

Please visit website: <http://cxyroad.com>

大厂Java面试题：两道来自京东的关于MyBatis执行器的面试题

大家好，我是[王有志](<http://cxyroad.com/>
"https://www.yuque.com/wangyouzhi-
u3woi/cr7d5y/uw8c5iyvpqngqzmg")。
今天给大家带来两道来自于京东关于的 MyBatis 面试题：

* **MyBatis 提供了哪些执行器 (Executor) ? 它们有什么区别? **
* **Mybatis 中如何指定 Executor 的类型? **

MyBatis 提供了哪些执行器 (Executor) ? 它们有什么区别?

MyBatis 提供的 Executor

严格意义上 MyBatis 中提供了 4 种 Executor:

![image](<https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/e3ece0e88b7b4f4e99ec902ee492741d~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2182&h=660&s=34415&e=png&b=1e1f22>)

其中具有执行 SQL 语句能力的是继承自 BaseExecutor 的 3 种 Executor:

- * **SimpleExecutor, 最基础的 Executor, 每次执行 SQL 语句时都会创建 Statement 实例对象, 完成 SQL 语句的执行后关闭 Statement 实例对象, 无任何性能上的优化**;
- * **ReuseExecutor, 提供复用 Statement 能力的 Executor, ReuseExecutor 会将 Statement 缓存到 Map<String, Statement> 实例对象中, 其中 key 是 String 类型的 SQL 语句, 而 value 是 SQL 语句的 Statement 对象, 这样避免了频繁创建和销毁 Statement 带来的性能损耗**;
- * **BatchExecutor, 提供了批量处理 Statement 的能力, 在执行 update 语句时, 将所有的 Statement 对象添加到 BatchExecutor 的 statementList 对象中, 等到执行** `SqlSession#commit` **时 统一提交, 避免了频繁与数据库交互带来的性能损耗**。

以上 3 种 Executor 除了自身的特点外，**它们还具备抽象类 BaseExecutor 提供的一级缓存的能力**。

CachingExecutor 本身不具备执行 SQL 语句的能力，它提供了对 MyBatis 二级缓存的支持。CachingExecutor 会持有一个继承自 BaseExecutor 的实例对象，CachingExecutor 在执行 SQL 语句时会调用自身持有的 Executor 实例对象来完成 SQL 语句的执行。

MyBatis 默认的 Executor

如果我们刨除不能够独立执行 SQL 语句的 CachingExecutor 的话，MyBatis 默认的 Executor 是 SimpleExecutor。如果把 CachingExecutor 也算在内的话，由于 MyBatis 是默认开启二级缓存的，因此默认的 Executor 就是 CachingExecutor。

这点可以在 MyBatis 的源码中得到印证：

```
...
public class Configuration {

    protected boolean cacheEnabled = true;

    public Executor newExecutor(Transaction transaction, ExecutorType
executorType) {
        executorType = executorType == null ? defaultExecutorType :
executorType;
        Executor executor;
        if (ExecutorType.BATCH == executorType) {
            executor = new BatchExecutor(this, transaction);
        } else if (ExecutorType.REUSE == executorType) {
            executor = new ReuseExecutor(this, transaction);
        } else {
            executor = new SimpleExecutor(this, transaction);
        }
        if (cacheEnabled) {
            executor = new CachingExecutor(executor);
        }
        return (Executor) interceptorChain.pluginAll(executor);
    }
}
```

...

源码的第 5~14 行中，是根据配置的 ExecutorType 创建继承自 BaseExecutor 的 Executor 实例对象，第 15~17 行中，判断了 cacheEnabled 的配置情况，并决定是否要使用 CachingExecutor。

Mybatis 中如何指定 Executor 的类型？

MyBatis 配置 Executor 的方式

MyBatis 中有两种方式指定 Executor 的类型：

- * 通过 MyBatis 的核心配置指定 MyBatis 的 Executor 类型；
- * 创建 SqlSession 时可以指定指定 Executor 的类型。

MyBatis 核心配置文件中配置 Executor

可以在核心配置文件 mybatis-config.xml 中通过 settings 元素来配置 Executor 的类型，例如：

...

```
<configuration>
  <!-- 省略 -->

  <settings>
    <setting name="defaultExecutorType" value="REUSE"/>
  </settings>

  <!-- 省略 -->
</configuration>
```

...

这样我们通过 mybatis-config.xml 创建的 SqlSession 中都会使用 ReuseExecutor。

创建 SqlSession 时指定 Executor

除了的核心配置文件 mybatis-config.xml 中配置 Executor 的类型外，还在获取 MyBatis 的 SqlSession 实例对象时指定 Executor 的类型，如下：

```
...  
    Reader mysqlReader = Resources.getResourceAsReader("mybatis-  
config.xml");  
    sqlSessionFactory = new  
SqlSessionFactoryBuilder().build(mysqlReader);  
  
    // 指定 Executor的类型  
    SqlSession sqlSession =  
sqlSessionFactory.openSession(ExecutorType.SIMPLE);  
...  
  
**如果同时在核心配置文件 mybatis-coonfig.xml 中配置了 Executor 的类型  
，且在创建 SqlSession 的实例对象时也指定了 Executor 的类型，此时以创建  
SqlSession 的实例对象时指定的 Executor 类型为准**。
```

...

好了，今天的内容就到这里了，如果本文对你有帮助的话，希望多多点赞支持，如果文章中出现任何错误，还请批评指正。最后欢迎大家分享硬核 Java 技术的金融摸鱼侠[王有志](<http://cxyroad.com/>
"https://www.yuque.com/wangyouzhi-
u3woi/cr7d5y/uw8c5iyvpqngpzmng"), 我们下次再见!

原文链接: <https://juejin.cn/post/7361949850327744549>