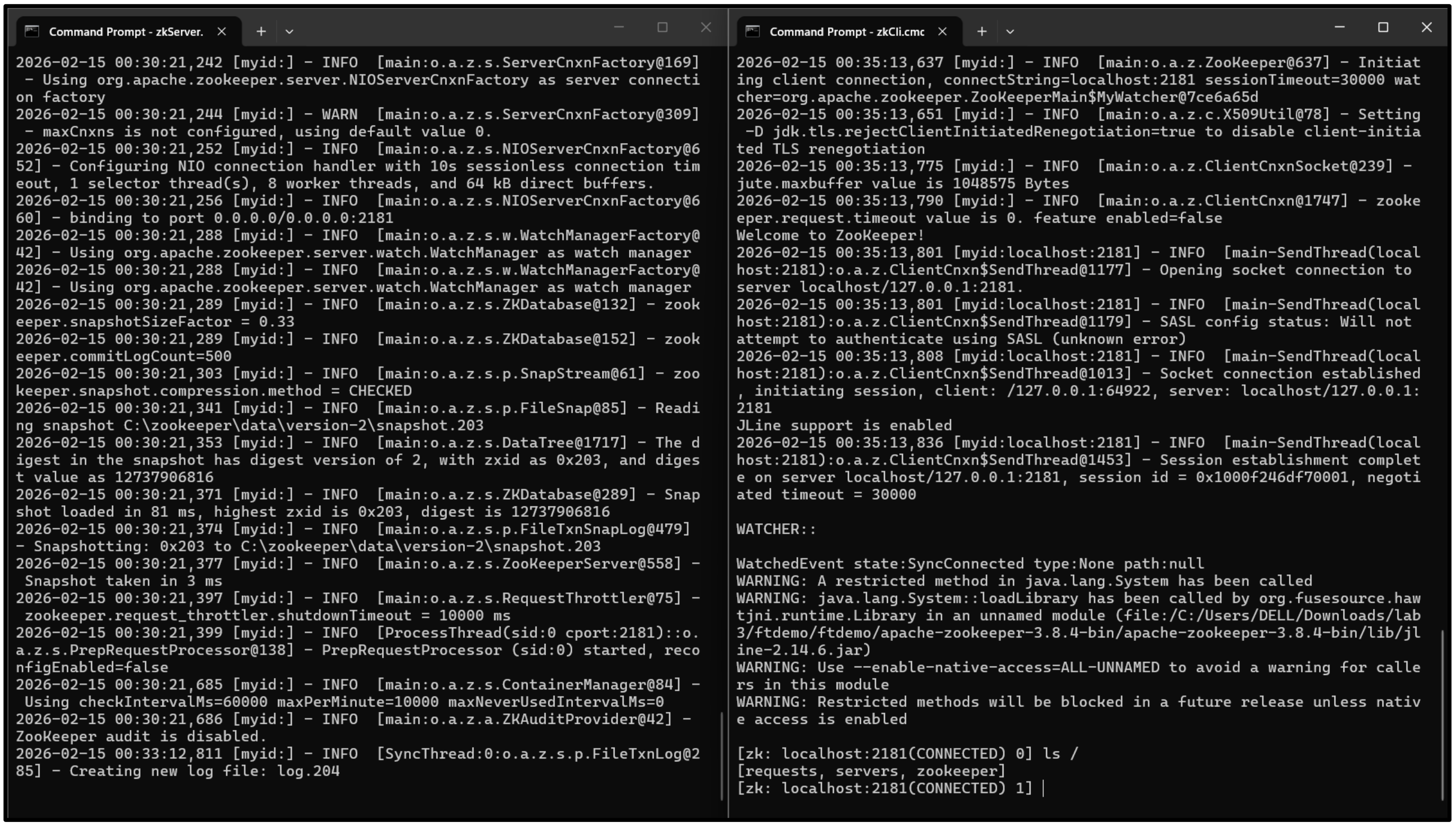
# Lab 3 - Fault-Tolerant Applications with Apache ZooKeeper – IT24103581

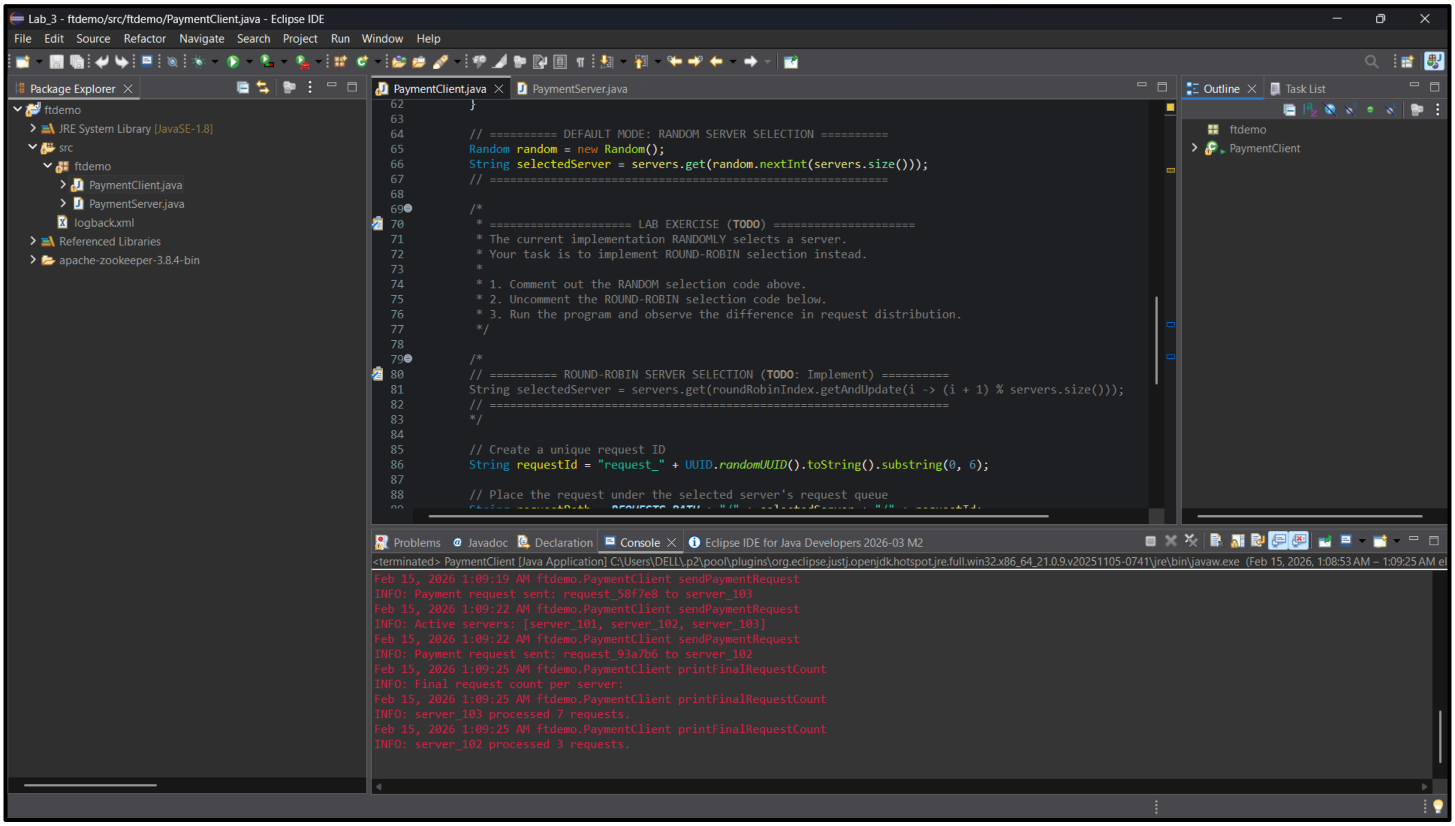
## 1. Running ZooKeeper



```
2026-02-15 00:30:21,242 [myid:] - INFO  [main:o.a.z.s.ServerCnxnFactory@169]
 - Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connecti
on factory
2026-02-15 00:30:21,244 [myid:] - WARN  [main:o.a.z.s.ServerCnxnFactory@309]
 - maxCnxns is not configured, using default value 0.
2026-02-15 00:30:21,252 [myid:] - INFO  [main:o.a.z.s.NIOServerCnxnFactory@6
52] - Configuring NIO connection handler with 10s sessionless connection tim
eout, 1 selector thread(s), 8 worker threads, and 64 kB direct buffers.
2026-02-15 00:30:21,256 [myid:] - INFO  [main:o.a.z.s.NIOServerCnxnFactory@6
60] - binding to port 0.0.0.0/0.0.0.0:2181
2026-02-15 00:30:21,288 [myid:] - INFO  [main:o.a.z.s.w.WatchManagerFactory@
42] - Using org.apache.zookeeper.server.watch.WatchManager as watch manager
2026-02-15 00:30:21,288 [myid:] - INFO  [main:o.a.z.s.w.WatchManagerFactory@
42] - Using org.apache.zookeeper.server.watch.WatchManager as watch manager
2026-02-15 00:30:21,289 [myid:] - INFO  [main:o.a.z.s.ZKDatabase@132] - zook
eeper.snapshotSizeFactor = 0.33
2026-02-15 00:30:21,289 [myid:] - INFO  [main:o.a.z.s.ZKDatabase@152] - zook
eeper.commitLogCount=500
2026-02-15 00:30:21,303 [myid:] - INFO  [main:o.a.z.s.p.SnapStream@61] - zoo
keeper.snapshot.compression.method = CHECKED
2026-02-15 00:30:21,341 [myid:] - INFO  [main:o.a.z.s.p.FileSnap@85] - Readi
ng snapshot C:\zookeeper\data\version-2\snapshot.203
2026-02-15 00:30:21,353 [myid:] - INFO  [main:o.a.z.s.DataTree@1717] - The d
igest in the snapshot has digest version of 2, with zxid as 0x203, and diges
t value as 12737906816
2026-02-15 00:30:21,371 [myid:] - INFO  [main:o.a.z.s.ZKDatabase@289] - Snap
shot loaded in 81 ms, highest zxid is 0x203, digest is 12737906816
2026-02-15 00:30:21,374 [myid:] - INFO  [main:o.a.z.s.p.FileTxnSnapLog@479]
- Snapshotting: 0x203 to C:\zookeeper\data\version-2\snapshot.203
2026-02-15 00:30:21,377 [myid:] - INFO  [main:o.a.z.s.ZooKeeperServer@558] -
 Snapshot taken in 3 ms
2026-02-15 00:30:21,397 [myid:] - INFO  [main:o.a.z.s.RequestThrottler@75] -
 zookeeper.request_throttler.shutdownTimeout = 10000 ms
2026-02-15 00:30:21,399 [myid:] - INFO  [ProcessThread(sid:0 cport:2181)::o.
a.z.s.PrepRequestProcessor@138] - PrepRequestProcessor (sid:0) started, reco
nfigEnabled=false
2026-02-15 00:30:21,685 [myid:] - INFO  [main:o.a.z.s.ContainerManager@84] -
 Using checkIntervalMs=60000 maxPerMinute=10000 maxNeverUsedIntervalMs=0
2026-02-15 00:30:21,686 [myid:] - INFO  [main:o.a.z.a.ZKAuditProvider@42] -
ZooKeeper audit is disabled.
2026-02-15 00:33:12,811 [myid:] - INFO  [SyncThread:0:o.a.z.s.p.FileTxnLog@2
85] - Creating new log file: log.204
```

```
2026-02-15 00:35:13,637 [myid:] - INFO  [main:o.a.z.ZooKeeper@637] - Initiat
ing client connection, connectString=localhost:2181 sessionTimeout=30000 wat
cher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@7ce6a65d
2026-02-15 00:35:13,651 [myid:] - INFO  [main:o.a.z.c.X509Util@78] - Setting
 -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initia
ted TLS renegotiation
2026-02-15 00:35:13,775 [myid:] - INFO  [main:o.a.z.ClientCnxnSocket@239] -
jute.maxbuffer value is 1048575 Bytes
2026-02-15 00:35:13,790 [myid:] - INFO  [main:o.a.z.ClientCnxn@1747] - zooke
eper.request.timeout value is 0. feature enabled=false
Welcome to ZooKeeper!
2026-02-15 00:35:13,801 [myid:localhost:2181] - INFO  [main-SendThread(local
host:2181):o.a.z.ClientCnxn$SendThread@1177] - Opening socket connection to
server localhost/127.0.0.1:2181.
2026-02-15 00:35:13,801 [myid:localhost:2181] - INFO  [main-SendThread(local
host:2181):o.a.z.ClientCnxn$SendThread@1179] - SASL config status: Will not
attempt to authenticate using SASL (unknown error)
2026-02-15 00:35:13,808 [myid:localhost:2181] - INFO  [main-SendThread(local
host:2181):o.a.z.ClientCnxn$SendThread@1013] - Socket connection established
, initiating session, client: /127.0.0.1:64922, server: localhost/127.0.0.1:
2181
JLine support is enabled
2026-02-15 00:35:13,836 [myid:localhost:2181] - INFO  [main-SendThread(local
host:2181):o.a.z.ClientCnxn$SendThread@1453] - Session establishment complet
e on server localhost/127.0.0.1:2181, session id = 0x1000f246df70001, negoti
ated timeout = 30000

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
WARNING: A restricted method in java.lang.System has been called
WARNING: java.lang.System::loadLibrary has been called by org.fusesource.haw
tjni.runtime.Library in an unnamed module (file:/C:/Users/DELL/Downloads/lab
3/ftdemo/ftdemo/apache-zookeeper-3.8.4-bin/apache-zookeeper-3.8.4-bin/lib/jl
ine-2.14.6.jar)
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for calle
rs in this module
WARNING: Restricted methods will be blocked in a future release unless nativ
e access is enabled

[zk: localhost:2181(CONNECTED) 0] ls /
[requests, servers, zookeeper]
[zk: localhost:2181(CONNECTED) 1]
```

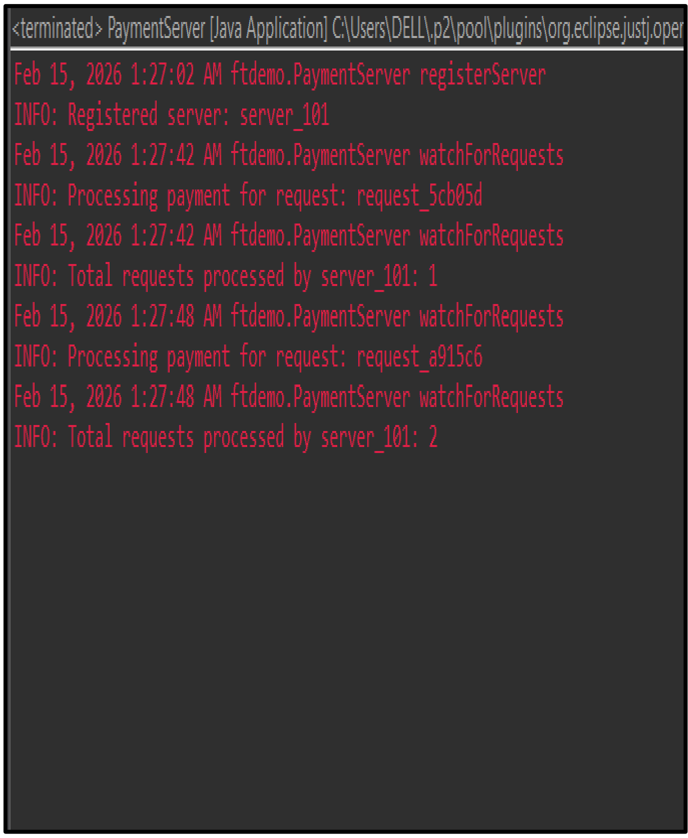
## 2. Running the Payment System


Lab_3 - ftdemo/src/ftdemo/PaymentClient.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ftdemo
JRE System Library [JavaSE-1.8]
src
ftdemo
PaymentClient.java
PaymentServer.java
logback.xml
Referenced Libraries
apache-zookeeper-3.8.4-bin
PaymentClient.java
PaymentServer.java
Outline
Task List
ftdemo
PaymentClient
Problems
Javadoc
Declaration
Console
Eclipse IDE for Java Developers 2026-03 M2

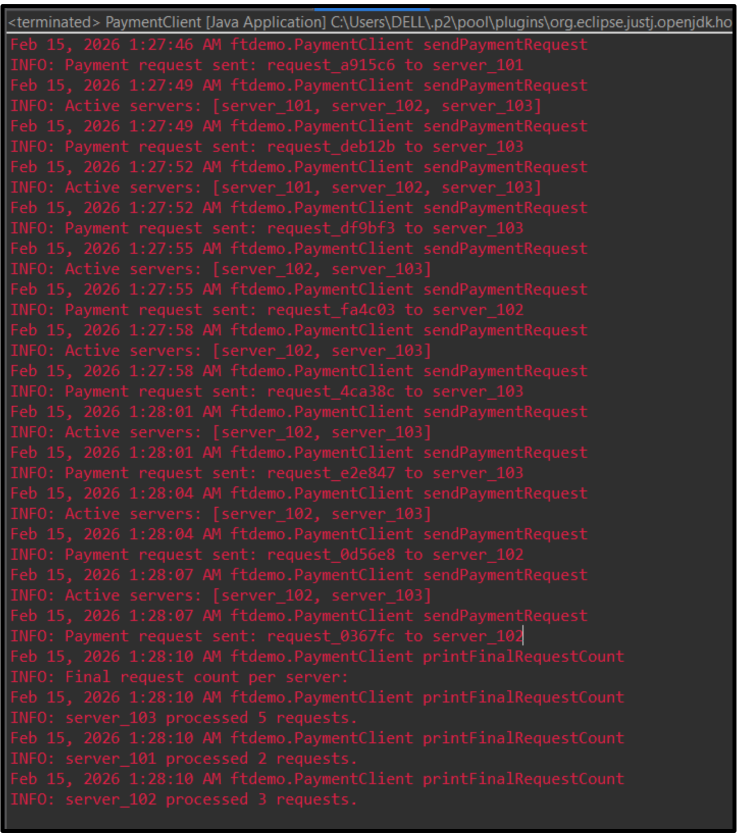
```java
        }

        // ========== DEFAULT MODE: RANDOM SERVER SELECTION ==========
        Random random = new Random();
        String selectedServer = servers.get(random.nextInt(servers.size()));
        // ===========================================================

        /*
         * ==================== LAB EXERCISE (TODO) =====================
         * The current implementation RANDOMLY selects a server.
         * Your task is to implement ROUND-ROBIN selection instead.
         *
         * 1. Comment out the RANDOM selection code above.
         * 2. Uncomment the ROUND-ROBIN selection code below.
         * 3. Run the program and observe the difference in request distribution.
         */

        /*
        // ========== ROUND-ROBIN SERVER SELECTION (TODO: Implement) ==========
        String selectedServer = servers.get(roundRobinIndex.getAndUpdate(i -> (i + 1) % servers.size()));
        // ==================================================================
        */

        // Create a unique request ID
        String requestId = "request_" + UUID.randomUUID().toString().substring(0, 6);

        // Place the request under the selected server's request queue
```

```
<terminated> PaymentClient [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-0741\jre\bin\javaw.exe  (Feb 15, 2026, 1:08:53 AM – 1:09:25 AM el
Feb 15, 2026 1:09:19 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_58f7e8 to server_103
Feb 15, 2026 1:09:22 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:09:22 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_93a7b6 to server_102
Feb 15, 2026 1:09:25 AM ftdemo.PaymentClient printFinalRequestCount
INFO: Final request count per server:
Feb 15, 2026 1:09:25 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_103 processed 7 requests.
Feb 15, 2026 1:09:25 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_102 processed 3 requests.
```

## 3. Observing Failures



```
<terminated> PaymentServer [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.oper
Feb 15, 2026 1:27:02 AM ftdemo.PaymentServer registerServer
INFO: Registered server: server_101
Feb 15, 2026 1:27:42 AM ftdemo.PaymentServer watchForRequests
INFO: Processing payment for request: request_5cb05d
Feb 15, 2026 1:27:42 AM ftdemo.PaymentServer watchForRequests
INFO: Total requests processed by server_101: 1
Feb 15, 2026 1:27:48 AM ftdemo.PaymentServer watchForRequests
INFO: Processing payment for request: request_a915c6
Feb 15, 2026 1:27:48 AM ftdemo.PaymentServer watchForRequests
INFO: Total requests processed by server_101: 2
```



```
<terminated> PaymentClient [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.ho
Feb 15, 2026 1:27:46 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_a915c6 to server_101
Feb 15, 2026 1:27:49 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:27:49 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_deb12b to server_103
Feb 15, 2026 1:27:52 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:27:52 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_df9bf3 to server_103
Feb 15, 2026 1:27:55 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_102, server_103]
Feb 15, 2026 1:27:55 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_fa4c03 to server_102
Feb 15, 2026 1:27:58 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_102, server_103]
Feb 15, 2026 1:27:58 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_4ca38c to server_103
Feb 15, 2026 1:28:01 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_102, server_103]
Feb 15, 2026 1:28:01 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_e2e847 to server_103
Feb 15, 2026 1:28:04 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_102, server_103]
Feb 15, 2026 1:28:04 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_0d56e8 to server_102
Feb 15, 2026 1:28:07 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_102, server_103]
Feb 15, 2026 1:28:07 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_0367fc to server_102
Feb 15, 2026 1:28:10 AM ftdemo.PaymentClient printFinalRequestCount
INFO: Final request count per server:
Feb 15, 2026 1:28:10 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_103 processed 5 requests.
Feb 15, 2026 1:28:10 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_101 processed 2 requests.
Feb 15, 2026 1:28:10 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_102 processed 3 requests.
```

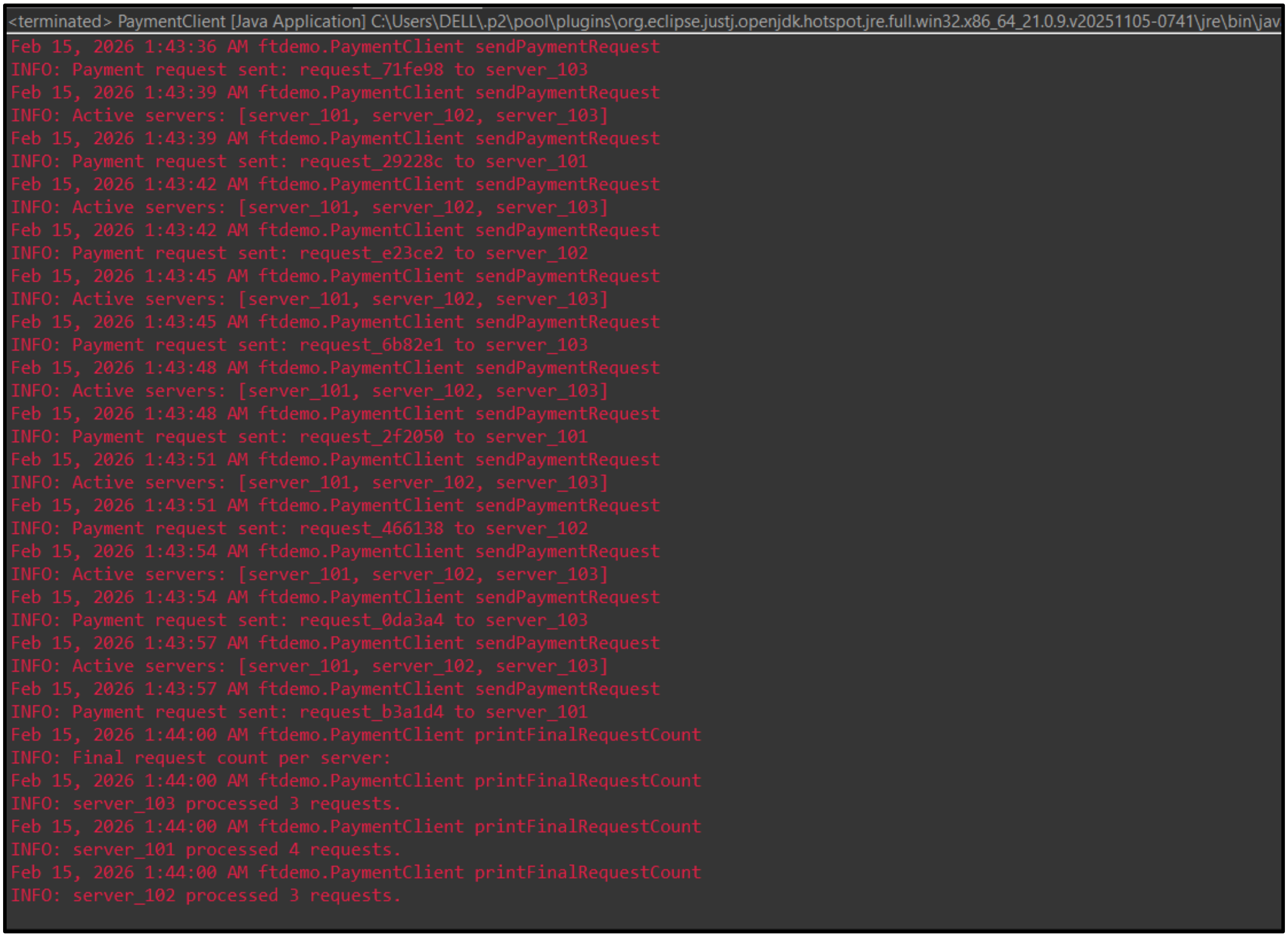
### 1. What happens when a server is killed?

When a PaymentServer is terminated, its ZooKeeper session ends.
Since the server is registered as an EPHEMERAL node under /servers, ZooKeeper automatically removes the node.
The PaymentClient no longer sees the failed server in the active server list. New requests are only sent to remaining active servers. This demonstrates fault tolerance.

### 2. How does ZooKeeper handle failure?

ZooKeeper monitors client sessions.
When a server disconnects or crashes, its session expires.
All EPHEMERAL nodes created by that session are automatically deleted.
This ensures the system remains consistent and up-to-date.

## 4. Changing the Load Balancing Strategy



```
<terminated> PaymentClient [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-0741\jre\bin\jav
Feb 15, 2026 1:43:36 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_71fe98 to server_103
Feb 15, 2026 1:43:39 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:39 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_29228c to server_101
Feb 15, 2026 1:43:42 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:42 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_e23ce2 to server_102
Feb 15, 2026 1:43:45 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:45 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_6b82e1 to server_103
Feb 15, 2026 1:43:48 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:48 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_2f2050 to server_101
Feb 15, 2026 1:43:51 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:51 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_466138 to server_102
Feb 15, 2026 1:43:54 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:54 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_0da3a4 to server_103
Feb 15, 2026 1:43:57 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Active servers: [server_101, server_102, server_103]
Feb 15, 2026 1:43:57 AM ftdemo.PaymentClient sendPaymentRequest
INFO: Payment request sent: request_b3a1d4 to server_101
Feb 15, 2026 1:44:00 AM ftdemo.PaymentClient printFinalRequestCount
INFO: Final request count per server:
Feb 15, 2026 1:44:00 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_103 processed 3 requests.
Feb 15, 2026 1:44:00 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_101 processed 4 requests.
Feb 15, 2026 1:44:00 AM ftdemo.PaymentClient printFinalRequestCount
INFO: server_102 processed 3 requests.
```

### 1. Why does round-robin ensure even distribution?

Round-robin selects servers in sequential order.
Each server receives requests in turn.
This ensures all servers handle approximately the same number of requests.

### 2. Advantages and disadvantages of Random vs Round-Robin

Random:
+ Simple implementation
+ No need to track index
- Uneven distribution possible

Round-Robin:
+ Even load distribution
+ Fair request allocation
- Slightly more logic required