- Actor的管理，与spring的结合
- 流控
- 系统架构
- 性能测试数据

不频繁申请释放的大对象直接放到老年代

```
Class#newInstance();
```

创建一个由 this 代表的类的对象。该类以类似 new A()的方式被实例化。该类会被初始化，如果没有过的话(诸如一些静态成员，静态初始化代码段)。此方法会传递所有由无参构造函数所产生的异常。该方法的调用会绕过本来会执行的编译器异常检查。java.lang.reflect.Constructor#newInstance通过将构造函数抛出的异常包装在java.lang.reflect.InvocationTargetException中来避免这个问题。

```
DriverManager#getConnection(String url, java.util.Properties info) throws SQLException
```

对给定的数据库URL建立连接。DriverManager会试着从注册过的JDBC drivers中选择合适的driver。如果某个属性既在url指定了，又在properties中设置了，那么优先级会有具体实现(什么的实现？ driver?)决定。该函数会抛出SQLException(数据库访问错误)，和SQLTimeoutException(连接超时，诸如发了报文一直没有响应)。

```
Thread#getContextClassLoader()
```

返回这个线程的上下文类加载器(ClassLoader)。ClassLoader是由该线程的创建者提供，用来在线程代码执行前装载类和资源。如果没有调用 `setContextClassLoader` 设置加载器，默认使用父线程的ClassLoader。原始线程的ClassLoader一般设置为装载应用本身的类装载器(这是哪个？)。

`driver.connect(url, info)`

未提交读 read uncommitted ，可以读到其他会话未提交的修改

提交读 read committed ，只能读到已经提交的的修改

可重复读reapeated read，某查询在事务开始到提交期间都是一致的，不会因为本次事务的修改而改变，innoDB默认级别。

串行读 serializable，每次读都要获取表级共享锁，读写互相阻塞，innoDB默认行锁。

```
Object#await(long timeout)
```

```
PooledDataSource#popConnection
```

mybatis连接池获取连接，首先判断是否存在空闲连接，如果有，就从空闲连接池取出，如果没有就获取当前最老的活跃连接，判断其是否过期(poolMaximumCheckoutTime)，如果过期了，就注销pooledConnect(大概是socket连接的一层壳，用来标志某个应用获取的连接，找这块代码的意思，应用获取的每个连接的有效时间都不怎长啊，尽管最底层tcp连接是一直建着的长连接)

充值押金请求同时重复提交问题。随便充用户有一项属性是押金，每个用户的押金存在两种状态：1未充值 2已充值。假设当前用户未充值，此时同时有两个充值请求到来，先获取用户进行中的充值单，如果不存在，则创建一笔充值单。此时出现问题生成了两笔充值单。原因是innoDB默认隔离级别为repeated read，所以两笔事务同时select时都没有找到进行中的订单，这导致两个事务都会进行insert操作，而数据库并没有通过唯一键等手段来保证这个约束(同一用户同一时间只能有一笔进行中的充值订单)。

解决方法：

1. 要保证本地数据合法，最安全的是通过db本身提供的约束。给充值订单表添加字段 `active:Bool`，对active和userId建立联合唯一索引，进行中的订单active插入1，失效之后设置为null。
2. 由数据库保证本地数据合法是不够的，比如创建充值订单是会调用远端服务创建其他记录，在本地事务插入失败回滚的时候，远端的事务是没法回退的，当然通过脚本扫描数据库检查可以保证最终一致性。如果能保证相同user的押金充值请求是串行的，就可以简单高效的解决并发充值的问题，因此在引入akka库之后，创建router，使用ConsistentHashing按userId做分发，将充值请求分发到10000个worker actor上，在保证并发的同时确保同一用户的充值请求是串行处理的。性能分析记录在《xxxx》。

系统内所有类似select existence then do business的地方都有这个问题，通过在模块入口使用akka actor+定制路由的实现可以规避此类问题(当然也可以用BlockingQueue)，而对数据库设置严格的约束，可以保证上层业务系统无法对数据造成污染。再不济最后还有脚本检测数据一致性。

actor注意点,transcational不能直接用spring的，因为spring的事务基于threadlocal variable

- return code 与 exception 的选择

  https://stackoverflow.com/questions/99683/which-and-why-do-you-prefer-exceptions-or-return-codes

  exception优势有三:

  1. 代码结构更清晰，错误处理不影响正常流程的代码
  2. 异常不容易被函数调用方忽略，相比之下返回码更脆弱
  3. 异常有类型化的，同一个函数可以抛出不同异常，相比之下返回值是类型确定而且单一的。

sqlsession, executor

binding 绑定 interface里面的函数-----MapperMethod & SqlCommand

```
MappedStatement 存储sql语句
```

```
[root@t1kvm01 lab]# jmap -histo 1387

 num     #instances         #bytes  class name
----------------------------------------------
   1:           394         940696  [I
   2:          1461         140256  [C
   3:           444          50664  java.lang.Class
   4:           188          46496  [B
   5:          1157          27768  java.lang.String
   6:           510          25632  [Ljava.lang.Object;
   7:           103           7416  java.lang.reflect.Field
   8:           258           4128  java.lang.Integer
   9:            92           3680  java.lang.ref.SoftReference
  10:           114           3648  java.util.Hashtable$Entry
  11:            71           2496  [Ljava.lang.String;
  12:           103           2472  java.lang.StringBuilder
  13:             6           2256  java.lang.Thread
  14:            58           1856  java.io.File
  15:            38           1824  sun.util.locale.LocaleObjectCache$CacheEntry
  16:            12           1760  [Ljava.util.Hashtable$Entry;
  17:            23           1472  java.net.URL
  18:            45           1440  java.util.concurrent.ConcurrentHashMap$Node
  19:             2           1064  [Ljava.lang.invoke.MethodHandle;
  20:             1           1040  [Ljava.lang.Integer;
  21:            12           1024  [Ljava.util.HashMap$Node;
  22:            28            896  java.util.HashMap$Node
  23:            20            800  sun.util.locale.BaseLocale$Key
  24:            19            760  java.io.ObjectStreamField
  25:            15            720  java.util.HashMap
  26:            18            720  java.util.LinkedHashMap$Entry
  27:            19            608  java.util.Locale
  28:            19            608  sun.util.locale.BaseLocale
  29:             7            560  [S
  30:            10            560  java.lang.Class$ReflectionData
  31:             5            528  [Ljava.util.concurrent.ConcurrentHashMap$Node;
  32:            21            504  java.net.Parts
  33:             7            488  [Ljava.lang.reflect.Field;
  34:             6            480  java.lang.reflect.Constructor
  35:            20            480  java.util.Locale$LocaleKey
  36:            20            480  sun.security.action.GetPropertyAction
  37:            28            448  java.lang.Object
  38:            18            432  java.io.ExpiringCache$Entry
  39:             9            432  sun.misc.URLClassPath$JarLoader
  40:            10            400  java.security.AccessControlContext
  41:             1            384  java.lang.ref.Finalizer$FinalizerThread
  42:             6            384  java.nio.DirectByteBuffer
  43:             6            384  java.util.concurrent.ConcurrentHashMap
  44:             1            376  java.lang.ref.Reference$ReferenceHandler
  45:             6            336  java.nio.DirectLongBufferU
  46:            10            320  java.lang.OutOfMemoryError
  47:            13            312  java.lang.StringBuffer
  48:             9            240  [Ljava.io.ObjectStreamField;
  49:             3            240  [Ljava.util.WeakHashMap$Entry;
```

```
  50:             6            240  java.lang.ref.Finalizer
  51:            10            240  sun.misc.URLClassPath$3
  52:            13            232  [Ljava.lang.Class;
  53:             7            224  java.util.Vector
  54:             9            216  sun.misc.MetaIndex
  55:             5            200  java.util.WeakHashMap$Entry
  56:             6            192  java.lang.ref.ReferenceQueue
  57:             4            192  java.util.Hashtable
  58:             2            176  java.lang.reflect.Method
  59:             7            168  java.util.ArrayList
  60:             3            168  sun.nio.cs.UTF_8$Encoder
  61:             5            160  java.io.FileDescriptor
  62:             3            144  java.nio.HeapByteBuffer
  63:             3            144  java.util.WeakHashMap
  64:             6            144  sun.misc.PerfCounter
  65:             2            128  java.io.ExpiringCache$1
  66:             8            128  java.lang.ref.ReferenceQueue$Lock
  67:             3            120  java.security.ProtectionDomain
  68:             1            104  sun.net.www.protocol.file.FileURLConnection
  69:             3             96  java.io.FileInputStream
  70:             2             96  java.lang.ThreadGroup
  71:             2             96  java.nio.HeapCharBuffer
  72:             3             96  java.security.CodeSource
  73:             2             96  java.util.Properties
  74:             3             96  java.util.Stack
  75:             2             96  java.util.StringTokenizer
  76:             1             96  sun.misc.Launcher$AppClassLoader
  77:             2             96  sun.misc.URLClassPath
  78:             2             96  sun.nio.cs.StreamEncoder
  79:             2             88  [Ljava.net.URL;
  80:             1             88  sun.misc.Launcher$ExtClassLoader
  81:             1             80  [Ljava.lang.ThreadLocal$ThreadLocalMap$Entry;
  82:             2             80  java.io.BufferedWriter
  83:             2             80  java.io.ExpiringCache
  84:             2             80  sun.nio.cs.UTF_8$Decoder
  85:             3             72  [Ljava.lang.reflect.Constructor;
  86:             3             72  java.lang.Class$1
  87:             3             72  java.lang.RuntimePermission
  88:             3             72  java.util.Collections$SynchronizedSet
  89:             3             72  sun.misc.Signal
  90:             3             72  sun.reflect.NativeConstructorAccessorImpl
  91:             2             64  [Ljava.lang.Thread;
  92:             2             64  java.io.FileOutputStream
  93:             2             64  java.io.FilePermission
  94:             2             64  java.io.PrintStream
  95:             4             64  java.lang.Class$3
  96:             2             64  java.lang.ClassValue$Entry
  97:             2             64  java.lang.NoSuchMethodError
  98:             2             64  java.lang.ThreadLocal$ThreadLocalMap$Entry
  99:             2             64  java.lang.VirtualMachineError
 100:             2             64  java.lang.ref.ReferenceQueue$Null
 101:             2             56  [Ljava.io.File;

 102:             1             48  [J
```

```
 103:             2             48  [Ljava.lang.reflect.Method;
 104:             3             48  [Ljava.security.Principal;
 105:             2             48  java.io.BufferedOutputStream
 106:             1             48  java.io.BufferedReader
 107:             2             48  java.io.File$PathStatus
 108:             2             48  java.io.OutputStreamWriter
 109:             3             48  java.lang.ThreadLocal
 110:             2             48  java.net.URLClassLoader$1
 111:             2             48  java.nio.charset.CoderResult
 112:             3             48  java.nio.charset.CodingErrorAction
 113:             3             48  java.security.ProtectionDomain$Key
 114:             1             48  java.util.Hashtable$Enumerator
 115:             2             48  sun.misc.NativeSignalHandler
 116:             1             48  sun.nio.cs.StreamDecoder
 117:             3             48  sun.reflect.DelegatingConstructorAccessorImpl
 118:             3             48  sun.reflect.ReflectionFactory$GetReflectionFactoryAction
 119:             1             40  java.io.BufferedInputStream
 120:             1             40  java.lang.ClassLoader$NativeLibrary
 121:             1             40  java.lang.ClassNotFoundException
 122:             1             40  sun.nio.cs.StandardCharsets$Aliases
 123:             1             40  sun.nio.cs.StandardCharsets$Cache
 124:             1             40  sun.nio.cs.StandardCharsets$Classes
 125:             1             32  [Ljava.lang.OutOfMemoryError;
 126:             2             32  [Ljava.lang.StackTraceElement;
 127:             1             32  [Ljava.lang.ThreadGroup;
 128:             2             32  java.io.FileInputStream$1
 129:             2             32  java.io.FilePermission$1
 130:             1             32  java.io.UnixFileSystem
 131:             1             32  java.lang.ArithmeticException
 132:             2             32  java.lang.Boolean
 133:             1             32  java.lang.NullPointerException
 134:             1             32  java.lang.StringCoding$StringDecoder
 135:             1             32  java.lang.StringCoding$StringEncoder
 136:             2             32  java.nio.ByteOrder
 137:             1             32  java.security.BasicPermissionCollection
 138:             1             32  java.security.Permissions
 139:             2             32  java.util.HashSet
 140:             2             32  java.util.concurrent.atomic.AtomicInteger
 141:             1             32
java.util.concurrent.atomic.AtomicReferenceFieldUpdater$AtomicReferenceFieldUpdaterImpl
 142:             1             32  sun.misc.URLClassPath$FileLoader$1
 143:             2             32  sun.net.www.protocol.jar.Handler
 144:             1             32  sun.nio.cs.StandardCharsets
 145:             2             32  sun.reflect.ReflectionFactory$1
 146:             1             24  [Ljava.io.File$PathStatus;
 147:             1             24  [Ljava.lang.ClassValue$Entry;
 148:             1             24  [Lsun.launcher.LauncherHelper;
 149:             1             24  java.io.FilePermissionCollection
 150:             1             24  java.io.FileReader
 151:             1             24  java.lang.Class$MethodArray
 152:             1             24  java.lang.ClassValue$Version
 153:             1             24  java.lang.ThreadLocal$ThreadLocalMap

 154:             1             24  java.lang.invoke.MethodHandleImpl$4
```

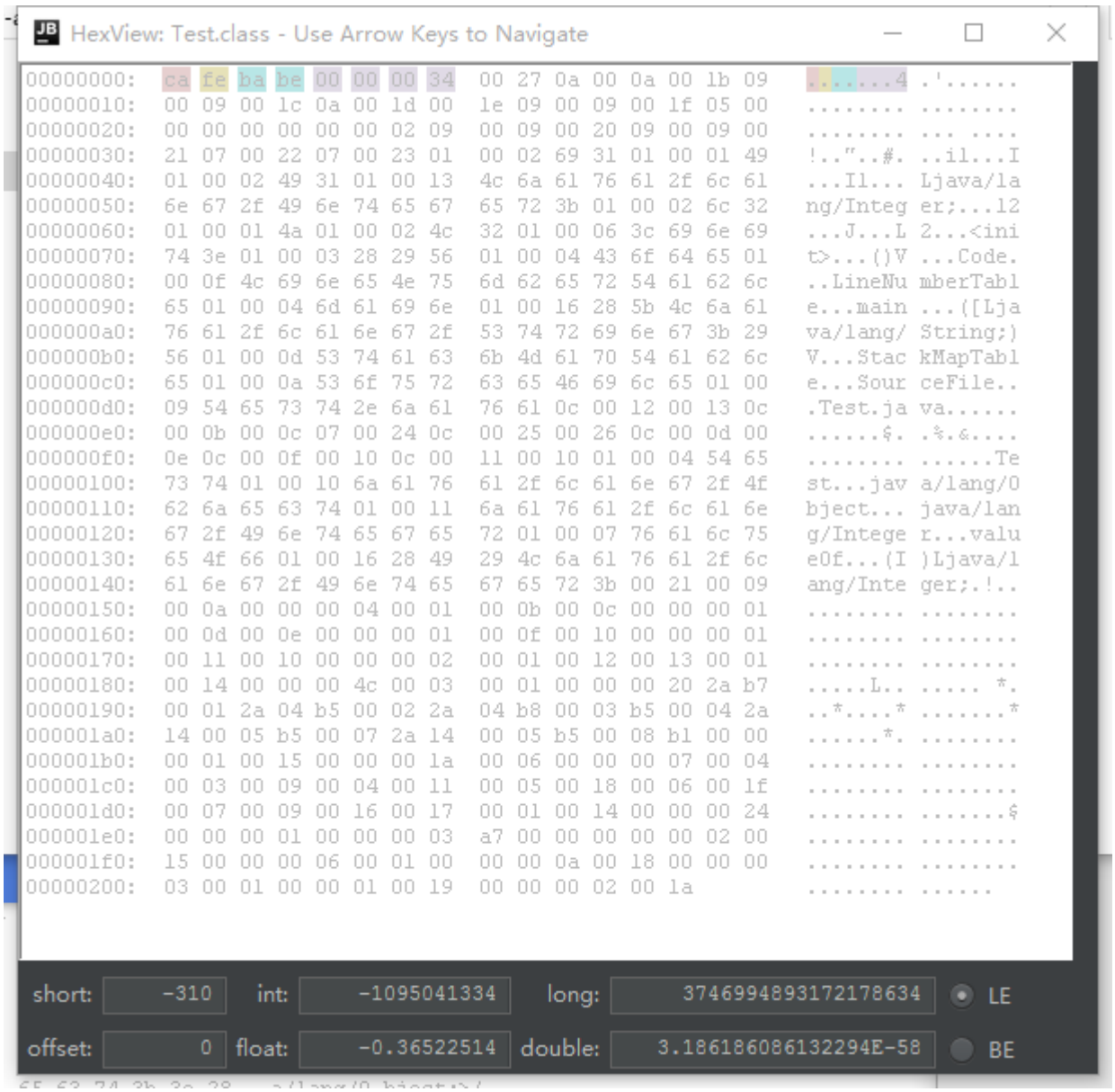
```
 155:             1             24  java.lang.reflect.ReflectPermission
 156:             1             24  java.util.BitSet
 157:             1             24  java.util.Collections$EmptyMap
 158:             1             24  java.util.Collections$SetFromMap
 159:             1             24  java.util.Collections$UnmodifiableRandomAccessList
 160:             1             24  java.util.Locale$Cache
 161:             1             24
java.util.concurrent.atomic.AtomicReferenceFieldUpdater$AtomicReferenceFieldUpdaterImpl$1
 162:             1             24  sun.launcher.LauncherHelper
 163:             1             24  sun.misc.Launcher$AppClassLoader$1
 164:             1             24  sun.misc.URLClassPath$FileLoader
 165:             1             24  sun.net.www.MessageHeader
 166:             1             24  sun.nio.cs.UTF_8
 167:             1             24  sun.util.locale.BaseLocale$Cache
 168:             1             16  [Ljava.lang.Throwable;
 169:             1             16  [Ljava.security.cert.Certificate;
 170:             1             16  java.io.FileDescriptor$1
 171:             1             16  java.lang.CharacterDataLatin1
 172:             1             16  java.lang.ClassLoader$3
 173:             1             16  java.lang.ClassValue$Identity
 174:             1             16  java.lang.Compiler$1
 175:             1             16  java.lang.Runtime
 176:             1             16  java.lang.String$CaseInsensitiveComparator
 177:             1             16  java.lang.System$2
 178:             1             16  java.lang.SystemClassLoaderAction
 179:             1             16  java.lang.Terminator$1
 180:             1             16  java.lang.invoke.MemberName$Factory
 181:             1             16  java.lang.invoke.MethodHandleImpl$1
 182:             1             16  java.lang.invoke.MethodHandleImpl$2
 183:             1             16  java.lang.invoke.MethodHandleImpl$3
 184:             1             16  java.lang.invoke.MethodHandleStatics$1
 185:             1             16  java.lang.ref.Reference$Lock
 186:             1             16  java.lang.reflect.ReflectAccess
 187:             1             16  java.net.URLClassLoader$7
 188:             1             16  java.nio.Bits$1
 189:             1             16  java.nio.charset.CoderResult$1
 190:             1             16  java.nio.charset.CoderResult$2
 191:             1             16  java.security.ProtectionDomain$1
 192:             1             16  java.security.ProtectionDomain$3
 193:             1             16  java.util.Collections$EmptyList
 194:             1             16  java.util.Collections$EmptySet
 195:             1             16  java.util.Hashtable$EntrySet
 196:             1             16  java.util.WeakHashMap$KeySet
 197:             1             16  java.util.zip.ZipFile$1
 198:             1             16  sun.misc.Launcher
 199:             1             16  sun.misc.Launcher$ExtClassLoader$1
 200:             1             16  sun.misc.Launcher$Factory
 201:             1             16  sun.misc.Perf
 202:             1             16  sun.misc.Perf$GetPerfAction
 203:             1             16  sun.misc.Unsafe
 204:             1             16  sun.net.www.protocol.file.Handler
 205:             1             16  sun.reflect.ReflectionFactory

Total          5755        1291736
```

```
[root@t1kvm01 lab]#
```

bytes无法整除instances，因为有些bytes是用来字节对其的？

https://docs.oracle.com/javase/specs/jvms/se7/html/jvms-4.html

```
ClassFile {
    u4             magic;
    u2             minor_version;
    u2             major_version;
    u2             constant_pool_count;
    cp_info        constant_pool[constant_pool_count-1];
    u2             access_flags;
    u2             this_class;
    u2             super_class;
    u2             interfaces_count;
    u2             interfaces[interfaces_count];
    u2             fields_count;
    field_info     fields[fields_count];
    u2             methods_count;
    method_info    methods[methods_count];
    u2             attributes_count;
    attribute_info attributes[attributes_count];
}
```



field	value
magic	ca fe ba be
minor_version	00 00
major_version	00 34
constant_pool_count	00 27(常量池大小 39)

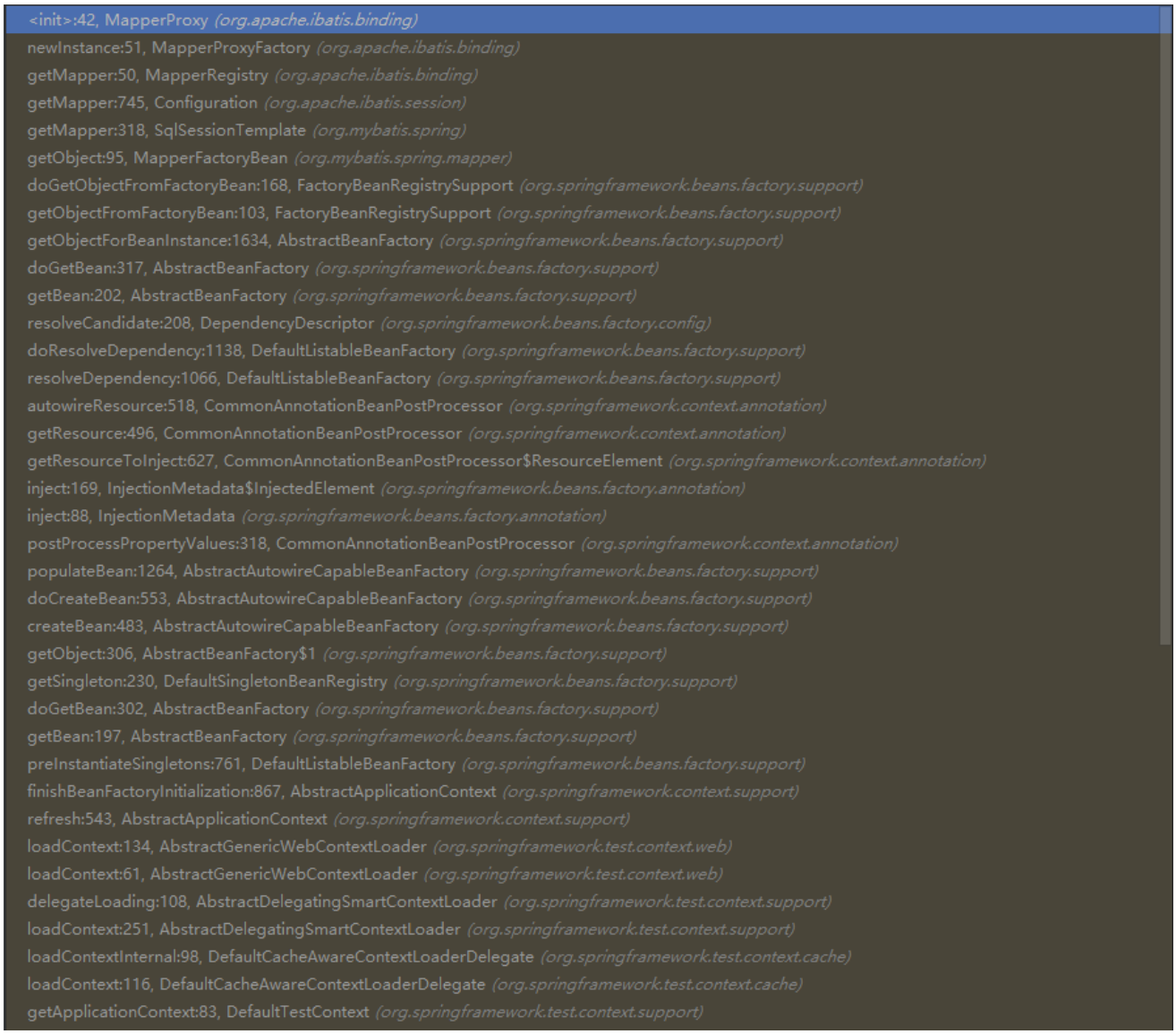
Q:把mybatis库引入之后，它是如何执行的呢？没有搜索到main()函数，那就只有通过spring加载了，如何加载？

A: SqlSessionFactoryBean为mybatis-spring的入口类

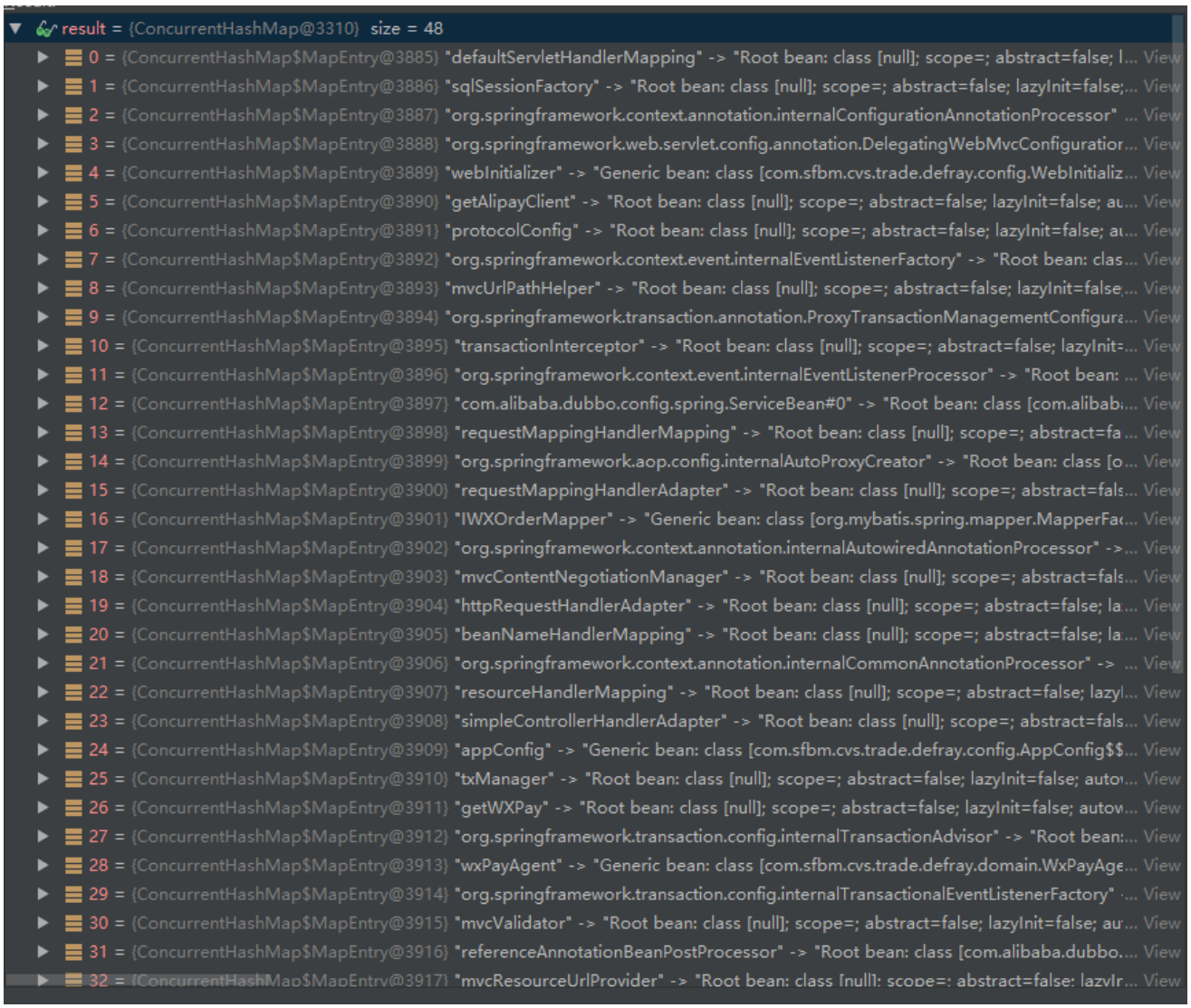
XMLConfigBuilder是mybatis的入口类之一 创建sqlSessionFactory 初始化configuration environment transcationManager datasource

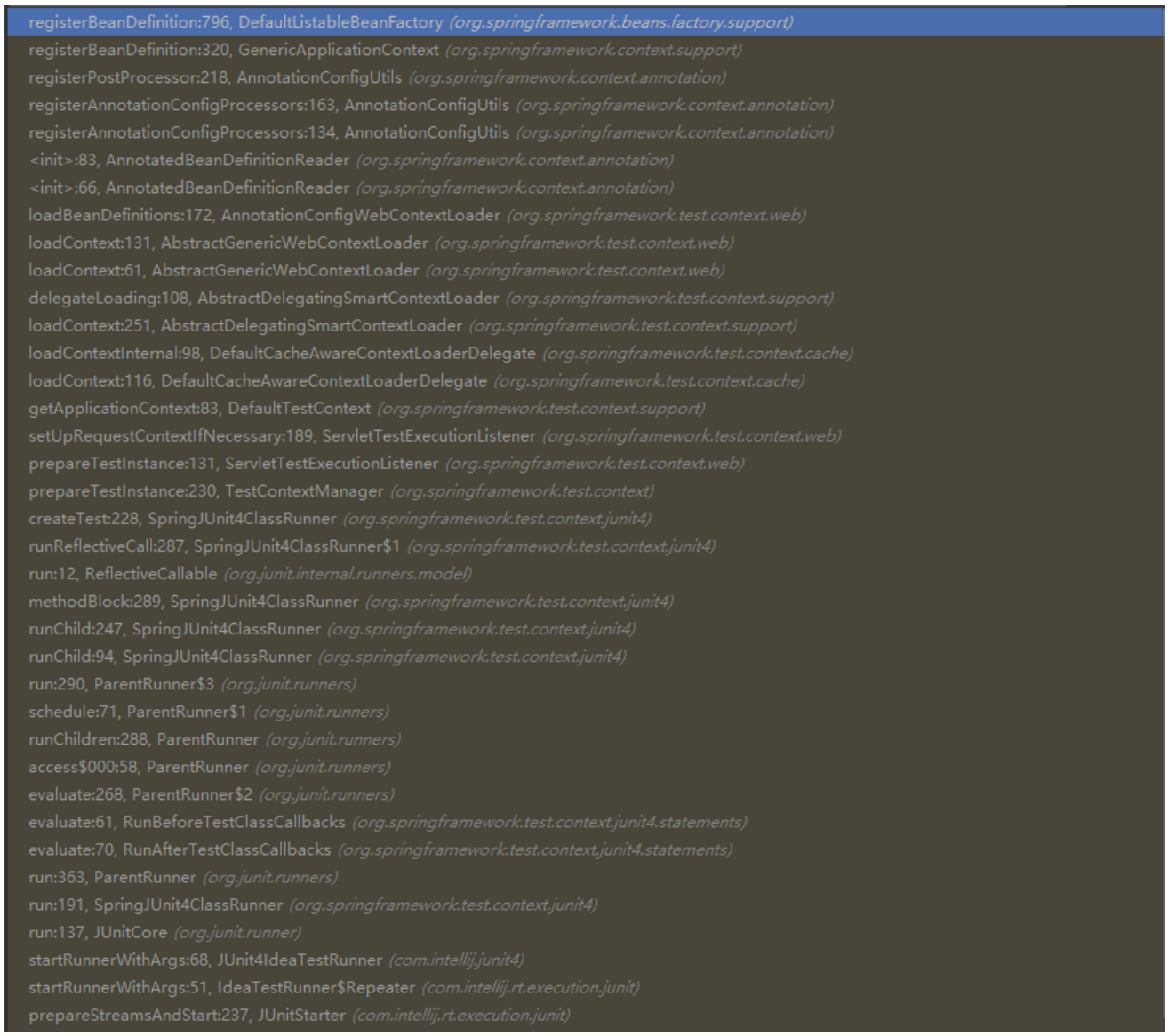
Q:调用dao接口时debug发现实现对象为MapperProxy，MapperProxy是何时生成&如何注入的？

在MapperProxy构造函数打断点，查看调用栈




```
    BeanDefinition getBeanDefinition(String beanName) throws NoSuchBeanDefinitionException
    //通过名称从beanDefinitionMap(registerBeanDefinition)中获取BeanDefinition
```


result = {ConcurrentHashMap@3310} size = 48
0 = {ConcurrentHashMap$MapEntry@3885} "defaultServletHandlerMapping" -> "Root bean: class [null]; scope=; abstract=false; l... View
1 = {ConcurrentHashMap$MapEntry@3886} "sqlSessionFactory" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false;... View
2 = {ConcurrentHashMap$MapEntry@3887} "org.springframework.context.annotation.internalConfigurationAnnotationProcessor" ... View
3 = {ConcurrentHashMap$MapEntry@3888} "org.springframework.web.servlet.config.annotation.DelegatingWebMvcConfiguratior... View
4 = {ConcurrentHashMap$MapEntry@3889} "webInitializer" -> "Generic bean: class [com.sfbm.cvs.trade.defray.config.WebInitializ... View
5 = {ConcurrentHashMap$MapEntry@3890} "getAlipayClient" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false; au... View
6 = {ConcurrentHashMap$MapEntry@3891} "protocolConfig" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false; au... View
7 = {ConcurrentHashMap$MapEntry@3892} "org.springframework.context.event.internalEventListenerFactory" -> "Root bean: clas... View
8 = {ConcurrentHashMap$MapEntry@3893} "mvcUrlPathHelper" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false;... View
9 = {ConcurrentHashMap$MapEntry@3894} "org.springframework.transaction.annotation.ProxyTransactionManagementConfigura... View
10 = {ConcurrentHashMap$MapEntry@3895} "transactionInterceptor" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=... View
11 = {ConcurrentHashMap$MapEntry@3896} "org.springframework.context.event.internalEventListenerProcessor" -> "Root bean: ... View
12 = {ConcurrentHashMap$MapEntry@3897} "com.alibaba.dubbo.config.spring.ServiceBean#0" -> "Root bean: class [com.alibab... View
13 = {ConcurrentHashMap$MapEntry@3898} "requestMappingHandlerMapping" -> "Root bean: class [null]; scope=; abstract=fa... View
14 = {ConcurrentHashMap$MapEntry@3899} "org.springframework.aop.config.internalAutoProxyCreator" -> "Root bean: class [o... View
15 = {ConcurrentHashMap$MapEntry@3900} "requestMappingHandlerAdapter" -> "Root bean: class [null]; scope=; abstract=fals... View
16 = {ConcurrentHashMap$MapEntry@3901} "IWXOrderMapper" -> "Generic bean: class [org.mybatis.spring.mapper.MapperFac... View
17 = {ConcurrentHashMap$MapEntry@3902} "org.springframework.context.annotation.internalAutowiredAnnotationProcessor" ->... View
18 = {ConcurrentHashMap$MapEntry@3903} "mvcContentNegotiationManager" -> "Root bean: class [null]; scope=; abstract=fals... View
19 = {ConcurrentHashMap$MapEntry@3904} "httpRequestHandlerAdapter" -> "Root bean: class [null]; scope=; abstract=false; la... View
20 = {ConcurrentHashMap$MapEntry@3905} "beanNameHandlerMapping" -> "Root bean: class [null]; scope=; abstract=false; la... View
21 = {ConcurrentHashMap$MapEntry@3906} "org.springframework.context.annotation.internalCommonAnnotationProcessor" -> ... View
22 = {ConcurrentHashMap$MapEntry@3907} "resourceHandlerMapping" -> "Root bean: class [null]; scope=; abstract=false; lazyI... View
23 = {ConcurrentHashMap$MapEntry@3908} "simpleControllerHandlerAdapter" -> "Root bean: class [null]; scope=; abstract=fals... View
24 = {ConcurrentHashMap$MapEntry@3909} "appConfig" -> "Generic bean: class [com.sfbm.cvs.trade.defray.config.AppConfig$$... View
25 = {ConcurrentHashMap$MapEntry@3910} "txManager" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false; autow... View
26 = {ConcurrentHashMap$MapEntry@3911} "getWXPay" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false; autow... View
27 = {ConcurrentHashMap$MapEntry@3912} "org.springframework.transaction.config.internalTransactionAdvisor" -> "Root bean:... View
28 = {ConcurrentHashMap$MapEntry@3913} "wxPayAgent" -> "Generic bean: class [com.sfbm.cvs.trade.defray.domain.WxPayAge... View
29 = {ConcurrentHashMap$MapEntry@3914} "org.springframework.transaction.config.internalTransactionalEventListenerFactory" ... View
30 = {ConcurrentHashMap$MapEntry@3915} "mvcValidator" -> "Root bean: class [null]; scope=; abstract=false; lazyInit=false; au... View
31 = {ConcurrentHashMap$MapEntry@3916} "referenceAnnotationBeanPostProcessor" -> "Root bean: class [com.alibaba.dubbo.... View
32 = {ConcurrentHashMap$MapEntry@3917} "mvcResourceUrlProvider" -> "Root bean: class [null]; scope=; abstract=false; lazyIr... View



A: 搜索业务代码使用的MapperScan注解，调用关系如下：

```
MapperScannerRegistrar:ImportBeanDefinitionRegistrar$registerBeanDefinitions<-
ConfigurationClassParser$processImports<-
ConfigurationClassPostProcessor$postProcessBeanDefinitionRegistry<-
```

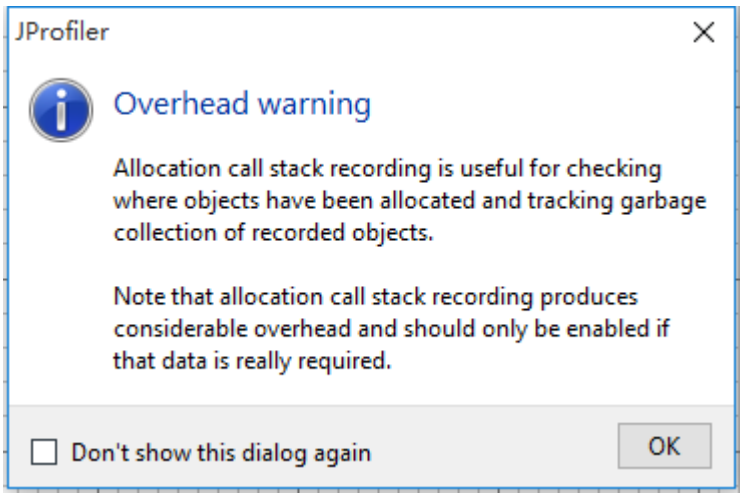
ImportBeanDefinitionRegistrar是spring提供的bean注册入口之一，spring搜索实现了该Interface的类，

实例化并添加到importBeanDefinitionRegistrars中，后续处理中调用registerBeanDefinitions创建bean。

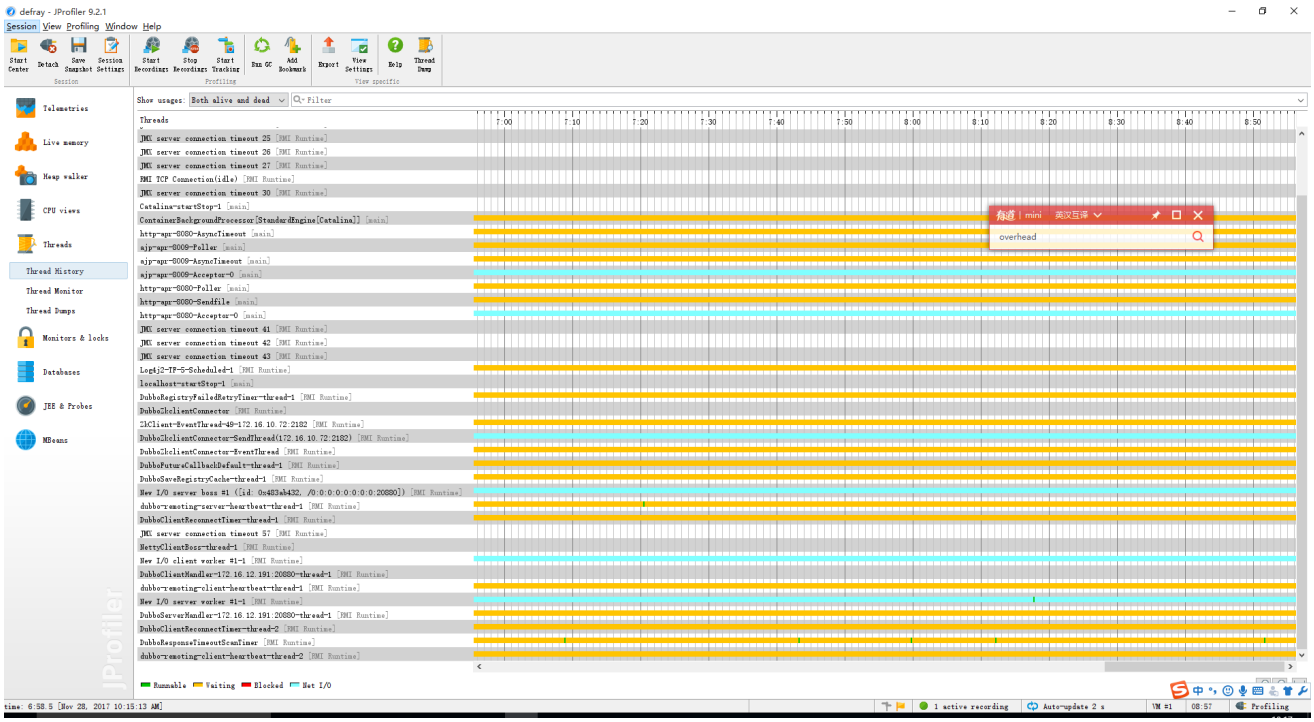
- Mybatis-spring的初始化通过spring提供的钩子ImportBeanDefinitionRegistrar实现，初始化内容包括加载Sqlmapper，configuration，实现。而提供给向业务层的功能入口则是SqlSesssionFactory，业务层通过创建 SqlSessionFactory Bean来

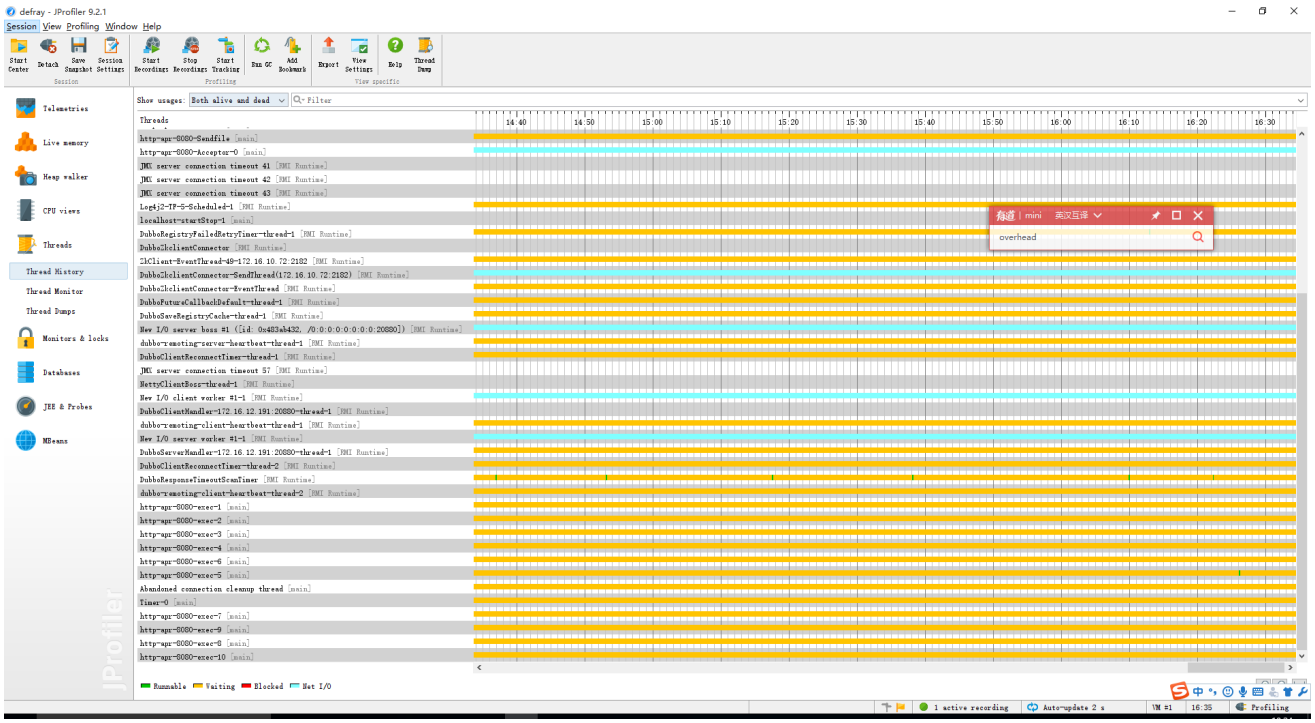
loadContext spring入口函数

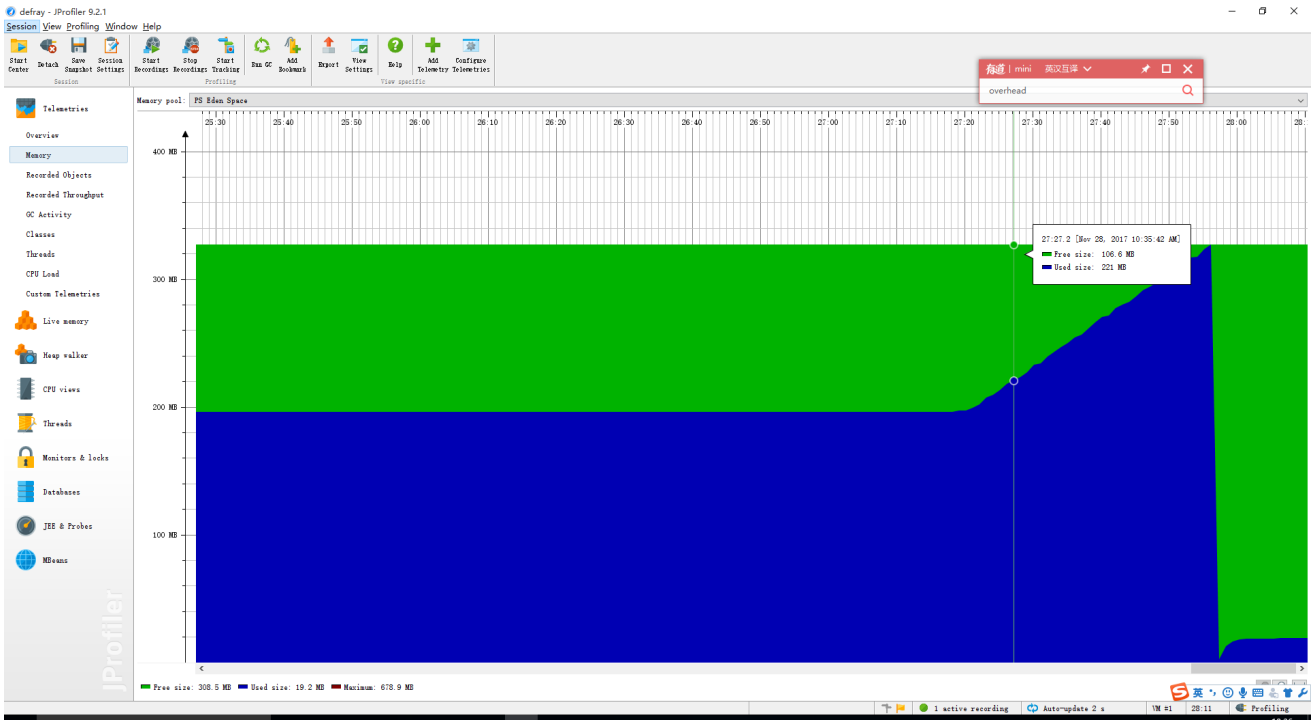
```
DubboComponentScanRegistrar
```



内存分配调用栈记录对于对象在哪分配和最终垃圾回收很有用，但是会产生相当一部分额外开销。







- 编写插件生成enum类
- 二方库里可以定义枚举类型，参数可以使用枚举类型，但是接口返回值不允许使用枚

举类型或者包含枚举类型的 POJO 对象。

- spring DAO
- 受检异常/非受检异常 ？
- 线程状态转换
- sync同步优化策略

- 阅读effitive java,完成alibaba编程规范认证,完成细节重构
- 阅读《重构 改善既有代码设计》,《reactive design pattern》完成随便充系统服务划分与设计

- 阅读mybatis,concurrentHashMap源码,部分spring-bean,spring-context源码,《深入理解java虚拟机》,《并发编程实践》,修改预防常见并发问题。

- dubbo 配置服务端方法级别 loadbalance, 配置不同protocal
- dubbo 服务端方法级别设置线程池，编写扩展线程池，根据参数做ConsistentHash，以实现部分接口的串行化调用

```
ExtensionLoader.getExtensionLoader(ThreadPool.class).getAdaptiveExtension().getExecutor(url);
```

```
name.substring(0, name.length() - type.getSimpleName().length());
```
获取扩展名和key的映射