

mybatis-plus 实践及架构原理



mybatis 增强工具包，简化 CURD 操作。

启动加载 XML 配置时注入单表 SQL 操作，为简化开发工作提供生产率而生。mybatis-plus 只做增强不做改变，这里不提倡 SQL 写在代码中。

Maven 坐标 <http://search.maven.org/#search%7Cga%7C1%7Cmybatis-plus>

```
1 <dependency>
2   <groupId>com.baomidou</groupId>
3   <artifactId>mybatis-plus</artifactId>
4   <version>maven 官方最新版本号为 准</version>
5 </dependency>
```

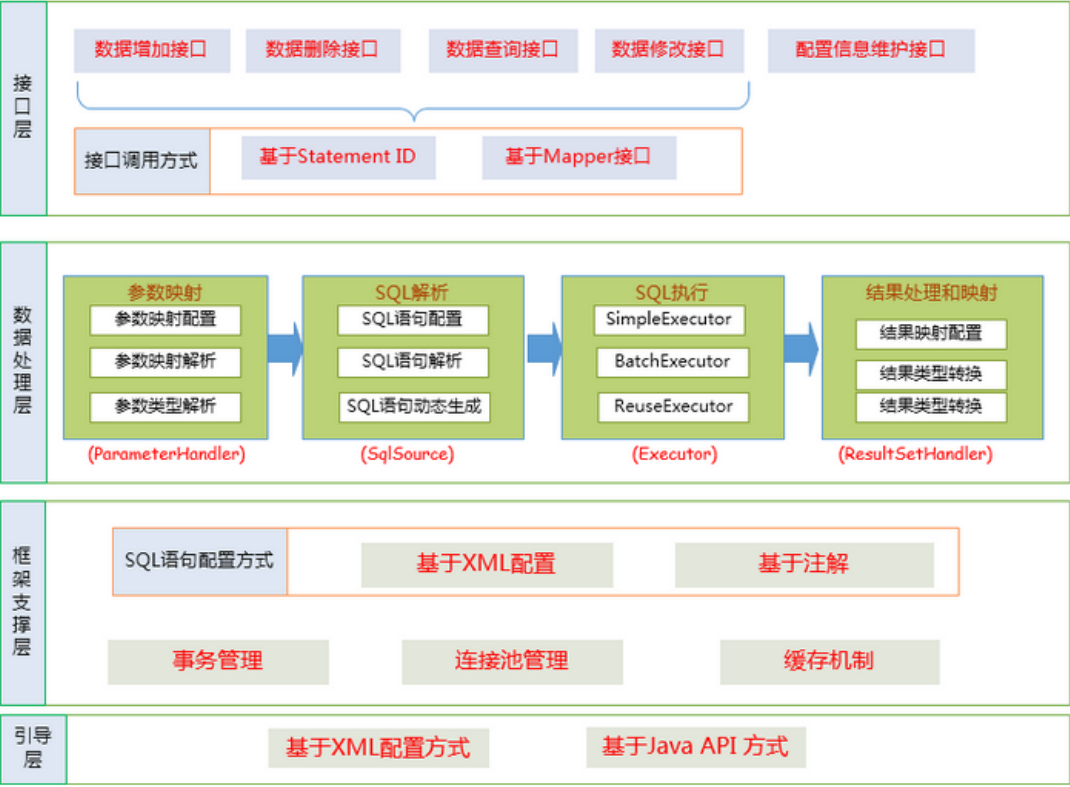
功能

- 1、单表 CURD（简单 + 批量）操作，自动完成。
- 2、分页插件，Count 查询自动或自定义 SQL 查询。
- 3、Spring 根据不同环境加载不同配置支持。

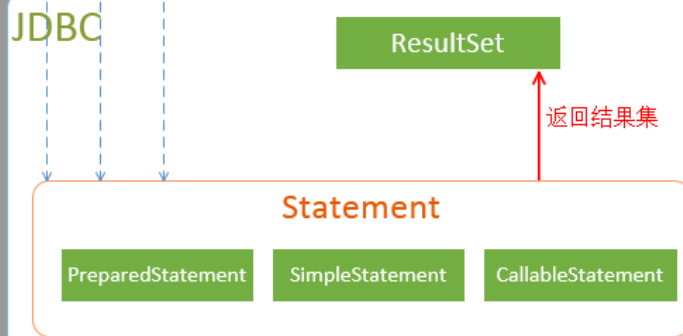
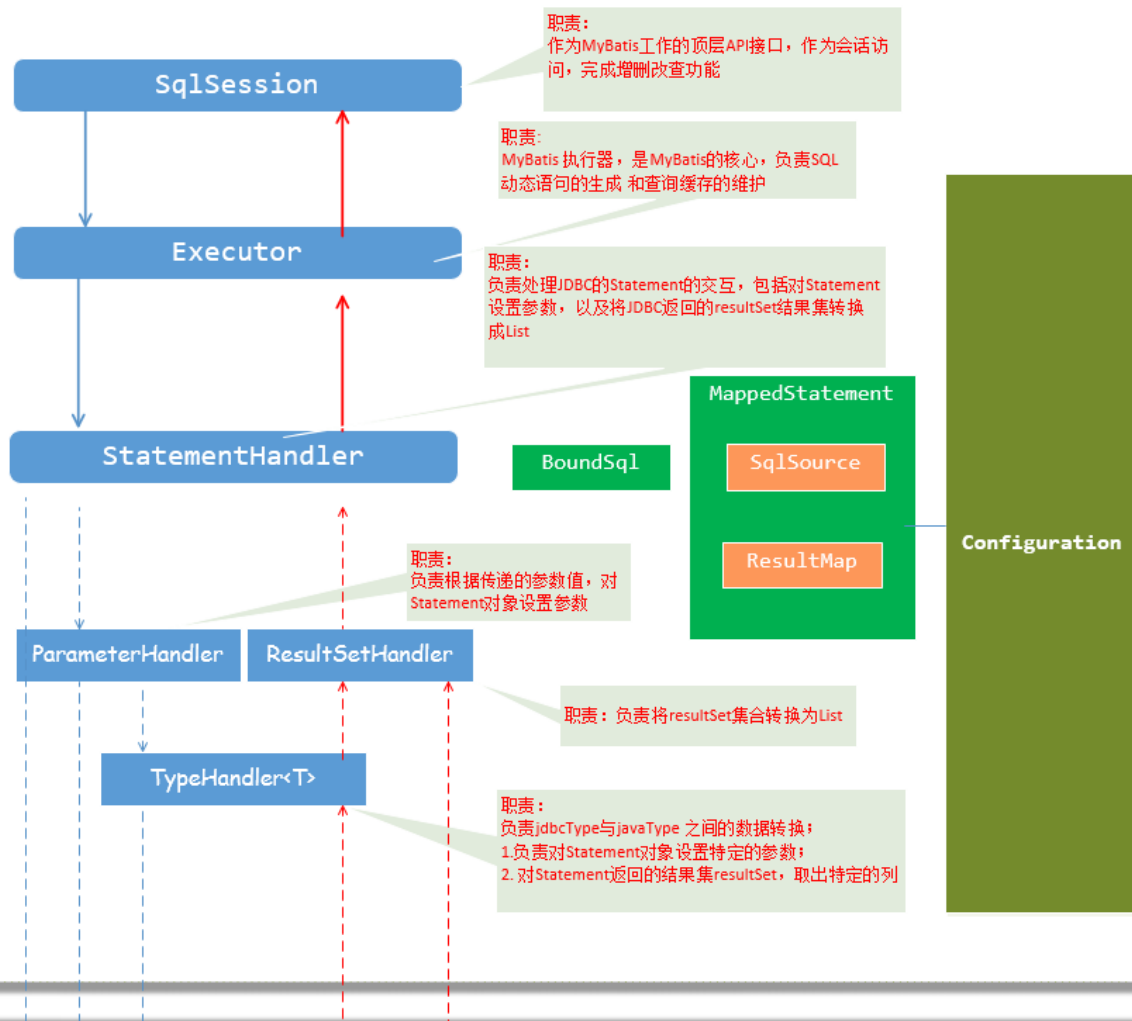
使用

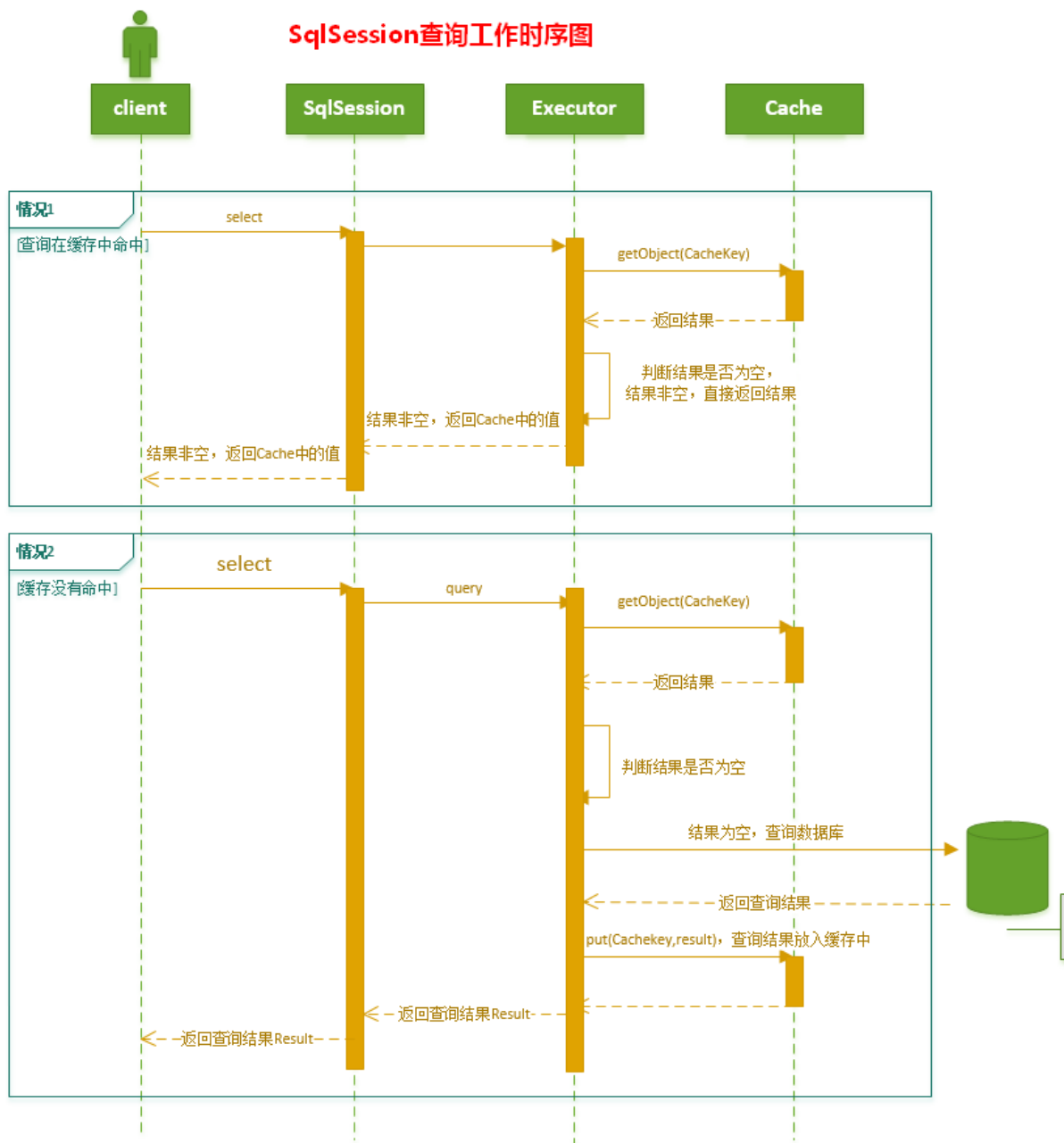
- 1、代码自动生成，查看类 `com.baomidou.mybatisplus.test.AutoGeneratorTest`
- 2、使用方法，查看类 `com.baomidou.mybatisplus.test.UserMapperTest`

mybatis 架构图

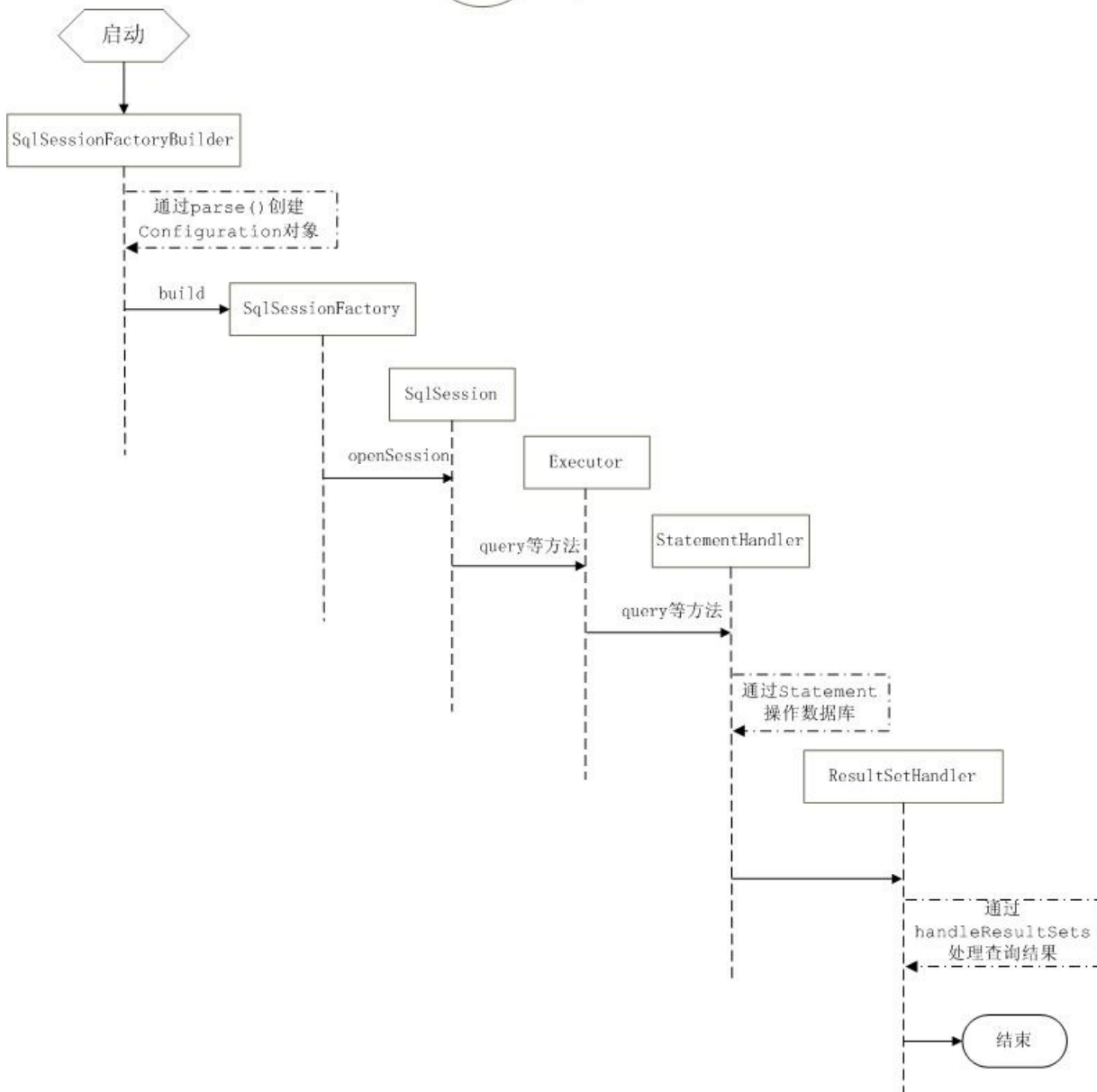


MyBatis层次结构

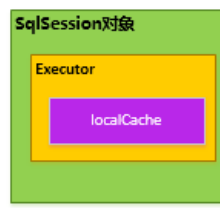




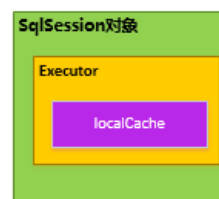
Mybatis整体流程图



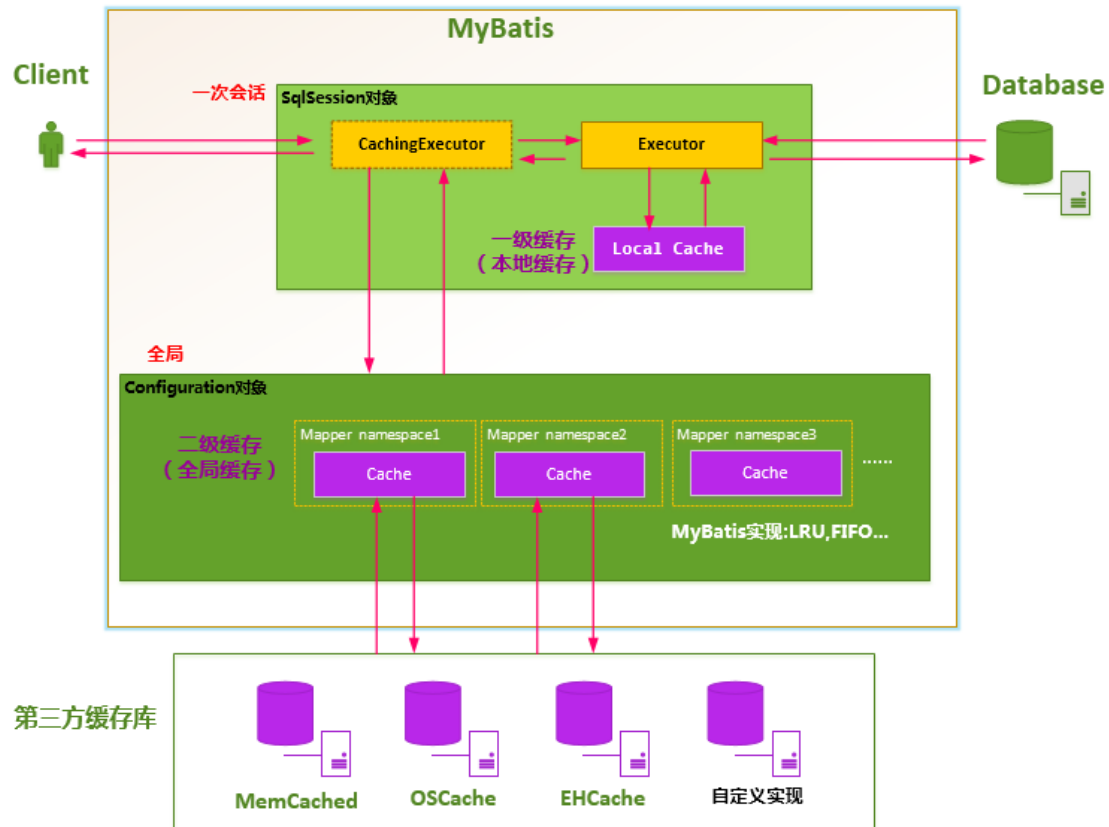
开启一个会话：
这时会创建一个SqlSession对象，
并且为SqlSession创建一个
Executor对象，而Executor内部持
有一个Cache实例对象



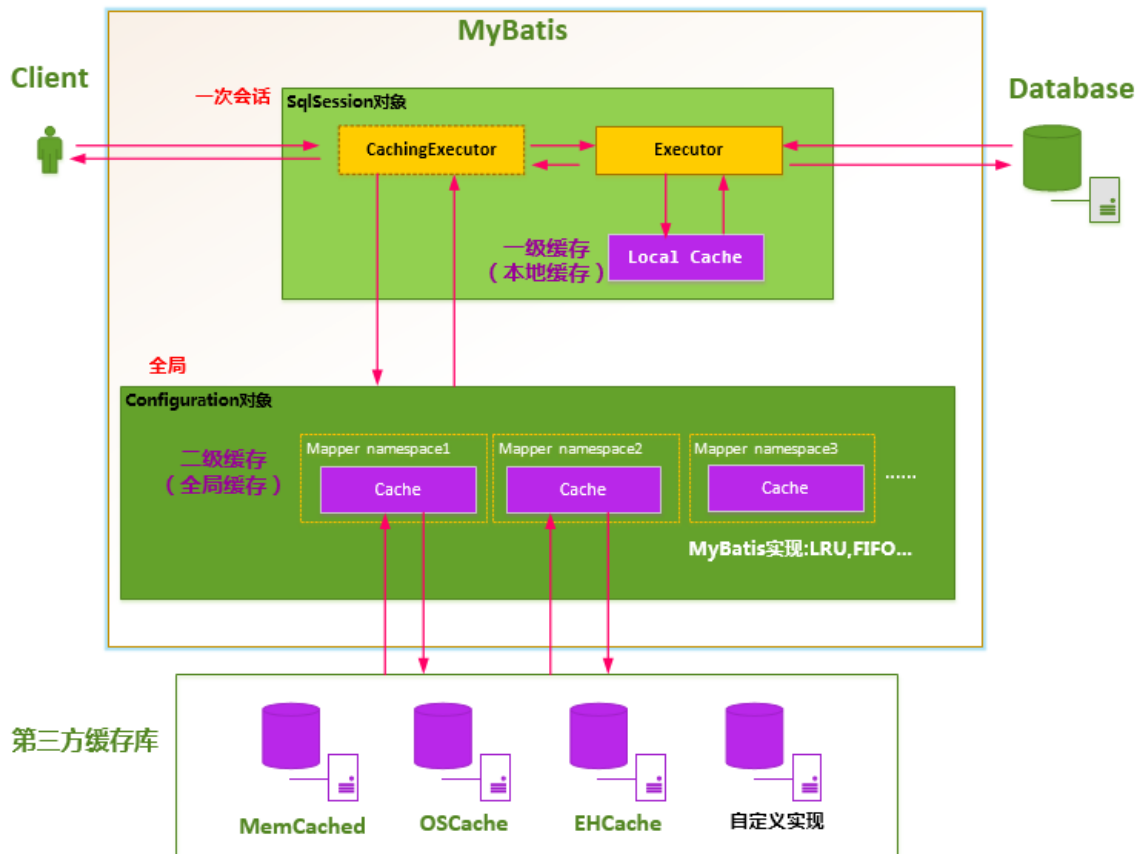
会话结束：
释放和销毁sqlSession，
Executor对象和内部的Cache对象
也会一并销毁



会话1 会话2 生命周期

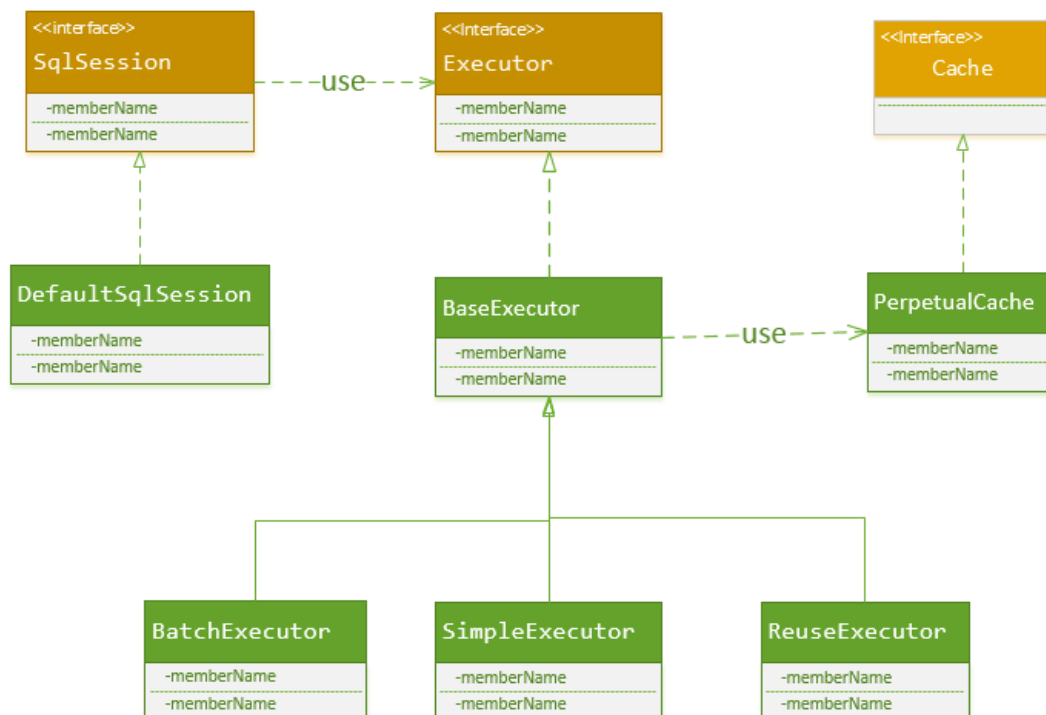


MyBatis缓存机制示意图



MyBatis缓存机制示意图

SqlSession和Executor以及Cache的关系类图



List<?> | int | void sqlSession.

select
selectList
selectMap
selectOne
update
delete
insert

(statementId [,parameterObject])

传统的MyBatis工作模式

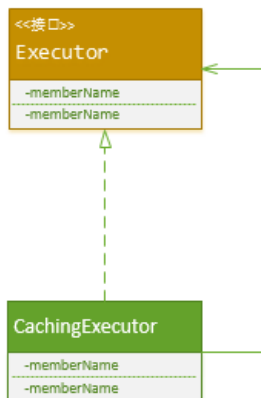
com.foo...XXMapper.java

```
public interface XXXMapper{  
    public List<?> querySome(...);  
  
    public int updateSome(...);  
  
    public int insertSome(...);  
  
    public int deleteSome(...);  
    ...  
}
```

com/foo.../XXMapper.xml

```
<mapper namespace= "com.foo... .XXMapper">  
  <select id="querySome" parameterType="..." resultMap="...">  
    ...  
  </select>  
  <update id="updateSome" parameterType="...">  
    ...  
  </update>  
  
  <insert id="insertSome" parameterType="..." >  
    ...  
  </insert>  
  
  <delete id="deleteSome" parameterType="..." >  
    ...  
  </delete>  
  ...  
</mapper>
```

Mapper接口和Mapper.xml配置文件之间的对应关系



运用decorator模式，使用CachingExecutor对Executor的功能进行了增强，增加了二级缓存的功能

未使用二级缓存的工作模式



使用二级缓存的工作模式

