

foodelivery.jps - JProfiler 12.0.4

Session View Profiling Window Help

Start Center Details Save Session Settings

Telemetries Live Memory Heap Walker CPU Views Threads Monitors & Locks Databases JEE & Probes MBeans

Run GC Add Bookmark Export View Settings Help Take Snapshot Mark Heap Back Forward Go To Start Show Selection

View Specific

Classes Allocations Biggest Objects References Time Inspections Graph

Object groups

Priority	Class Loader	Instance Count	Shallow Size
1	Default class loader	38,830	2,149 kB
2	jdk.internal.loader.ClassLoaders\$PlatformClassLoader (0x954)	7	1,376 bytes
3	jdk.internal.loader.ClassLoaders\$AppClassLoader (0x953)	1	320 bytes

Current object set: 38,830 objects in 728 classes.

3 selection steps, 2,149 kB shallow size, [Calculate retained and deep sizes](#) [Use retained objects](#)

Classes Use ... Group By Class Loaders [Calculate estimated retained sizes](#)

Name	Instance Count	Size
byte[]	8,227	470 kB
java.lang.String	8,030	192 kB
java.util.HashMap\$Node	3,111	99,552 bytes
java.lang.Object[]	2,453	152 kB
java.util.concurrent.ConcurrentHashMap\$Node	1,841	58,917 bytes
<b>Total:</b>	<b>38,830</b>	<b>2,149 kB</b>

Selection step 2: Inspection "Instances grouped by class loaders"

Selected group: **Default class loader**

38,830 objects in 728 classes.

Selection step 1: All objects, after full GC, retaining soft references

38,838 objects in 731 classes.

Unlicensed copy for evaluation purposes, 9 days remaining

19 recordings Dec 7, 2021 11:11:16 AM VM #1 03:55 Snapshot 12:01 07-12-2021

foodelivery.jps - JProfiler 12.0.4

Session View Profiling Window Help

Start Center Details Save Session Settings

Telemetries Live Memory Heap Walker CPU Views Threads Monitors & Locks Databases JEE & Probes HTTP Server HTTP Client Web Services JNDI JMS RMI Class Loaders Exceptions

Run GC Add Bookmark Export View Settings Help Record Probe Class Loaders

View Specific

Time Line Class Loaders Call Tree Telemetries Events Tracker

Class Loaders

Class loaders and their loaded classes

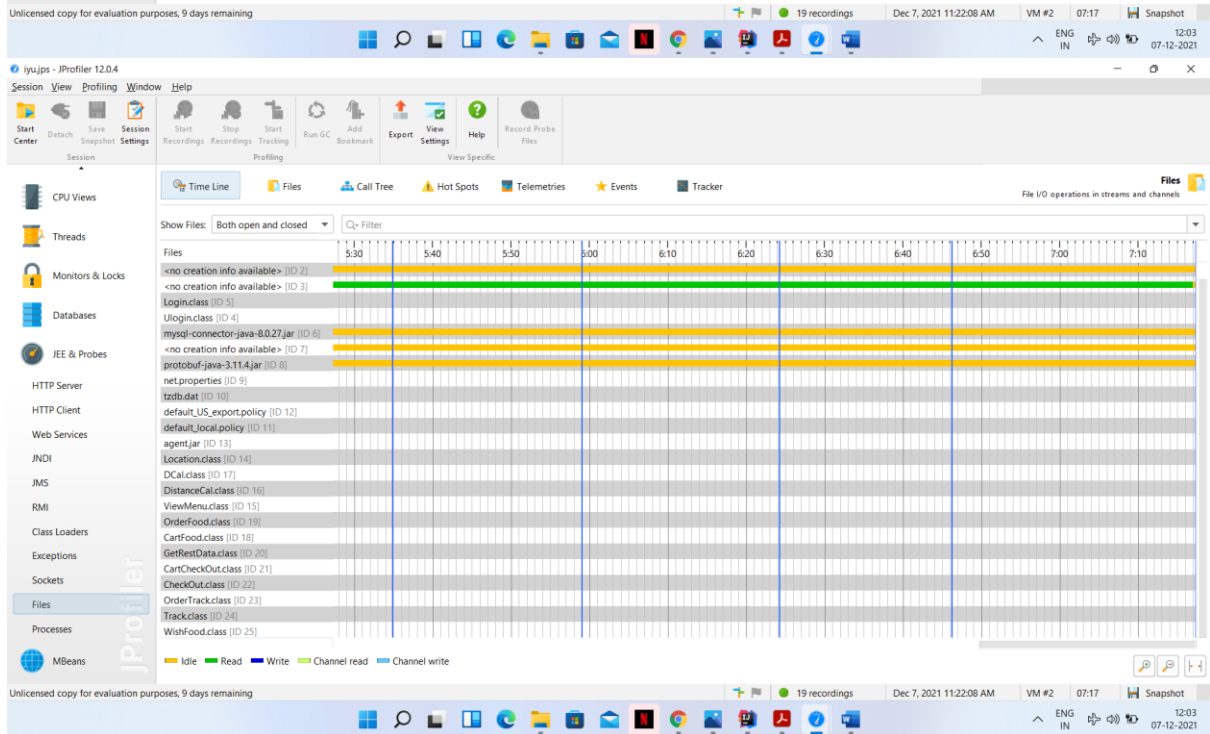
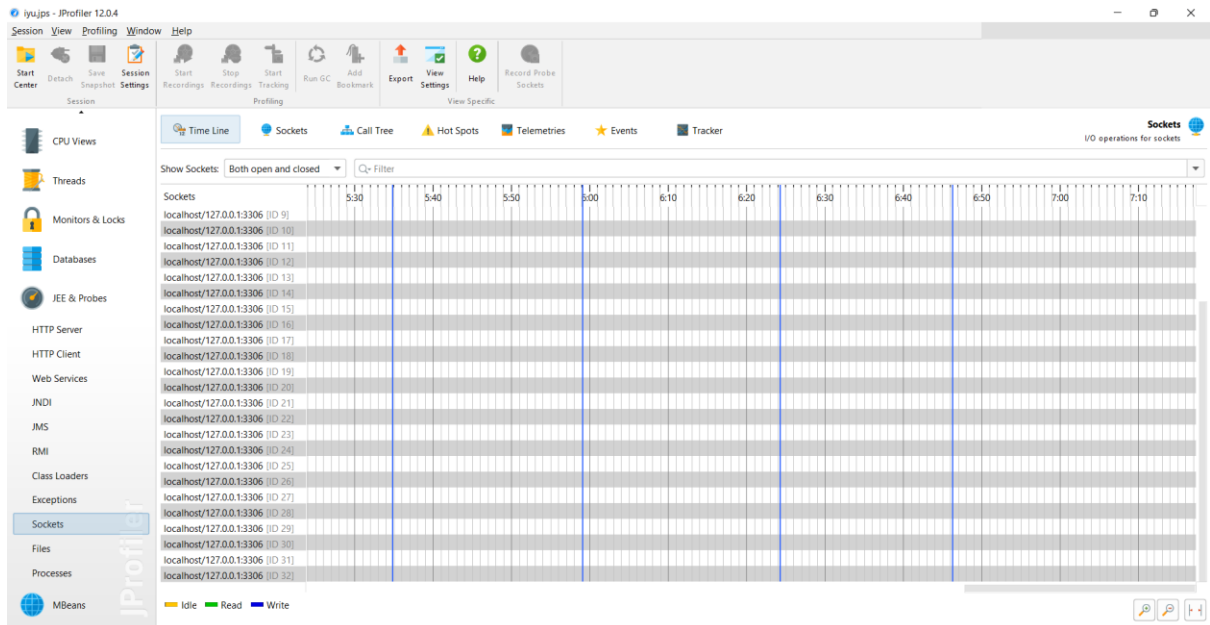
Show Class Loaders: Both open and closed Q-Filter

Class Loaders	5:40	5:50	6:00	6:10	6:20	6:30	6:40	6:50	7:00	7:10
jdk.internal.loader.ClassLoaders\$AppClassLoader (ID 1)										
jdk.internal.loader.ClassLoaders\$PlatformClassLoader (ID 2)										
jdk.internal.reflect.DelegatingClassLoader (ID 3)										
jdk.internal.reflect.DelegatingClassLoader (ID 4)										
jdk.internal.reflect.DelegatingClassLoader (ID 5)										
jdk.internal.reflect.DelegatingClassLoader (ID 6)										
jdk.internal.reflect.DelegatingClassLoader (ID 7)										
jdk.internal.reflect.DelegatingClassLoader (ID 8)										
jdk.internal.reflect.DelegatingClassLoader (ID 12)										
jdk.internal.reflect.DelegatingClassLoader (ID 11)										
jdk.internal.reflect.DelegatingClassLoader (ID 10)										
jdk.internal.reflect.DelegatingClassLoader (ID 9)										
jdk.internal.reflect.DelegatingClassLoader (ID 13)										

Idle Class load

Unlicensed copy for evaluation purposes, 9 days remaining

19 recordings Dec 7, 2021 11:22:08 AM VM #2 07:17 Snapshot 12:02 07-12-2021



Session

View

Profiling

Window

Help

Start Center

Details

Save Snapshot

Session Settings

Start Recordings

Stop Recordings

Start Tracking

Run GC

Add Bookmark

Export

View Settings

Help

Show Legend

Record CPU

Show Graph

Analyze

Session

Profiling

View Specific

Telemetries

Live Memory

Heap Walker

CPU Views

Call Tree

Hot Spots

Call Graph

Outlier Detection

Complexity Analysis

Call Tracer

JavaScript XHR

Threads

Monitors & Locks

Databases

JEE & Probes

MRBeans

Allocation recording was active at the same time as CPU recording. Measured times may be distorted significantly.

Thread status: Runnable Thread selection: All thread groups Aggregation level: Methods Hot spot options: Self times

Async tracking: Thread start, Executors and 3 other tracking types [Inline Async Executions](#)

Hot Spot	Self Time	Average Time	Invocations
> java.util.Scanner.nextInt	13,413 ms (3 %)	17,509 ms	24
> java.util.Scanner.next	2,235 ms	2,235 ms	6
> java.sql.DriverManager.getConnection	1,161 ms (0 %)	36,288 μs	32
> java.util.Scanner.<init>	57,888 μs (0 %)	7,236 μs	8
> java.sql.Statement.executeQuery	52,152 μs (0 %)	2,173 μs	24
> java.lang.Class.forName	38,642 μs (0 %)	920 μs	42
> com.mycompany.fooddelivery.ViewMenu.menu	37,773 μs (0 %)	37,773 μs	1
> com.mycompany.fooddelivery.Location.viewLocations	36,373 μs (0 %)	36,373 μs	1
> java.sql.Connection.prepareStatement	34,894 μs (0 %)	8,723 μs	4
> com.mycompany.fooddelivery.FoodDelivery.main	27,280 μs (0 %)	27,280 μs	1
> com.mycompany.fooddelivery.DistanceCal.distance	18,404 μs (0 %)	2,044 μs	9
> com.mycompany.fooddelivery.CartFood.viewCart	17,194 μs (0 %)	17,194 μs	1
> java.sql.Connection.close	15,570 μs (0 %)	502 μs	31
> java.io.PrintStream.println	14,516 μs (0 %)	110 μs	131
> com.mycompany.fooddelivery.DistanceCal.distCal	13,027 μs (0 %)	13,027 μs	1
> com.mycompany.fooddelivery.CartFood.priceCalculation	10,464 μs (0 %)	10,464 μs	1
> com.mycompany.fooddelivery.Login.couponUpdate	10,227 μs (0 %)	10,227 μs	1
> com.mycompany.fooddelivery.CartFood.addToCart	6,156 μs (0 %)	6,156 μs	1
> java.sql.PreparedStatement.execute	4,337 μs (0 %)	1,084 μs	4
> com.mycompany.fooddelivery.OrderTrack.track	2,652 μs (0 %)	2,652 μs	1
> java.sql.Connection.createStatement	2,057 μs (0 %)	73 μs	28
> com.mycompany.fooddelivery.CartCheckout.payment	1,876 μs (0 %)	1,876 μs	1
> java.sql.Statement.executeUpdate	1,613 μs (0 %)	403 μs	4
> com.mycompany.fooddelivery.Login.userCheck	1,480 μs (0 %)	1,480 μs	1
> com.mycompany.fooddelivery.CartCheckout.coupons	1,462 μs (0 %)	1,462 μs	1
> com.mycompany.fooddelivery.GetRestData.nameData	1,086 μs (0 %)	543 μs	2
> java.sql.ResultSet.next	842 μs (0 %)	13 μs	64
> java.sql.ResultSet.getInt(java.lang.String)	614 μs (0 %)	87 μs	7
> java.io.InputStream.readFully(byte[], int, int)	579 μs (0.001 %)	78 μs	7

Q: Hot Spot View Filters

Unlicensed copy for evaluation purposes, 9 days remaining

19 recordings

Dec 7, 2021 11:22:08 AM

VM #2

07:17

Snapshot

ENG IN

12:03

07-12-2021

Session

View

Profiling

Window

Help

Start Center

Details

Save Snapshot

Session Settings

Start Recordings

Stop Recordings

Start Tracking

Run GC

Add Bookmark

Export

View Settings

Help

Calculate Statistics

Record CPU

Session

Profiling

View Specific

Telemetries

Live Memory

Heap Walker

CPU Views

Call Tree

Hot Spots

Call Graph

Outlier Detection

Complexity Analysis

Call Tracer

JavaScript XHR

Threads

Monitors & Locks

Databases

JEE & Probes

MRBeans

Thread status: All states

Method	Total Time	Inv.	Avg. Time	Max. Time	Outlier Coeff.
java.lang.Class.forName(java.lang.String)	39,093 μs	42	930 μs	31,946 μs	33.35
java.util.Scanner.nextInt()	420 s	24	17,509 ms	239 s	12.88
java.sql.DriverManager.getConnection(java.lang.String, java.lang.String, java.lang.String)	1,476 ms	32	46,131 μs	585 ms	11.69
java.sql.Statement.executeQuery(java.lang.String)	293 ms	24	12,231 μs	105 ms	7.6

Q: Class View Filters

Unlicensed copy for evaluation purposes, 9 days remaining

19 recordings

Dec 7, 2021 11:22:08 AM

VM #2

07:17

Snapshot

ENG IN

12:04

07-12-2021

