使用Arthas抽丝剥茧深入应用

线上诊断利器之外

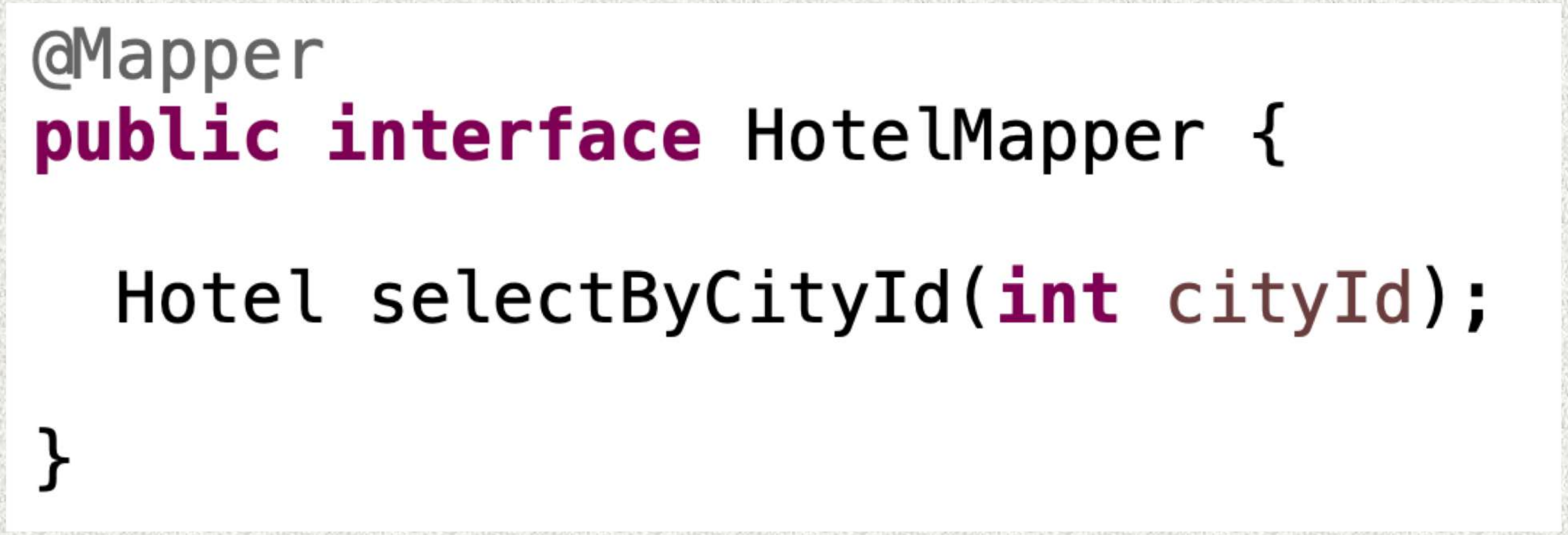
陈志轩/横云断岭

- 阿里巴巴-云原生-基础技术中台
- Spring Boot/Dubbo/Arthas
- Github: hengyunabc
- Email: hengyunabc@gmail.com
- Blog: http://hengyunabc.github.io/
- 公众号： 横云断岭的专栏

Arthas/阿尔萨斯

- Alibaba开源的Java诊断工具
- 2018-09开源，Star 19.7K
- Github: https://github.com/alibaba/arthas
- Wiki: https://alibaba.github.io/arthas/
- 国内镜像Wiki: https://arthas.gitee.io/
- Arthas开源交流QQ群：916328269
- Arthas开源交流钉钉群：21965291

一个简单的mybatis demo



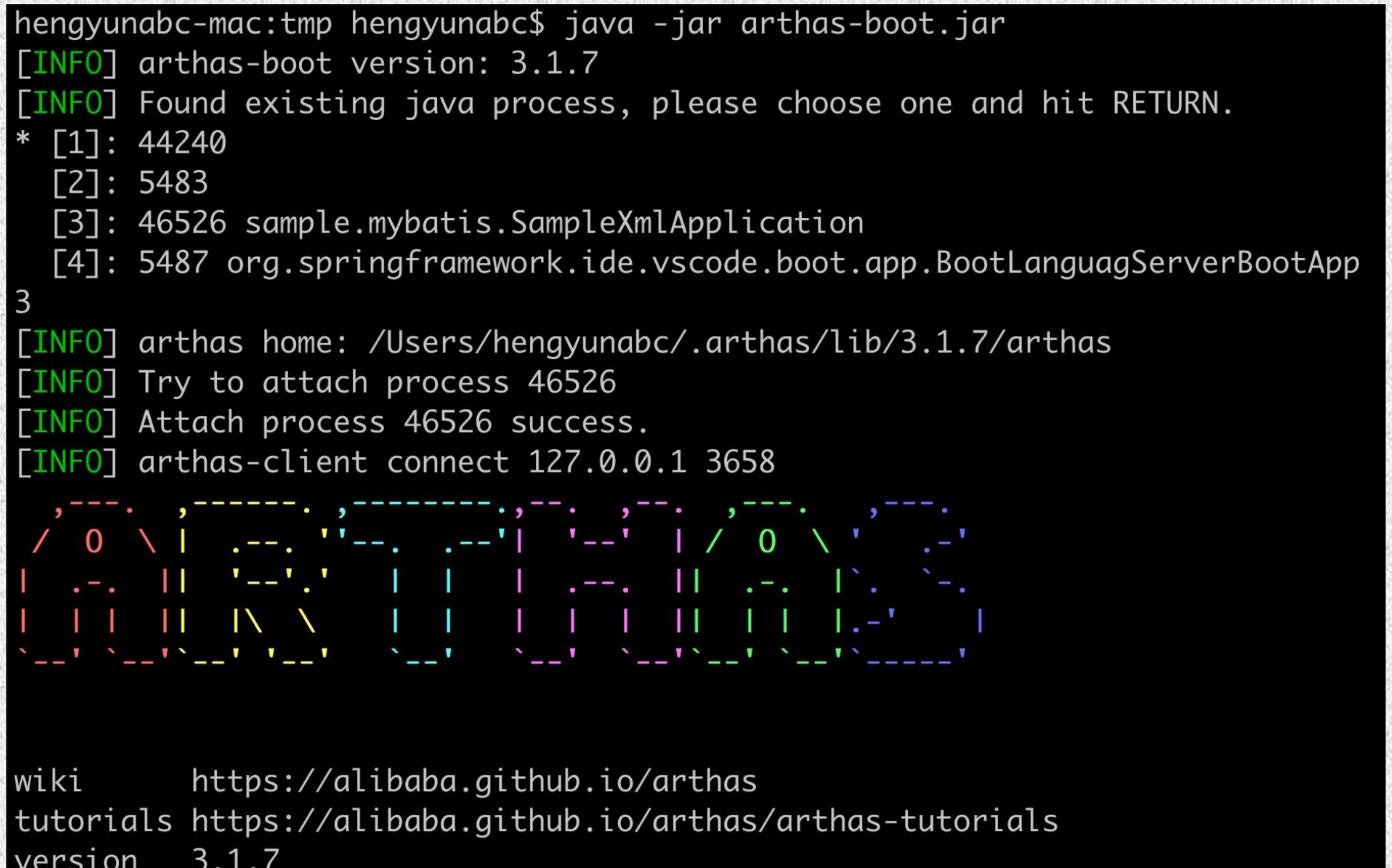
```java
@Mapper
public interface HotelMapper {

  Hotel selectByCityId(int cityId);

}
```

- 运行时，怎么查看 selectByCityId 具体执行的sql是怎样的？
- https://github.com/hengyunabc/arthas-mybatis-demo

Arthas快速开始

```
$ curl -O https://alibaba.github.io/arthas/arthas-boot.jar
$ java -jar arthas-boot.jar
```

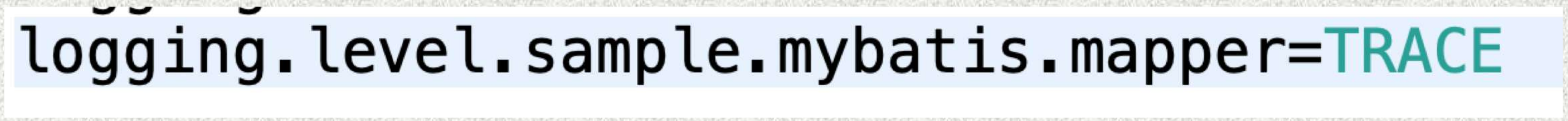
```
$ curl -O https://arthas.gitee.io/arthas-boot.jar
$ java -jar arthas-boot.jar
```



```
hengyunabc-mac:tmp hengyunabc$ java -jar arthas-boot.jar
[INFO] arthas-boot version: 3.1.7
[INFO] Found existing java process, please choose one and hit RETURN.
* [1]: 44240
  [2]: 5483
  [3]: 46526 sample.mybatis.SampleXmlApplication
  [4]: 5487 org.springframework.ide.vscode.boot.app.BootLanguagServerBootApp
3
[INFO] arthas home: /Users/hengyunabc/.arthas/lib/3.1.7/arthas
[INFO] Try to attach process 46526
[INFO] Attach process 46526 success.
[INFO] arthas-client connect 127.0.0.1 3658
  ,---.  ,------. ,--------.,--.  ,--.  ,---.   ,---.
 /  O  \ |  .--. ''--.  .--'|  '--'  | /  O  \ '   .-'
|  .-.  ||  '--'.'   |  |   |  .--.  ||  .-.  |`.  `-.
|  | |  ||  |\  \    |  |   |  |  |  ||  | |  |.-'    |
`--' `--'`--' '--'   `--'   `--'  `--'`--' `--'`-----'


wiki      https://alibaba.github.io/arthas
tutorials https://alibaba.github.io/arthas/arthas-tutorials
version   3.1.7
```

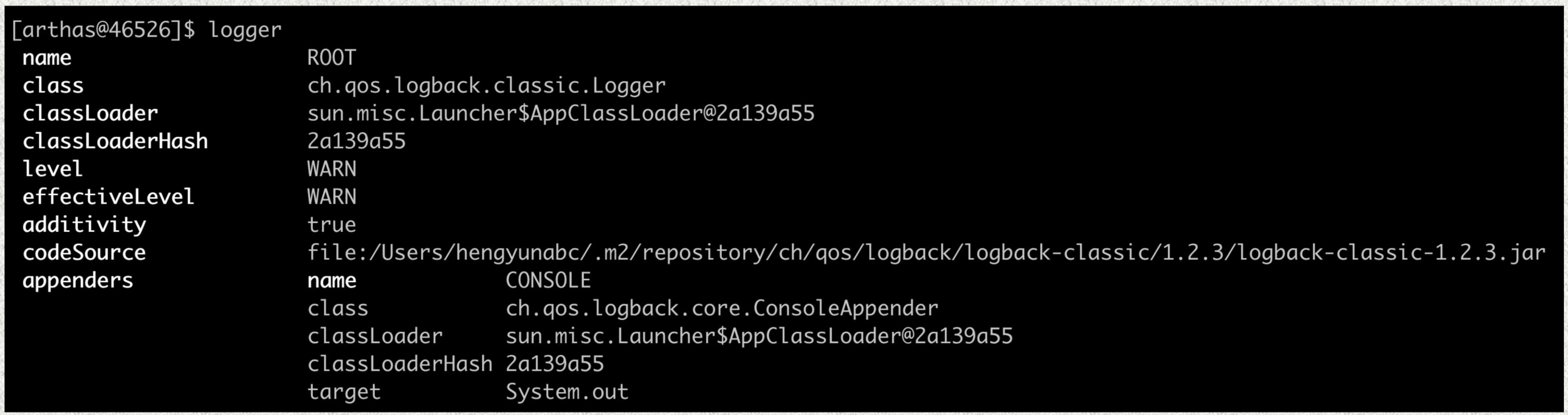
动态查看mybatis具体执行的sql

- mybatis本身的logger可以打印出sql



```
logging.level.sample.mybatis.mapper=TRACE
```

- 使用arthas查看应用的logger配置

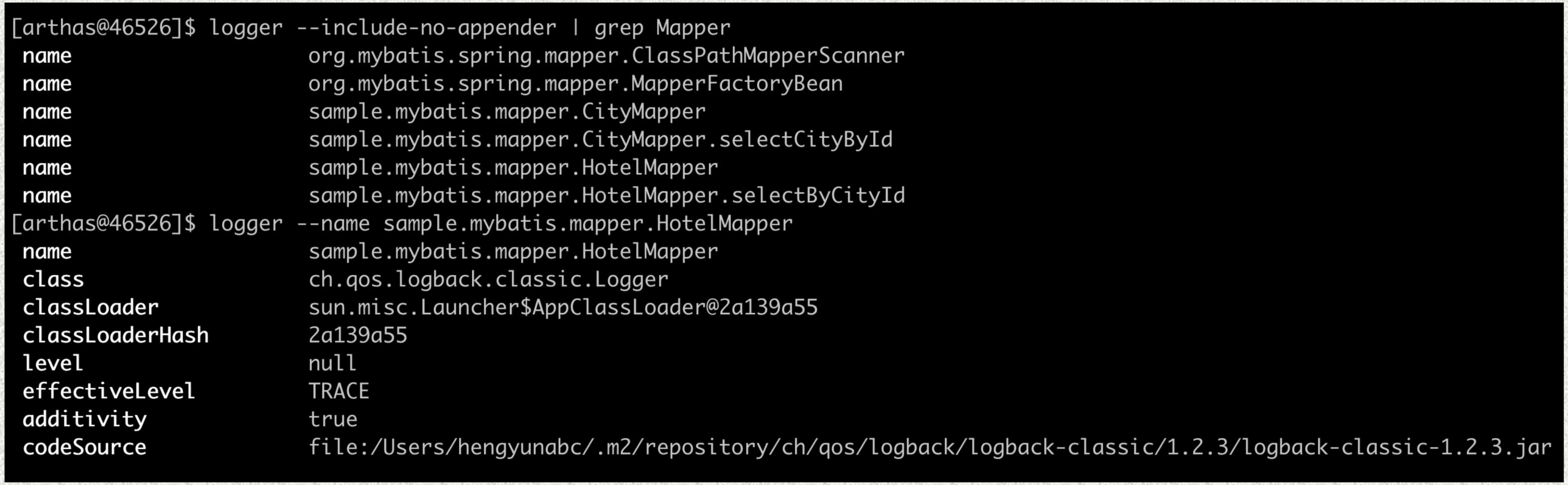


```
[arthas@46526]$ logger
 name                   ROOT
 class                  ch.qos.logback.classic.Logger
 classLoader            sun.misc.Launcher$AppClassLoader@2a139a55
 classLoaderHash        2a139a55
 level                  WARN
 effectiveLevel         WARN
 additivity             true
 codeSource             file:/Users/hengyunabc/.m2/repository/ch/qos/logback/logback-classic/1.2.3/logback-classic-1.2.3.jar
 appenders              name            CONSOLE
                        class           ch.qos.logback.core.ConsoleAppender
                        classLoader     sun.misc.Launcher$AppClassLoader@2a139a55
                        classLoaderHash 2a139a55
                        target          System.out
```

- 从appenders可以看到logger的打印到哪里

动态查看mybatis具体执行的sql

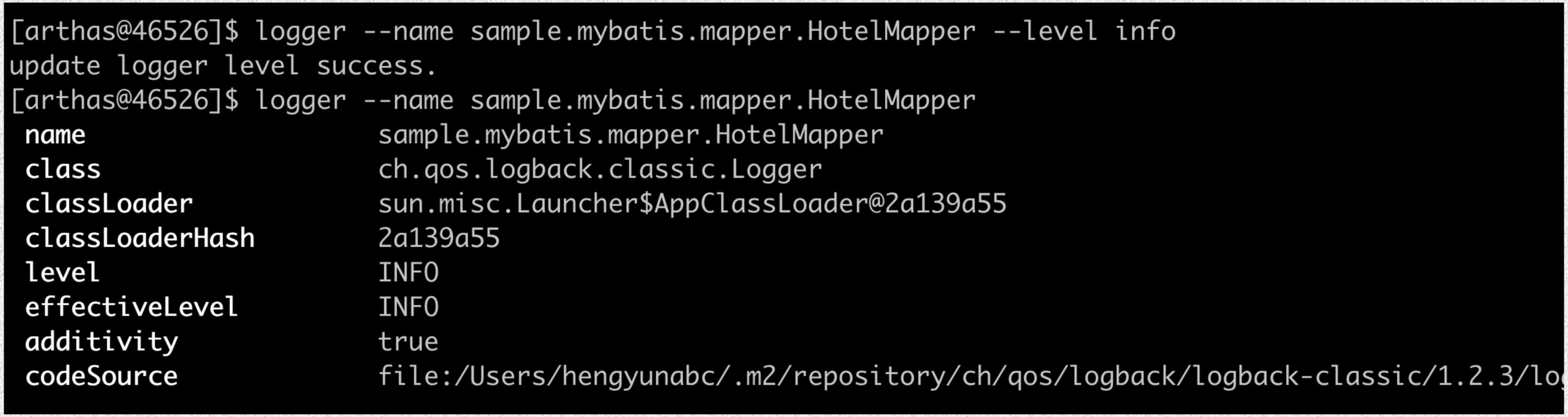
- 查找出mybatis mapper的logger



```
[arthas@46526]$ logger --include-no-appender | grep Mapper
 name                       org.mybatis.spring.mapper.ClassPathMapperScanner
 name                       org.mybatis.spring.mapper.MapperFactoryBean
 name                       sample.mybatis.mapper.CityMapper
 name                       sample.mybatis.mapper.CityMapper.selectCityById
 name                       sample.mybatis.mapper.HotelMapper
 name                       sample.mybatis.mapper.HotelMapper.selectByCityId
[arthas@46526]$ logger --name sample.mybatis.mapper.HotelMapper
 name                       sample.mybatis.mapper.HotelMapper
 class                      ch.qos.logback.classic.Logger
 classLoader                sun.misc.Launcher$AppClassLoader@2a139a55
 classLoaderHash            2a139a55
 level                      null
 effectiveLevel             TRACE
 additivity                 true
 codeSource                 file:/Users/hengyunabc/.m2/repository/ch/qos/logback/logback-classic/1.2.3/logback-classic-1.2.3.jar
```

动态查看mybatis具体执行的sql

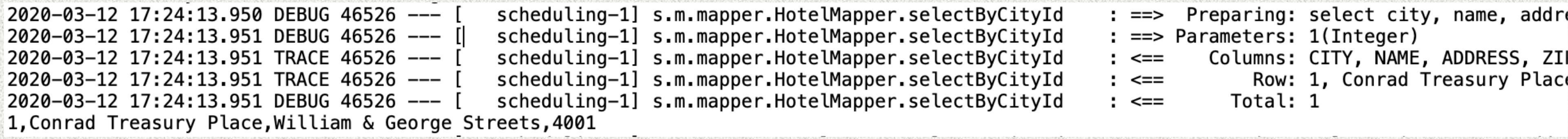
- 更新mybatis mapper的logger level



```
[arthas@46526]$ logger --name sample.mybatis.mapper.HotelMapper --level info
update logger level success.
[arthas@46526]$ logger --name sample.mybatis.mapper.HotelMapper
 name                   sample.mybatis.mapper.HotelMapper
 class                  ch.qos.logback.classic.Logger
 classLoader            sun.misc.Launcher$AppClassLoader@2a139a55
 classLoaderHash        2a139a55
 level                  INFO
 effectiveLevel         INFO
 additivity             true
 codeSource             file:/Users/hengyunabc/.m2/repository/ch/qos/logback/logback-classic/1.2.3/lo
```

mybatis具体在哪里打印出日志的？

- 已经知道修改mybatis的 logger level可以打印出sql，但具体是在mybatis的哪个类里？

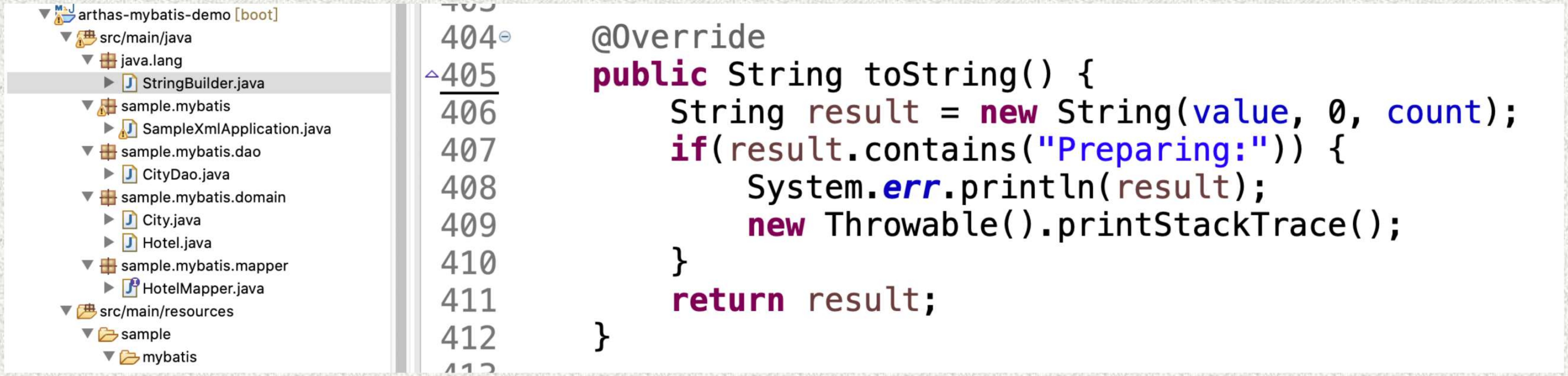


```
2020-03-12 17:24:13.950 DEBUG 46526 --- [   scheduling-1] s.m.mapper.HotelMapper.selectByCityId     : ==>  Preparing: select city, name, addr
2020-03-12 17:24:13.951 DEBUG 46526 --- [|  scheduling-1] s.m.mapper.HotelMapper.selectByCityId     : ==> Parameters: 1(Integer)
2020-03-12 17:24:13.951 TRACE 46526 --- [   scheduling-1] s.m.mapper.HotelMapper.selectByCityId     : <==    Columns: CITY, NAME, ADDRESS, ZI
2020-03-12 17:24:13.951 TRACE 46526 --- [   scheduling-1] s.m.mapper.HotelMapper.selectByCityId     : <==        Row: 1, Conrad Treasury Plac
2020-03-12 17:24:13.951 DEBUG 46526 --- [   scheduling-1] s.m.mapper.HotelMapper.selectByCityId     : <==      Total: 1
1,Conrad Treasury Place,William & George Streets,4001
```

- Logger最终打印时，会用StringBuilder拼接出结果
- redefine动态更新StringBuilder类的代码
- 在StringBuilder#toString() 函数里打印出调用栈

mybatis具体在哪里打印出日志的？

- 复制一份JDK里的StringBuilder代码到工程里



```java
    @Override
    public String toString() {
        String result = new String(value, 0, count);
        if(result.contains("Preparing:")) {
            System.err.println(result);
            new Throwable().printStackTrace();
        }
        return result;
    }
```

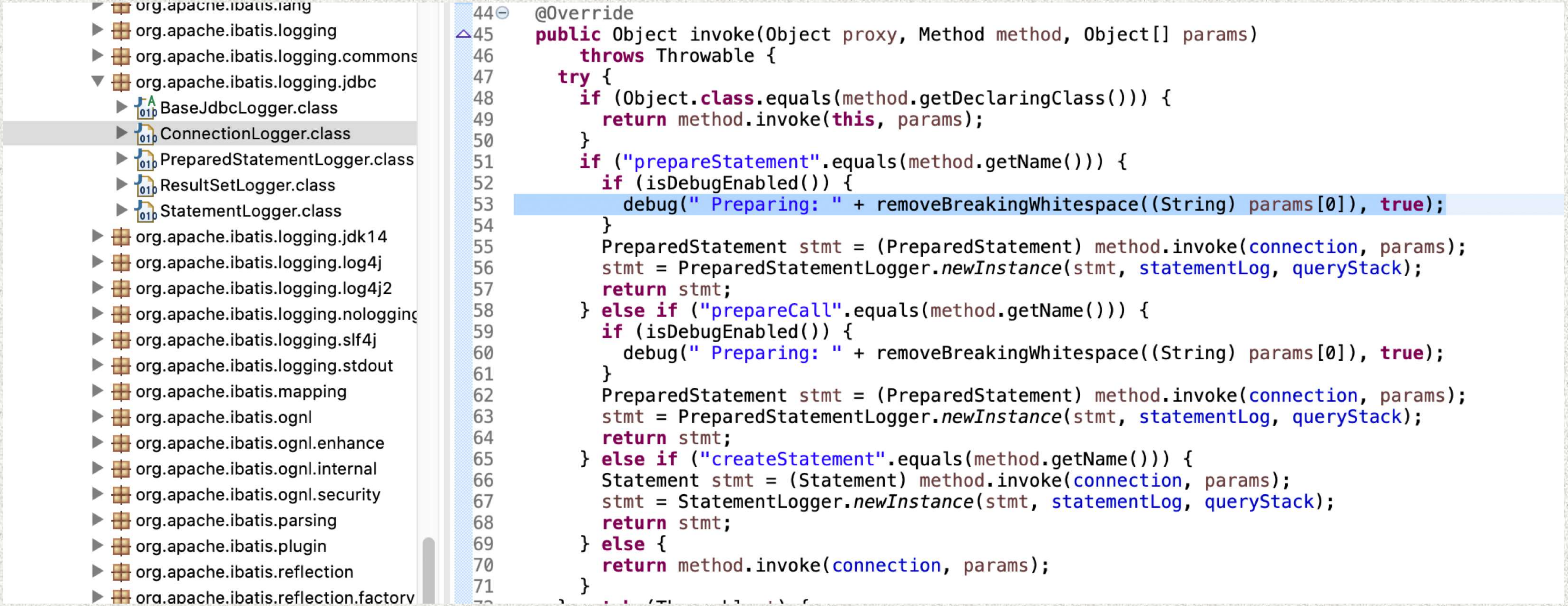
- redefine动态更新StringBuilder类的代码

```
[arthas@46526]$ redefine /Users/hengyunabc/code/java/arthas-mybatis-demo/target/classes/java/lang/StringBuilder.class
redefine success, size: 1
```

```
java.lang.Throwable
        at java.lang.StringBuilder.toString(StringBuilder.java:412)
        at ch.qos.logback.core.pattern.PatternLayoutBase.writeLoopOnConverters(PatternLayoutBase.java:118)
        at ch.qos.logback.classic.PatternLayout.doLayout(PatternLayout.java:141)
        at ch.qos.logback.classic.PatternLayout.doLayout(PatternLayout.java:39)
        at ch.qos.logback.core.encoder.LayoutWrappingEncoder.encode(LayoutWrappingEncoder.java:115)
        at ch.qos.logback.core.OutputStreamAppender.subAppend(OutputStreamAppender.java:230)
        at ch.qos.logback.core.OutputStreamAppender.append(OutputStreamAppender.java:102)
        at ch.qos.logback.core.UnsynchronizedAppenderBase.doAppend(UnsynchronizedAppenderBase.java:84)
        at ch.qos.logback.core.spi.AppenderAttachableImpl.appendLoopOnAppenders(AppenderAttachableImpl.java:51)
        at ch.qos.logback.classic.Logger.appendLoopOnAppenders(Logger.java:270)
        at ch.qos.logback.classic.Logger.callAppenders(Logger.java:257)
        at ch.qos.logback.classic.Logger.buildLoggingEventAndAppend(Logger.java:421)
        at ch.qos.logback.classic.Logger.filterAndLog_0_Or3Plus(Logger.java:383)
        at ch.qos.logback.classic.Logger.log(Logger.java:765)
        at org.apache.ibatis.logging.slf4j.Slf4jLocationAwareLoggerImpl.debug(Slf4jLocationAwareLoggerImpl.java:61)
        at org.apache.ibatis.logging.slf4j.Slf4jImpl.debug(Slf4jImpl.java:72)
        at org.apache.ibatis.logging.jdbc.BaseJdbcLogger.debug(BaseJdbcLogger.java:143)
        at org.apache.ibatis.logging.jdbc.ConnectionLogger.invoke(ConnectionLogger.java:53)
```

mybatis具体在哪里打印出日志的？

- org.apache.ibatis.logging.jdbc.ConnectionLogger

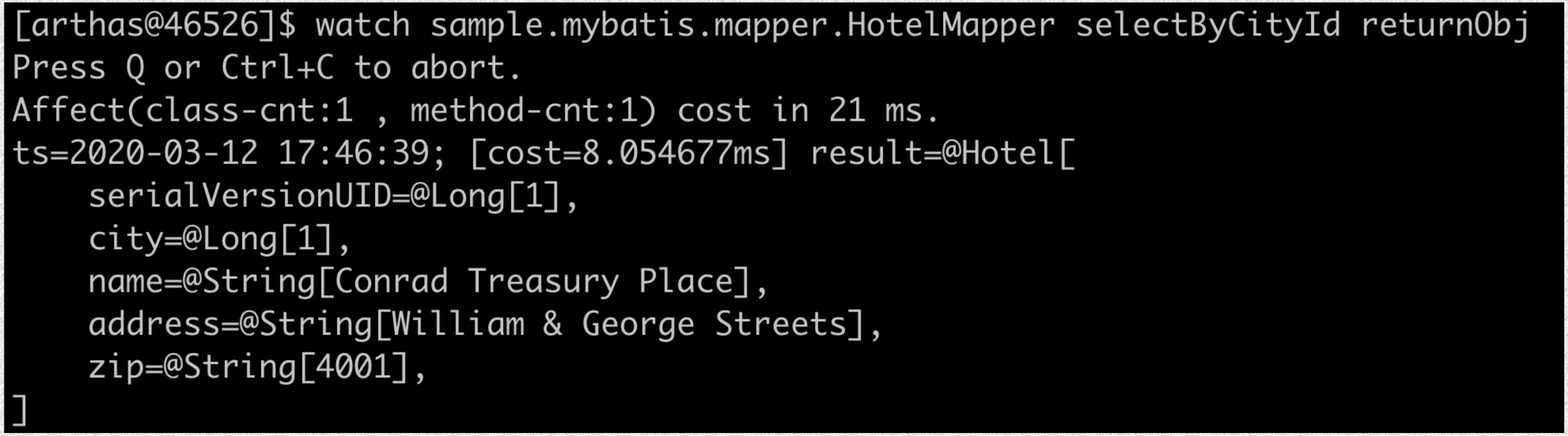


```
▶ org.apache.ibatis.lang
▶ org.apache.ibatis.logging
▶ org.apache.ibatis.logging.commons
▼ org.apache.ibatis.logging.jdbc
    ▶ BaseJdbcLogger.class
    ▶ ConnectionLogger.class
    ▶ PreparedStatementLogger.class
    ▶ ResultSetLogger.class
    ▶ StatementLogger.class
▶ org.apache.ibatis.logging.jdk14
▶ org.apache.ibatis.logging.log4j
▶ org.apache.ibatis.logging.log4j2
▶ org.apache.ibatis.logging.nologging
▶ org.apache.ibatis.logging.slf4j
▶ org.apache.ibatis.logging.stdout
▶ org.apache.ibatis.mapping
▶ org.apache.ibatis.ognl
▶ org.apache.ibatis.ognl.enhance
▶ org.apache.ibatis.ognl.internal
▶ org.apache.ibatis.ognl.security
▶ org.apache.ibatis.parsing
▶ org.apache.ibatis.plugin
▶ org.apache.ibatis.reflection
▶ org.apache.ibatis.reflection.factory
```

```java
@Override
public Object invoke(Object proxy, Method method, Object[] params)
    throws Throwable {
  try {
    if (Object.class.equals(method.getDeclaringClass())) {
      return method.invoke(this, params);
    }
    if ("prepareStatement".equals(method.getName())) {
      if (isDebugEnabled()) {
        debug(" Preparing: " + removeBreakingWhitespace((String) params[0]), true);
      }
      PreparedStatement stmt = (PreparedStatement) method.invoke(connection, params);
      stmt = PreparedStatementLogger.newInstance(stmt, statementLog, queryStack);
      return stmt;
    } else if ("prepareCall".equals(method.getName())) {
      if (isDebugEnabled()) {
        debug(" Preparing: " + removeBreakingWhitespace((String) params[0]), true);
      }
      PreparedStatement stmt = (PreparedStatement) method.invoke(connection, params);
      stmt = PreparedStatementLogger.newInstance(stmt, statementLog, queryStack);
      return stmt;
    } else if ("createStatement".equals(method.getName())) {
      Statement stmt = (Statement) method.invoke(connection, params);
      stmt = StatementLogger.newInstance(stmt, statementLog, queryStack);
      return stmt;
    } else {
      return method.invoke(connection, params);
    }
```

Mybatis数据库查询消耗了多少时间？

- 使用watch命令查看mapper执行时间和返回结果



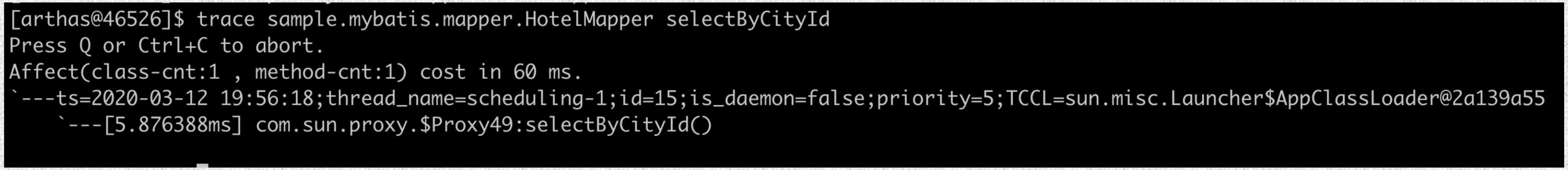
```
[arthas@46526]$ watch sample.mybatis.mapper.HotelMapper selectByCityId returnObj
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 21 ms.
ts=2020-03-12 17:46:39; [cost=8.054677ms] result=@Hotel[
    serialVersionUID=@Long[1],
    city=@Long[1],
    name=@String[Conrad Treasury Place],
    address=@String[William & George Streets],
    zip=@String[4001],
]
```

Mybatis数据库查询消耗了多少时间？

- 返回结果可以组合 params, target, returnObj, method, clazz等
- 条件表达式可以灵活判断，比如超过3ms
- watch –h
- 用tab可以灵活补全

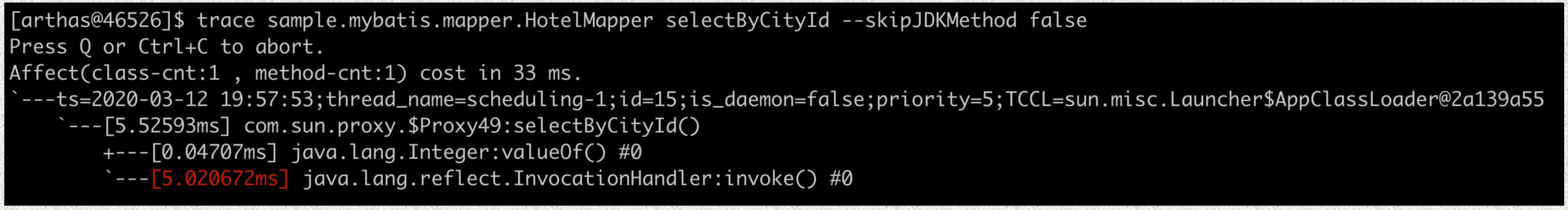
Mybatis的mapper是怎样实现的？

- mapper只是一个接口，没有具体的实现类
- 使用trace跟踪一个函数里的子调用



```
[arthas@46526]$ trace sample.mybatis.mapper.HotelMapper selectByCityId
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 60 ms.
`---ts=2020-03-12 19:56:18;thread_name=scheduling-1;id=15;is_daemon=false;priority=5;TCCL=sun.misc.Launcher$AppClassLoader@2a139a55
    `---[5.876388ms] com.sun.proxy.$Proxy49:selectByCityId()
```

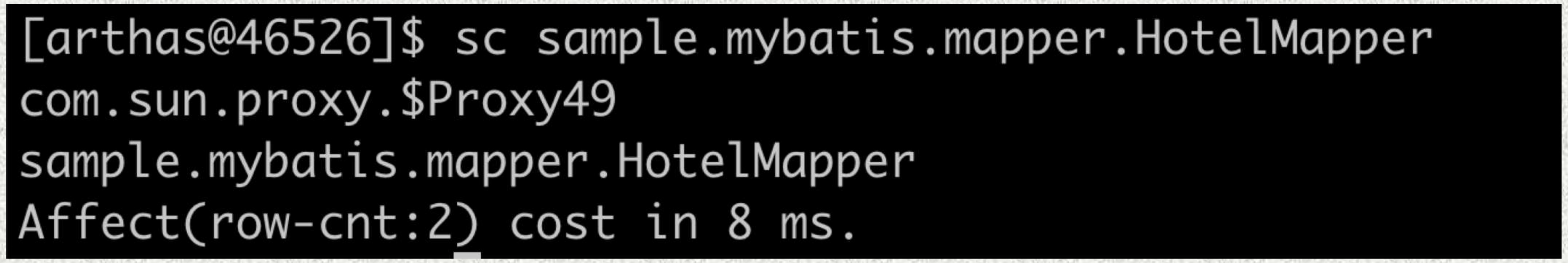
- 默认情况下会跳过jdk的函数，加上--skipJDKMethod false



```
[arthas@46526]$ trace sample.mybatis.mapper.HotelMapper selectByCityId --skipJDKMethod false
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 33 ms.
`---ts=2020-03-12 19:57:53;thread_name=scheduling-1;id=15;is_daemon=false;priority=5;TCCL=sun.misc.Launcher$AppClassLoader@2a139a55
    `---[5.52593ms] com.sun.proxy.$Proxy49:selectByCityId()
        +---[0.04707ms] java.lang.Integer:valueOf() #0
        `---[5.020672ms] java.lang.reflect.InvocationHandler:invoke() #0
```

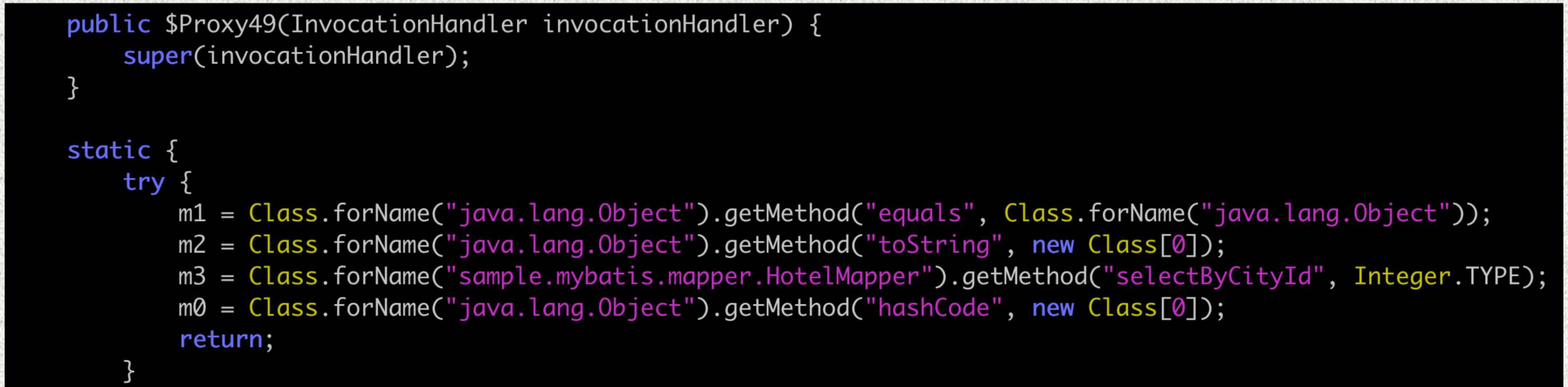
Mybatis的mapper是怎样实现的？

- mapper只是一个接口，没有具体的实现类
- 使用sc命令查找动态代理的实现



```
[arthas@46526]$ sc sample.mybatis.mapper.HotelMapper
com.sun.proxy.$Proxy49
sample.mybatis.mapper.HotelMapper
Affect(row-cnt:2) cost in 8 ms.
```

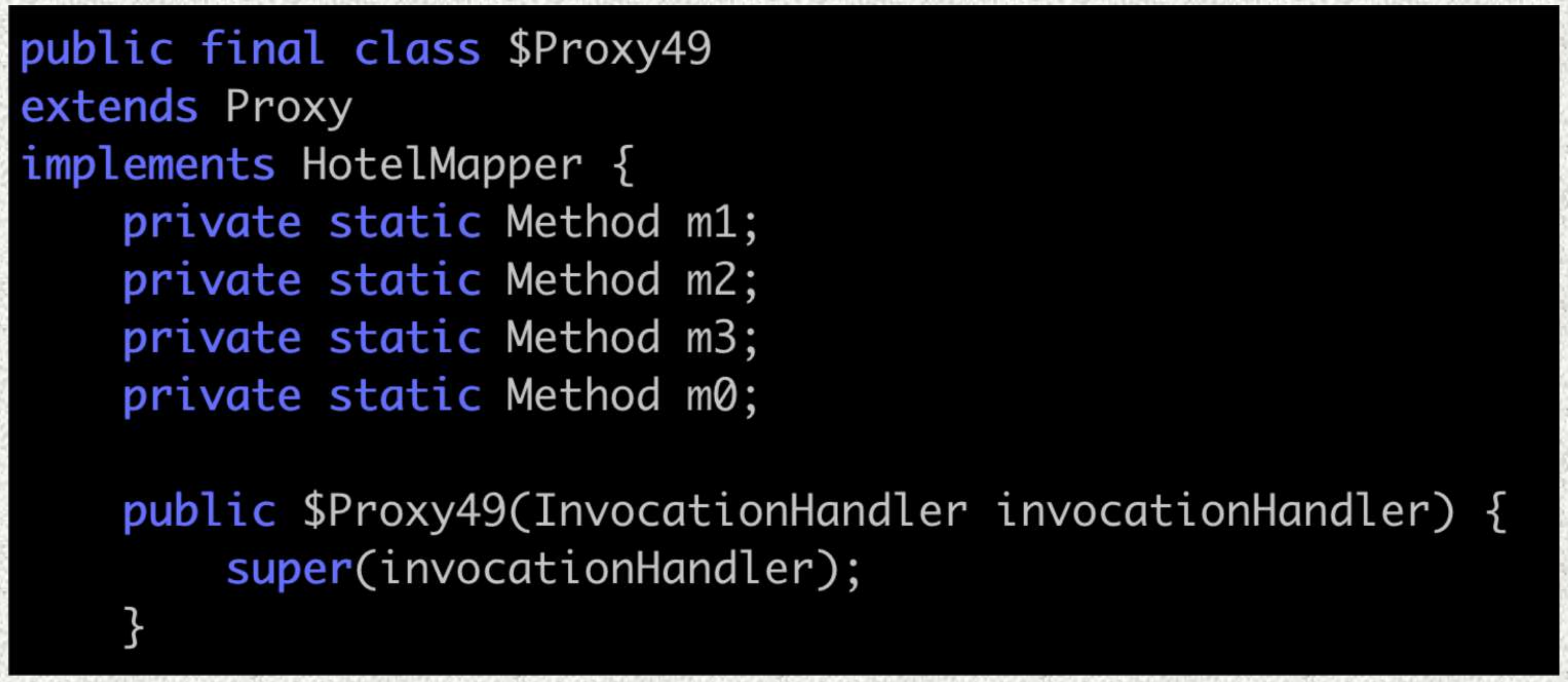
- 用jad反编译代码 jad com.sun.proxy.$Proxy49



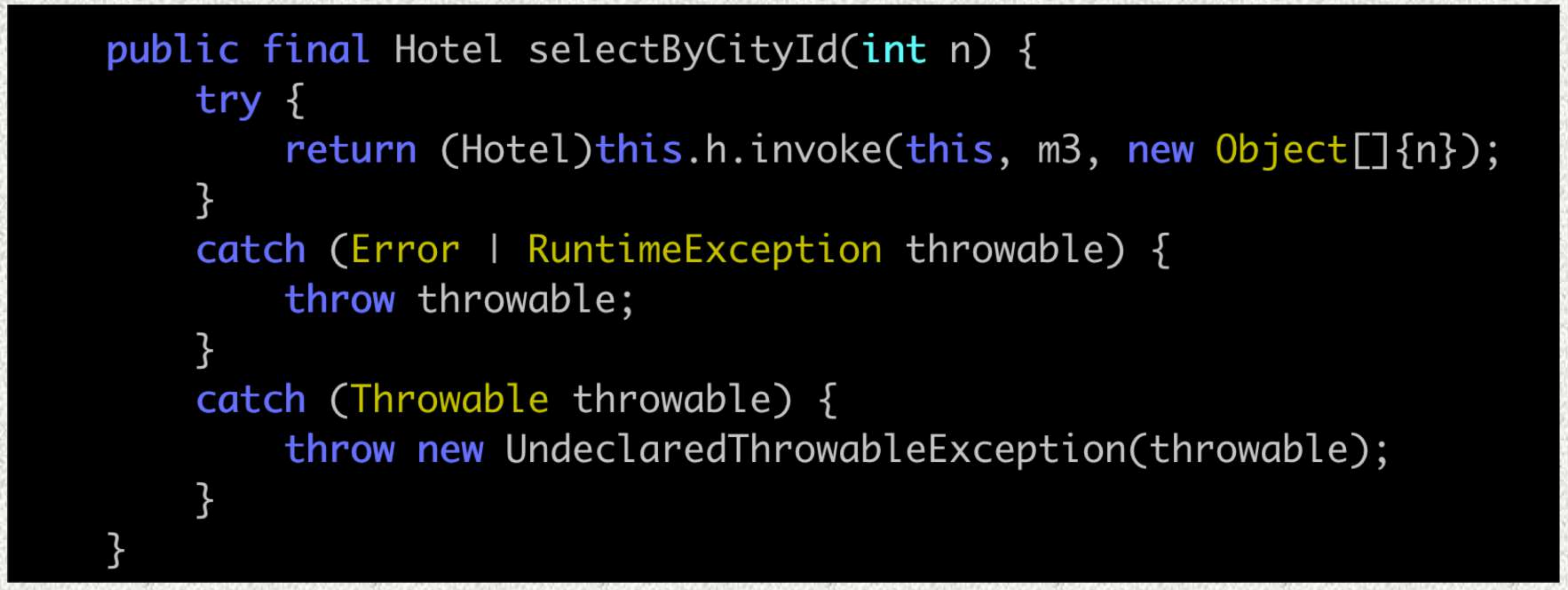
```
    public $Proxy49(InvocationHandler invocationHandler) {
        super(invocationHandler);
    }

    static {
        try {
            m1 = Class.forName("java.lang.Object").getMethod("equals", Class.forName("java.lang.Object"));
            m2 = Class.forName("java.lang.Object").getMethod("toString", new Class[0]);
            m3 = Class.forName("sample.mybatis.mapper.HotelMapper").getMethod("selectByCityId", Integer.TYPE);
            m0 = Class.forName("java.lang.Object").getMethod("hashCode", new Class[0]);
            return;
        }
```

Mybatis的mapper是怎样实现的？

- com.sun.proxy.$Proxy49 内部用InvocationHandler反射调用



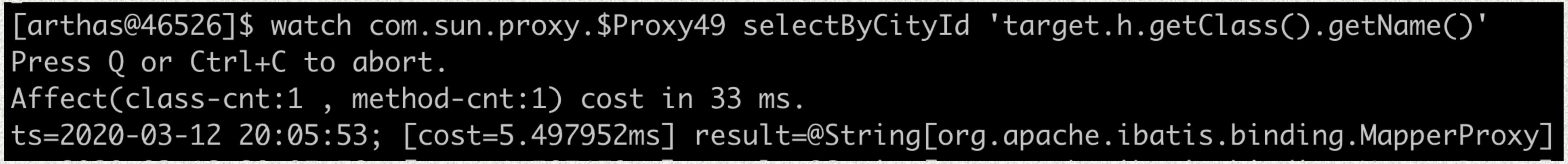
```java
public final class $Proxy49
extends Proxy
implements HotelMapper {
    private static Method m1;
    private static Method m2;
    private static Method m3;
    private static Method m0;

    public $Proxy49(InvocationHandler invocationHandler) {
        super(invocationHandler);
    }
```



```java
    public final Hotel selectByCityId(int n) {
        try {
            return (Hotel)this.h.invoke(this, m3, new Object[]{n});
        }
        catch (Error | RuntimeException throwable) {
            throw throwable;
        }
        catch (Throwable throwable) {
            throw new UndeclaredThrowableException(throwable);
        }
    }
```

- 用watch打印出h的具体类名

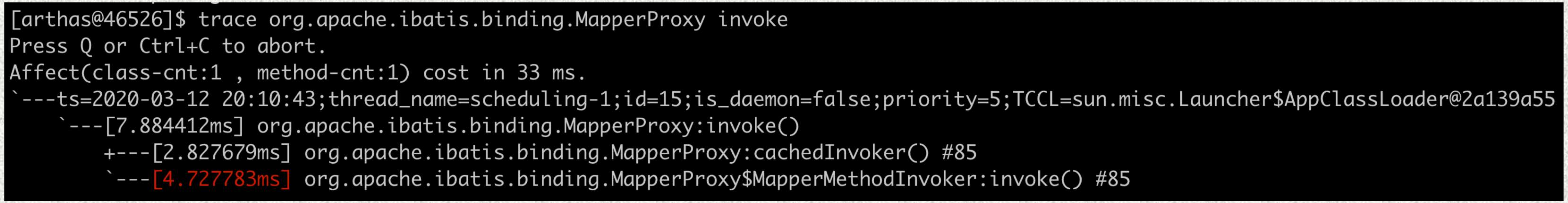
  org.apache.ibatis.binding.MapperProxy



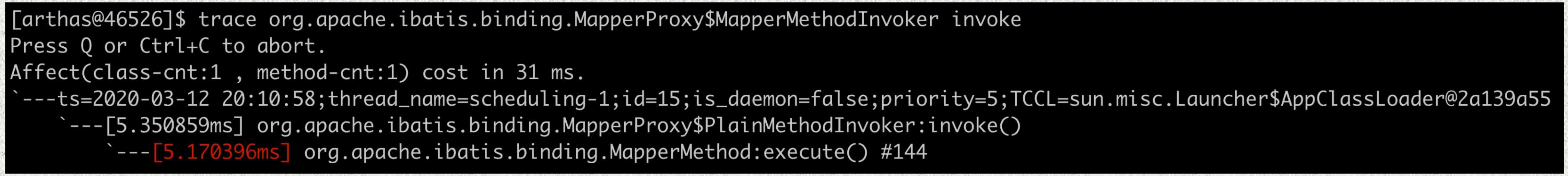
```
[arthas@46526]$ watch com.sun.proxy.$Proxy49 selectByCityId 'target.h.getClass().getName()'
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 33 ms.
ts=2020-03-12 20:05:53; [cost=5.497952ms] result=@String[org.apache.ibatis.binding.MapperProxy]
```

Mybatis的mapper是怎样实现的？

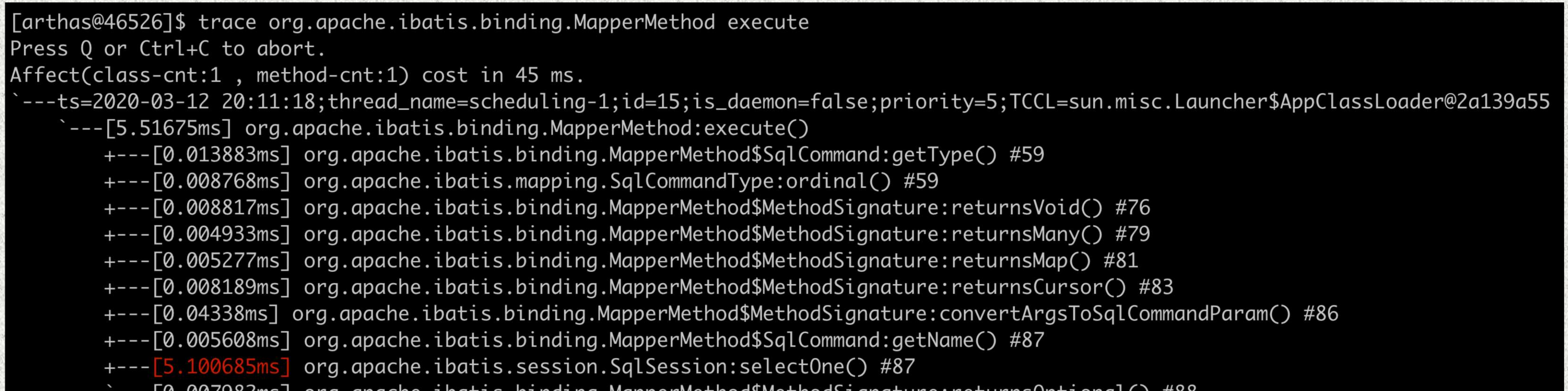
- 用trace层层深入



```
[arthas@46526]$ trace org.apache.ibatis.binding.MapperProxy invoke
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 33 ms.
`---ts=2020-03-12 20:10:43;thread_name=scheduling-1;id=15;is_daemon=false;priority=5;TCCL=sun.misc.Launcher$AppClassLoader@2a139a55
    `---[7.884412ms] org.apache.ibatis.binding.MapperProxy:invoke()
        +---[2.827679ms] org.apache.ibatis.binding.MapperProxy:cachedInvoker() #85
        `---[4.727783ms] org.apache.ibatis.binding.MapperProxy$MapperMethodInvoker:invoke() #85
```



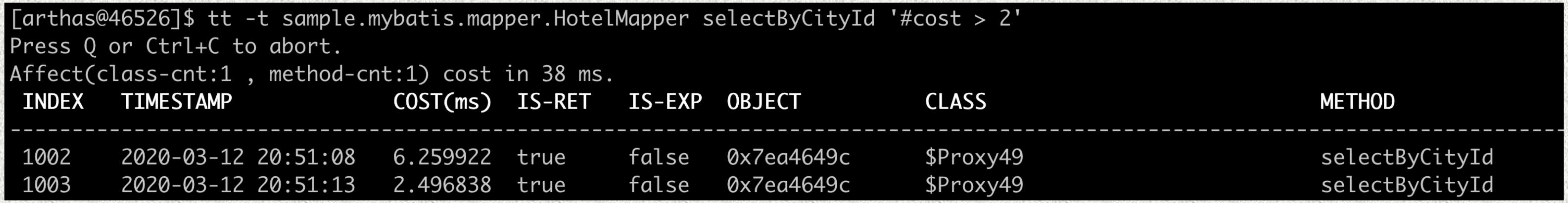
```
[arthas@46526]$ trace org.apache.ibatis.binding.MapperProxy$MapperMethodInvoker invoke
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 31 ms.
`---ts=2020-03-12 20:10:58;thread_name=scheduling-1;id=15;is_daemon=false;priority=5;TCCL=sun.misc.Launcher$AppClassLoader@2a139a55
    `---[5.350859ms] org.apache.ibatis.binding.MapperProxy$PlainMethodInvoker:invoke()
        `---[5.170396ms] org.apache.ibatis.binding.MapperMethod:execute() #144
```



```
[arthas@46526]$ trace org.apache.ibatis.binding.MapperMethod execute
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 45 ms.
`---ts=2020-03-12 20:11:18;thread_name=scheduling-1;id=15;is_daemon=false;priority=5;TCCL=sun.misc.Launcher$AppClassLoader@2a139a55
    `---[5.51675ms] org.apache.ibatis.binding.MapperMethod:execute()
        +---[0.013883ms] org.apache.ibatis.binding.MapperMethod$SqlCommand:getType() #59
        +---[0.008768ms] org.apache.ibatis.mapping.SqlCommandType:ordinal() #59
        +---[0.008817ms] org.apache.ibatis.binding.MapperMethod$MethodSignature:returnsVoid() #76
        +---[0.004933ms] org.apache.ibatis.binding.MapperMethod$MethodSignature:returnsMany() #79
        +---[0.005277ms] org.apache.ibatis.binding.MapperMethod$MethodSignature:returnsMap() #81
        +---[0.008189ms] org.apache.ibatis.binding.MapperMethod$MethodSignature:returnsCursor() #83
        +---[0.04338ms] org.apache.ibatis.binding.MapperMethod$MethodSignature:convertArgsToSqlCommandParam() #86
        +---[0.005608ms] org.apache.ibatis.binding.MapperMethod$SqlCommand:getName() #87
        +---[5.100685ms] org.apache.ibatis.session.SqlSession:selectOne() #87
        `---[0.007983ms] org.apache.ibatis.binding.MapperMethod$MethodSignature:returnsOptional() #88
```

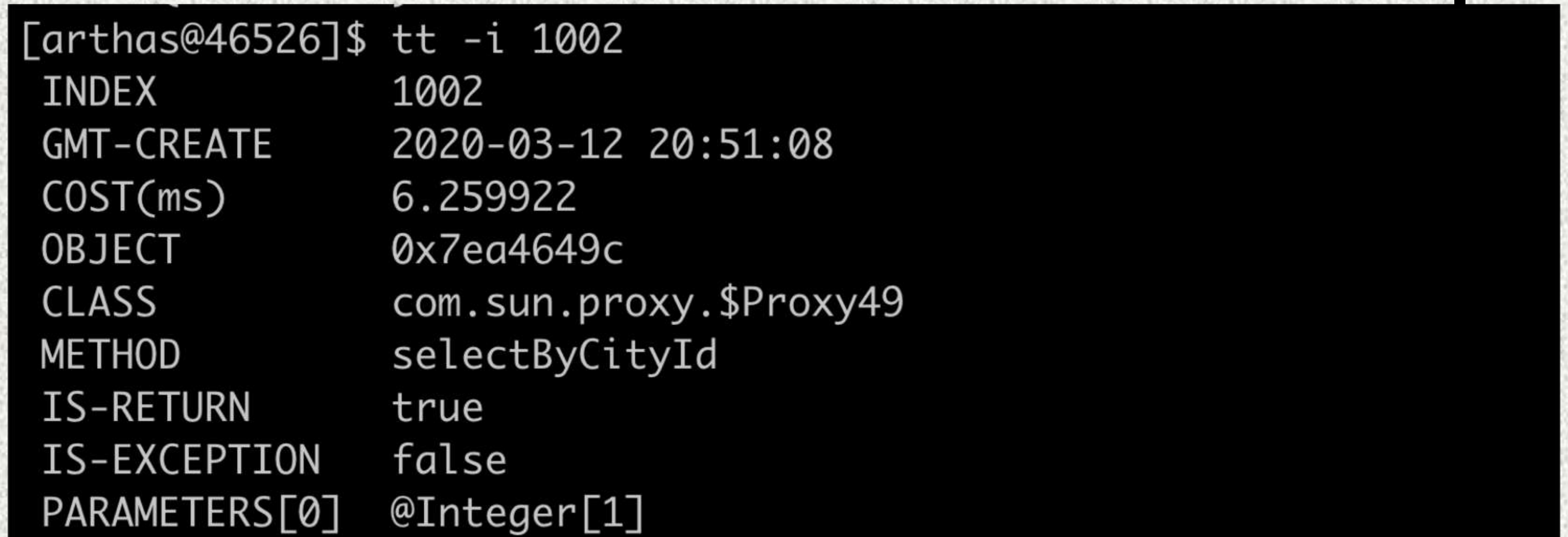
能重放调用不？

- demo里是一个定时任务调用查询，但如果抓到一个慢调用，可以重放来排查不？
- 用tt(timetunnel)记录慢调用



```
[arthas@46526]$ tt -t sample.mybatis.mapper.HotelMapper selectByCityId '#cost > 2'
Press Q or Ctrl+C to abort.
Affect(class-cnt:1 , method-cnt:1) cost in 38 ms.
 INDEX   TIMESTAMP             COST(ms)  IS-RET   IS-EXP  OBJECT          CLASS                           METHOD
------------------------------------------------------------------------------------------------------------------------------
 1002    2020-03-12 20:51:08   6.259922  true     false   0x7ea4649c      $Proxy49                        selectByCityId
 1003    2020-03-12 20:51:13   2.496838  true     false   0x7ea4649c      $Proxy49                        selectByCityId
```

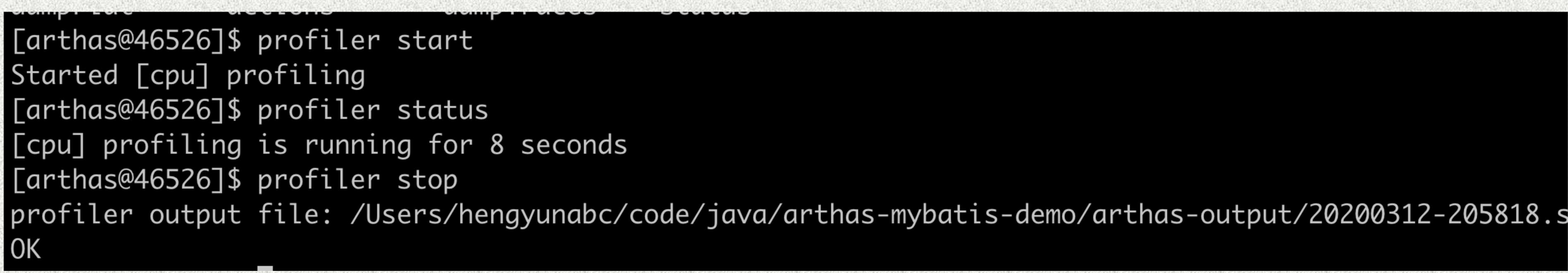
- 用 –i 参数查看调用详情， -p 参数重放



```
[arthas@46526]$ tt -i 1002
 INDEX          1002
 GMT-CREATE     2020-03-12 20:51:08
 COST(ms)       6.259922
 OBJECT         0x7ea4649c
 CLASS          com.sun.proxy.$Proxy49
 METHOD         selectByCityId
 IS-RETURN      true
 IS-EXCEPTION   false
 PARAMETERS[0]  @Integer[1]
```

生成热点火焰图

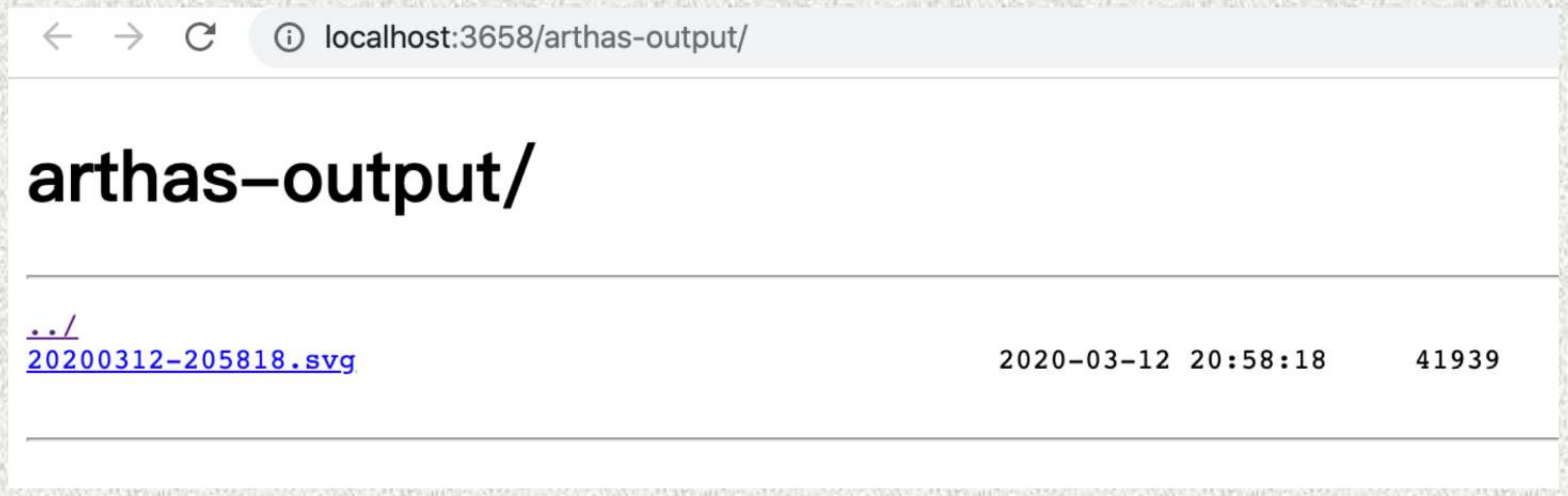
- 前面我们假设已知mybatis查询慢，如果未知代码怎样定位？
- 用profiler生成火焰图



```
[arthas@46526]$ profiler start
Started [cpu] profiling
[arthas@46526]$ profiler status
[cpu] profiling is running for 8 seconds
[arthas@46526]$ profiler stop
profiler output file: /Users/hengyunabc/code/java/arthas-mybatis-demo/arthas-output/20200312-205818.s
OK
```

在浏览器中查看火焰图

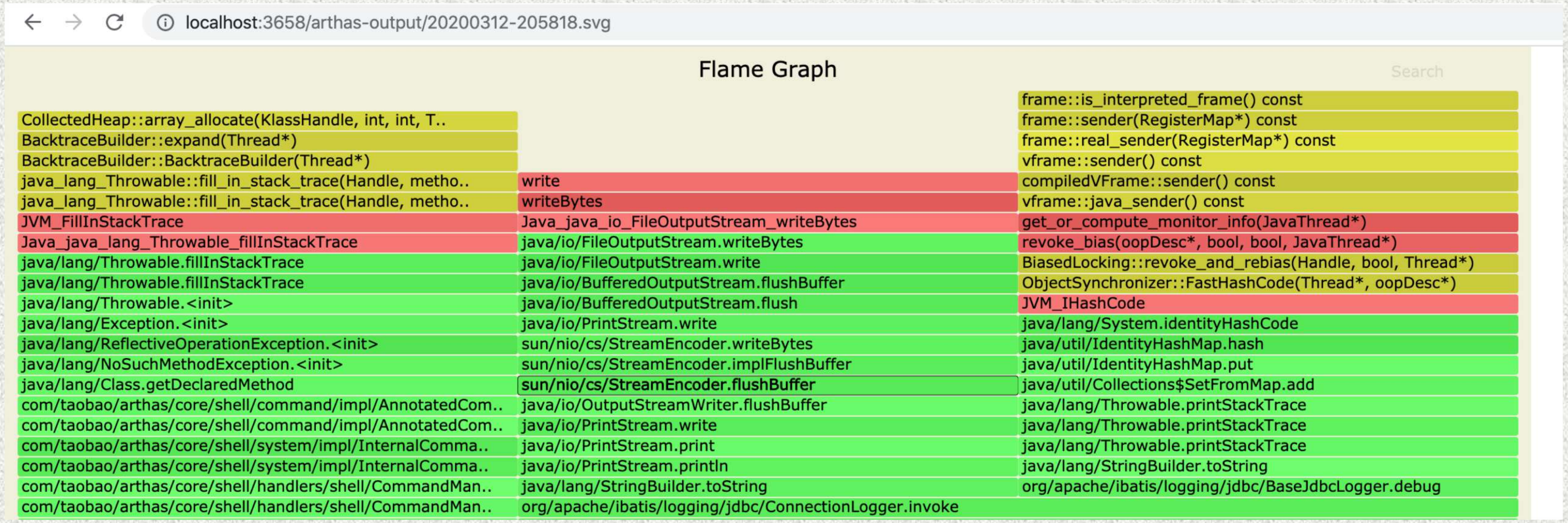
- http://localhost:3658/



localhost:3658/arthas-output/

arthas-output/

```
../
20200312-205818.svg                                    2020-03-12 20:58:18      41939
```



localhost:3658/arthas-output/20200312-205818.svg

Flame Graph

Search

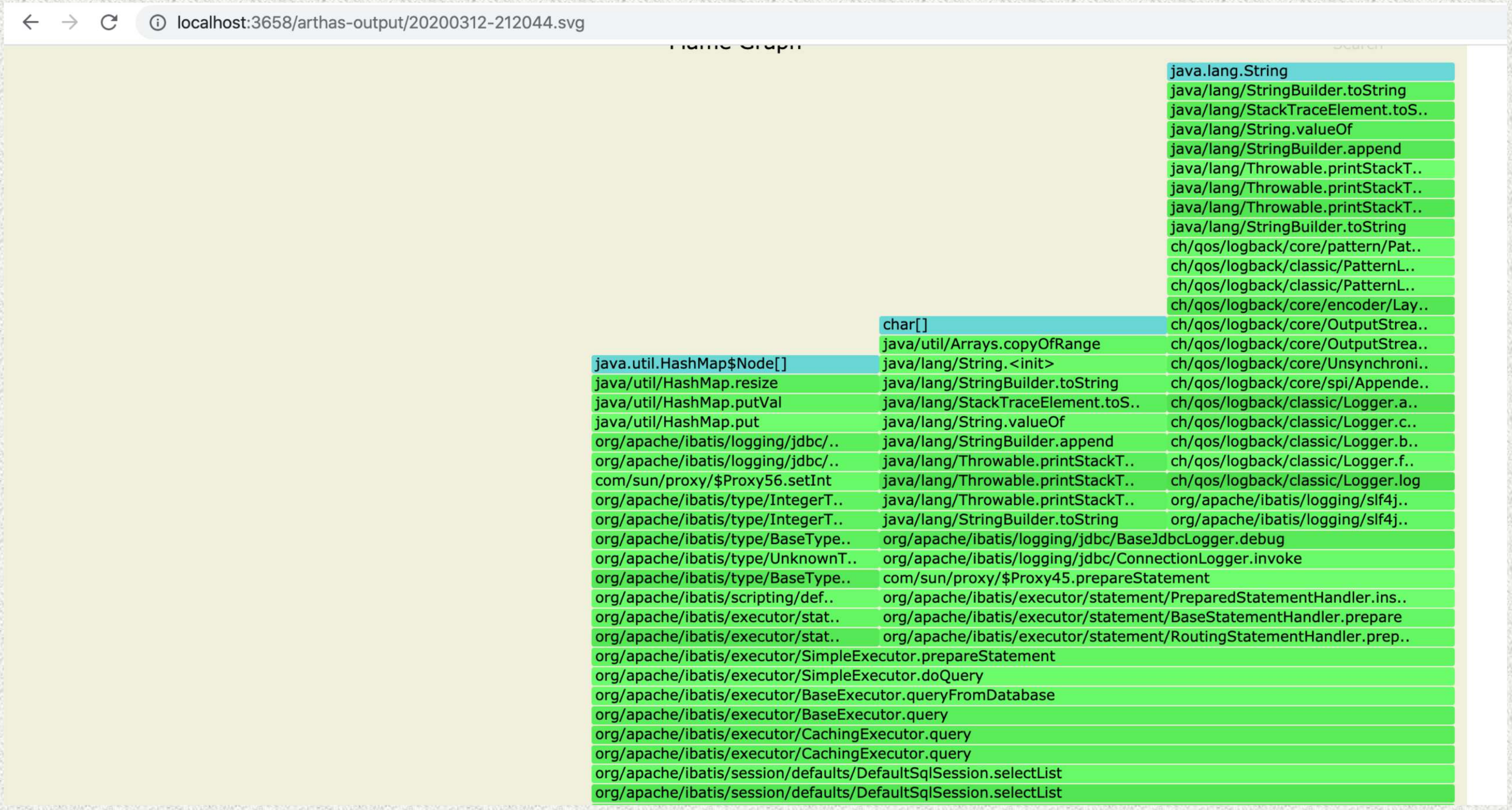
```
CollectedHeap::array_allocate(KlassHandle, int, int, T..
BacktraceBuilder::expand(Thread*)
BacktraceBuilder::BacktraceBuilder(Thread*)
java_lang_Throwable::fill_in_stack_trace(Handle, metho..
java_lang_Throwable::fill_in_stack_trace(Handle, metho..
JVM_FillInStackTrace
Java_java_lang_Throwable_fillInStackTrace
java/lang/Throwable.fillInStackTrace
java/lang/Throwable.fillInStackTrace
java/lang/Throwable.<init>
java/lang/Exception.<init>
java/lang/ReflectiveOperationException.<init>
java/lang/NoSuchMethodException.<init>
java/lang/Class.getDeclaredMethod
com/taobao/arthas/core/shell/command/impl/AnnotatedCom..
com/taobao/arthas/core/shell/command/impl/AnnotatedCom..
com/taobao/arthas/core/shell/system/impl/InternalComma..
com/taobao/arthas/core/shell/system/impl/InternalComma..
com/taobao/arthas/core/shell/handlers/shell/CommandMan..
com/taobao/arthas/core/shell/handlers/shell/CommandMan..
```

```
write
writeBytes
Java_java_io_FileOutputStream_writeBytes
java/io/FileOutputStream.writeBytes
java/io/FileOutputStream.write
java/io/BufferedOutputStream.flushBuffer
java/io/BufferedOutputStream.flush
java/io/PrintStream.write
sun/nio/cs/StreamEncoder.writeBytes
sun/nio/cs/StreamEncoder.implFlushBuffer
sun/nio/cs/StreamEncoder.flushBuffer
java/io/OutputStreamWriter.flushBuffer
java/io/PrintStream.write
java/io/PrintStream.print
java/io/PrintStream.println
java/lang/StringBuilder.toString
org/apache/ibatis/logging/jdbc/ConnectionLogger.invoke
```

```
frame::is_interpreted_frame() const
frame::sender(RegisterMap*) const
frame::real_sender(RegisterMap*) const
vframe::sender() const
compiledVFrame::sender() const
vframe::java_sender() const
get_or_compute_monitor_info(JavaThread*)
revoke_bias(oopDesc*, bool, bool, JavaThread*)
BiasedLocking::revoke_and_rebias(Handle, bool, Thread*)
ObjectSynchronizer::FastHashCode(Thread*, oopDesc*)
JVM_IHashCode
java/lang/System.identityHashCode
java/util/IdentityHashMap.hash
java/util/IdentityHashMap.put
java/util/Collections$SetFromMap.add
java/lang/Throwable.printStackTrace
java/lang/Throwable.printStackTrace
java/lang/Throwable.printStackTrace
java/lang/StringBuilder.toString
org/apache/ibatis/logging/jdbc/BaseJdbcLogger.debug
```

生成内存分配火焰图

- profiler start --event alloc



localhost:3658/arthas-output/20200312-212044.svg

```
                                                                        java.lang.String
                                                                        java/lang/StringBuilder.toString
                                                                        java/lang/StackTraceElement.toS..
                                                                        java/lang/String.valueOf
                                                                        java/lang/StringBuilder.append
                                                                        java/lang/Throwable.printStackT..
                                                                        java/lang/Throwable.printStackT..
                                                                        java/lang/Throwable.printStackT..
                                                                        java/lang/StringBuilder.toString
                                                                        ch/qos/logback/core/pattern/Pat..
                                                                        ch/qos/logback/classic/PatternL..
                                                                        ch/qos/logback/classic/PatternL..
                                                                        ch/qos/logback/core/encoder/Lay..
                                    char[]                              ch/qos/logback/core/OutputStrea..
                                    java/util/Arrays.copyOfRange        ch/qos/logback/core/OutputStrea..
java.util.HashMap$Node[]            java/lang/String.<init>             ch/qos/logback/core/Unsynchroni..
java/util/HashMap.resize            java/lang/StringBuilder.toString    ch/qos/logback/core/spi/Appende..
java/util/HashMap.putVal            java/lang/StackTraceElement.toS..   ch/qos/logback/classic/Logger.a..
java/util/HashMap.put               java/lang/String.valueOf            ch/qos/logback/classic/Logger.c..
org/apache/ibatis/logging/jdbc/..   java/lang/StringBuilder.append      ch/qos/logback/classic/Logger.b..
org/apache/ibatis/logging/jdbc/..   java/lang/Throwable.printStackT..   ch/qos/logback/classic/Logger.f..
com/sun/proxy/$Proxy56.setInt       java/lang/Throwable.printStackT..   ch/qos/logback/classic/Logger.log
org/apache/ibatis/type/IntegerT..   java/lang/Throwable.printStackT..   org/apache/ibatis/logging/slf4j..
org/apache/ibatis/type/IntegerT..   java/lang/StringBuilder.toString    org/apache/ibatis/logging/slf4j..
org/apache/ibatis/type/BaseType..   org/apache/ibatis/logging/jdbc/BaseJdbcLogger.debug
org/apache/ibatis/type/UnknownT..   org/apache/ibatis/logging/jdbc/ConnectionLogger.invoke
org/apache/ibatis/type/BaseType..   com/sun/proxy/$Proxy45.prepareStatement
org/apache/ibatis/scripting/def..   org/apache/ibatis/executor/statement/PreparedStatementHandler.ins..
org/apache/ibatis/executor/stat..   org/apache/ibatis/executor/statement/BaseStatementHandler.prepare
org/apache/ibatis/executor/stat..   org/apache/ibatis/executor/statement/RoutingStatementHandler.prep..
org/apache/ibatis/executor/SimpleExecutor.prepareStatement
org/apache/ibatis/executor/SimpleExecutor.doQuery
org/apache/ibatis/executor/BaseExecutor.queryFromDatabase
org/apache/ibatis/executor/BaseExecutor.query
org/apache/ibatis/executor/CachingExecutor.query
org/apache/ibatis/executor/CachingExecutor.query
org/apache/ibatis/session/defaults/DefaultSqlSession.selectList
org/apache/ibatis/session/defaults/DefaultSqlSession.selectList
```

Tips

- Arthas并不止是一个诊断工具，它是各种利器的集合，灵活组合可以抽丝剥茧，深入应用内部。了解得越多，越容易解决问题。
- Gitee文档镜像: https://arthas.gitee.io
- 键盘Up/Down匹配历史
- Tab补全
- 快捷键: keymap
- Pipeline/grep
- 在线交互入门教程：https://alibaba.github.io/arthas/arthas-tutorials
- 用户案例: https://github.com/alibaba/arthas/issues?q=label%3Auser-case
- 当DUBBO遇上Arthas-排查问题的实践
- 当SpringBoot遇上Arthas-深入细节和排查问题的实践

Arthas钉钉/QQ交流群

Arthas开源交流群
2314人

扫一扫群二维码，立刻加入该群。

Alibaba Arthas开源交流群
扫一扫二维码，加入群聊。

Thank you !