

Lettuce 客户端连接池

现象

- 测试发现：

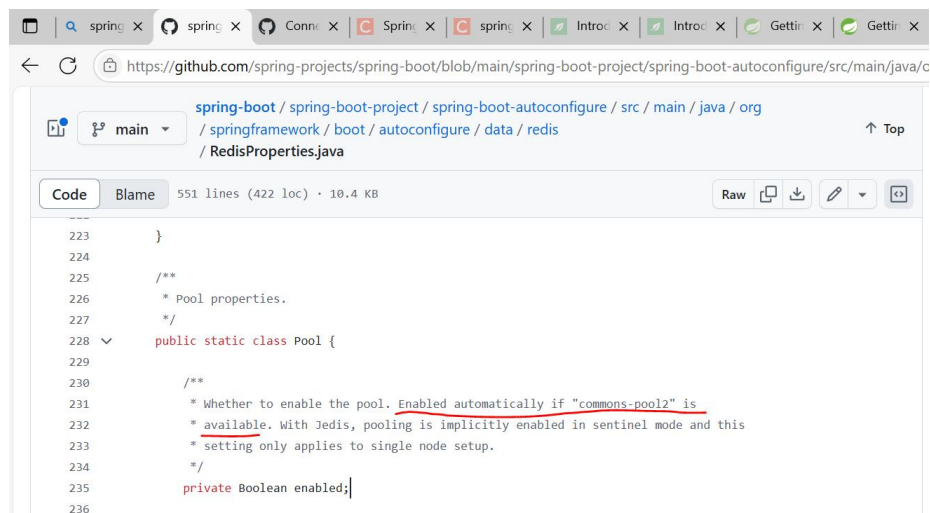
1. Spring Boot Data 使用 Jedis 的时候，无需配置连接池参数 enabled 即可使连接池生效
2. Spring Boot Data 使用 Lettuce 的时候，**必须**配置连接池参数 enabled 才使连接池生效

原因

- 根据新版本的源码注释（旧版本没注释...），maven项目中如果包含 commons-pool2 包，则自动 enable 连接池。查看maven依赖发现：

1. jedis 依赖包包含了 commons-pool2 包
2. lettuce 依赖包没有包含 commons-pool2 包

这就是原因。

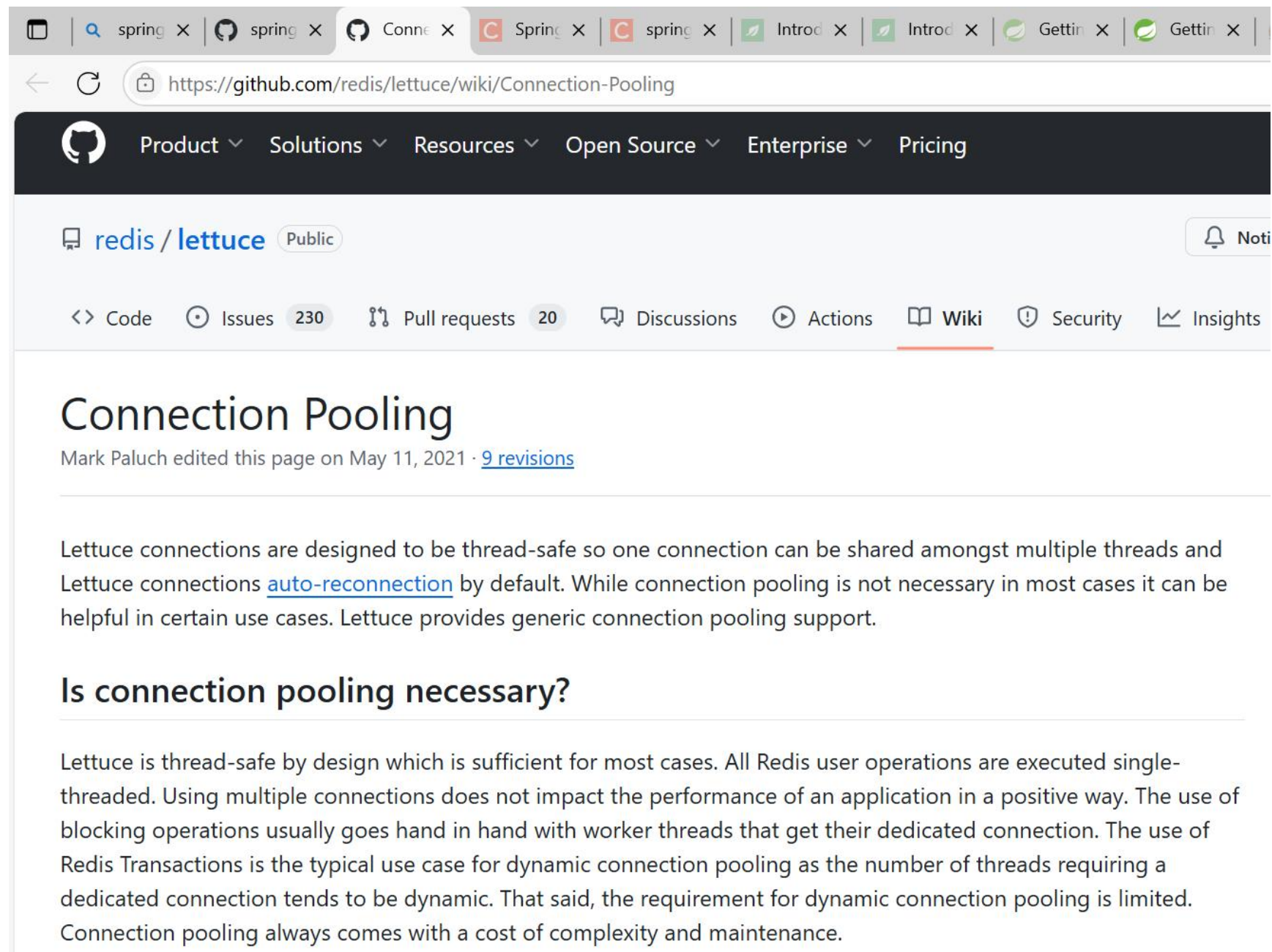


```
223 }
224
225 /**
226  * Pool properties.
227  */
228 public static class Pool {
229
230     /**
231      * Whether to enable the pool. Enabled automatically if "commons-pool2" is
232      * available. With Jedis, pooling is implicitly enabled in sentinel mode and this
233      * setting only applies to single node setup.
234      */
235     private Boolean enabled;
```

引申

- Jedis 由于线程不安全，在多线程环境下必须使用连接池（这不仅是性能问题，更重要的是线程安全问题）。换句话说，使用没有连接池的Jedis就是错误的。这也是为什么 jedis 依赖包包含了 commons-pool2 包
- 相反，Lettuce 无须使用连接池（甚至Lettuce官方不建议使用连接池），因为Lettuce 是异步的且底层使用Netty，利用多路复用（multiplexing）原理管理多个redis请求。单个连接就支持多线程操作。使用连接池并不会提高性能
- 有两种情况，Lettuce 需要使用连接池：① 阻塞操作 ② 事务性操作

- Lettuce 官网wiki



The screenshot shows a web browser with multiple tabs open, including 'spring', 'Conne', 'Spring', and 'Intro'. The active tab displays the GitHub page for 'redis/lettuce' with the title 'Connection Pooling'. The page header includes the GitHub logo and navigation links: Product, Solutions, Resources, Open Source, Enterprise, and Pricing. Below the header, the repository name 'redis / lettuce' is shown with a 'Public' badge. A navigation bar contains links for Code, Issues (230), Pull requests (20), Discussions, Actions, Wiki (highlighted with a red underline), Security, and Insights. The main content area features the title 'Connection Pooling' and a note that 'Mark Paluch edited this page on May 11, 2021 · 9 revisions'. The text explains that Lettuce connections are thread-safe and designed for sharing among multiple threads, with a link to 'auto-reconnection'. It states that while connection pooling is not necessary in most cases, it can be helpful in certain use cases. A section titled 'Is connection pooling necessary?' follows, explaining that Lettuce is thread-safe by design and that using multiple connections does not impact performance. It also mentions that blocking operations usually go hand in hand with worker threads that get their dedicated connection, and that Redis Transactions are a typical use case for dynamic connection pooling. The text concludes that connection pooling always comes with a cost of complexity and maintenance.

spring x spring x Conne x Spring x spring x Intro x Intro x Gettin x Gettin x

← ↻ https://github.com/redis/lettuce/wiki/Connection-Pooling

Product Solutions Resources Open Source Enterprise Pricing

redis / lettuce Public

<> Code Issues 230 Pull requests 20 Discussions Actions Wiki Security Insights

Connection Pooling

Mark Paluch edited this page on May 11, 2021 · [9 revisions](#)

Lettuce connections are designed to be thread-safe so one connection can be shared amongst multiple threads and Lettuce connections [auto-reconnection](#) by default. While connection pooling is not necessary in most cases it can be helpful in certain use cases. Lettuce provides generic connection pooling support.

Is connection pooling necessary?

Lettuce is thread-safe by design which is sufficient for most cases. All Redis user operations are executed single-threaded. Using multiple connections does not impact the performance of an application in a positive way. The use of blocking operations usually goes hand in hand with worker threads that get their dedicated connection. The use of Redis Transactions is the typical use case for dynamic connection pooling as the number of threads requiring a dedicated connection tends to be dynamic. That said, the requirement for dynamic connection pooling is limited. Connection pooling always comes with a cost of complexity and maintenance.

引申

- Redis官网

The screenshot shows a web browser with multiple tabs open, including 'StatefulRe', 'Asynchron', 'Connect t', 'Stateful C', 'springbo', 'LettuceCo', 'spring-bo', 'Connectic', and 'Getting Si'. The address bar shows the URL `https://redis.io/docs/latest/develop/clients/lettuce/connect/`. The website header features the Redis logo, navigation links for Products, Solutions, Support, Company, Docs, and Pricing, a search icon, and links for Login, Book a meeting, and a red 'Try Redis' button. On the left, a sidebar titled 'Develop with Redis' contains links for 'What's new?', 'Quick starts', and 'Client tools'. The main content area is titled 'Connection pooling' and contains the following text:

A typical approach with Lettuce is to create a single `RedisClient` instance and reuse it to establish connections to your Redis server(s). These connections are multiplexed; that is, multiple commands can be run concurrently over a single or a small set of connections, making explicit pooling less practical. See [Connection pools and multiplexing](#) for more information.

Lettuce provides pool config to be used with Lettuce asynchronous connection methods.

Ref

- <https://github.com/spring-projects/spring-boot/blob/main/spring-boot-project/spring-boot-autoconfigure/src/main/java/org/springframework/boot/autoconfigure/data/redis/RedisProperties.java>
- <https://github.com/redis/lettuce/wiki/Connection-Pooling>
- <https://redis.io/docs/latest/develop/clients/pools-and-muxing/>
- <https://redis.io/docs/latest/develop/clients/lettuce/connect/>