256) + "]: " + e.getMessage());
    }
}

if (StringUtils.isBlank(html)) {
    html = toHtmlByFlexmark(markdownText);
}
```

关于 Lute 引擎更多的细节请看 <https://hacpai.com/article/1567047822949>。

超时控制

Solo 的 Markdown 渲染是在一个新建线程中完成的，获取处理结果带超时控制（默认 10s）。这样是为了防止一些无法预期的问题，比如引擎缺陷或是文本怪异导致渲染太耗时，从而耗尽服务线程。

站外链接处理

Markdown 渲染后得到的 HTML 还会通过 jsoup 解析一次，将其中的 a 标签获取出来，如果发现是外链接，就将其 target 设置为 _blank，点击时以新页签方式打开。

```
final Document doc = Jsoup.parse(html);
doc.select("a").forEach(a -> {
    final String src = a.attr("href");
    if (!StringUtils.startsWithIgnoreCase(src, Latkes.getServePath())) {
        a.attr("target", "_blank");
    }
});
```

结果缓存

每次 Markdown 渲染后的最终 HTML 都会放入缓存，下次进行渲染时先从缓存中获取。缓存键是将 Markdown 文本进行 MD5 哈希得到的，虽然存在一定碰撞的可能，但是目前的结果来看，这个可能性是非常低的。