

VanillaComm

Getting Started

Introduction to Databases

DataLab

CS, NTHU

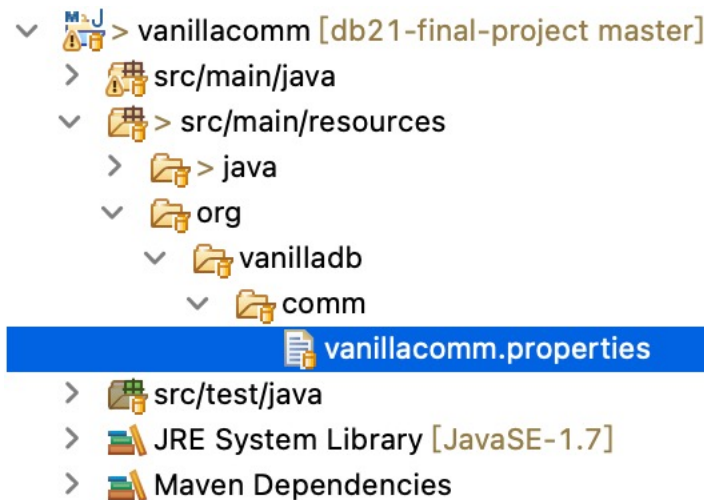
What is VanillaComm?

- You have learned about Group Communication.
- A project that aims to provide a reliable communication channel through networks.
- Two main functionalities
 - Total-ordered broadcasting
 - Total-order: a global order agreed by all participants
 - Point-to-point messaging

Let's Run a Demo!

Getting Started

1. Decides how many servers and clients you want to use in this demonstration.
2. Sets up the addresses of servers and clients:



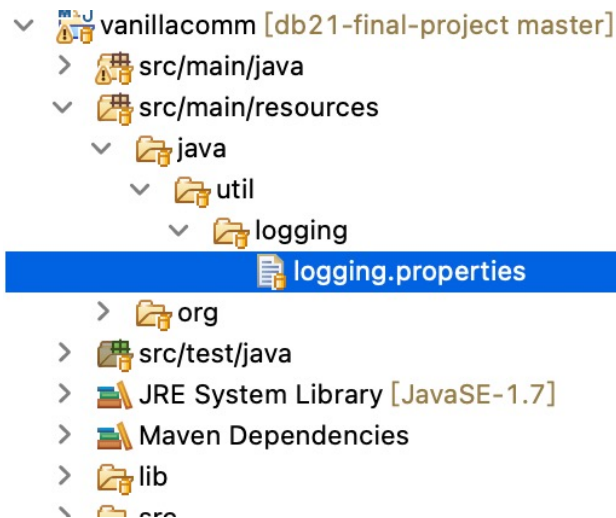
vanillacomm.properties

```
14# The views of the machine
15# A machine is represented by "ID IP PORT"
16# Each machine is split by a comma (,)
17org.vanilladb.comm.view.ProcessView.SERVER_VIEW=0 127.0.0.1 42961, 1 127.0.0.1 42962, 2 127.0.0.1 42963
18org.vanilladb.comm.view.ProcessView.CLIENT_VIEW=0 127.0.0.1 30000, 1 127.0.0.1 30001
19|
```

3 servers and 2 clients

logging.properties

- The current logs show too many messages.
 - You can change logging setting.

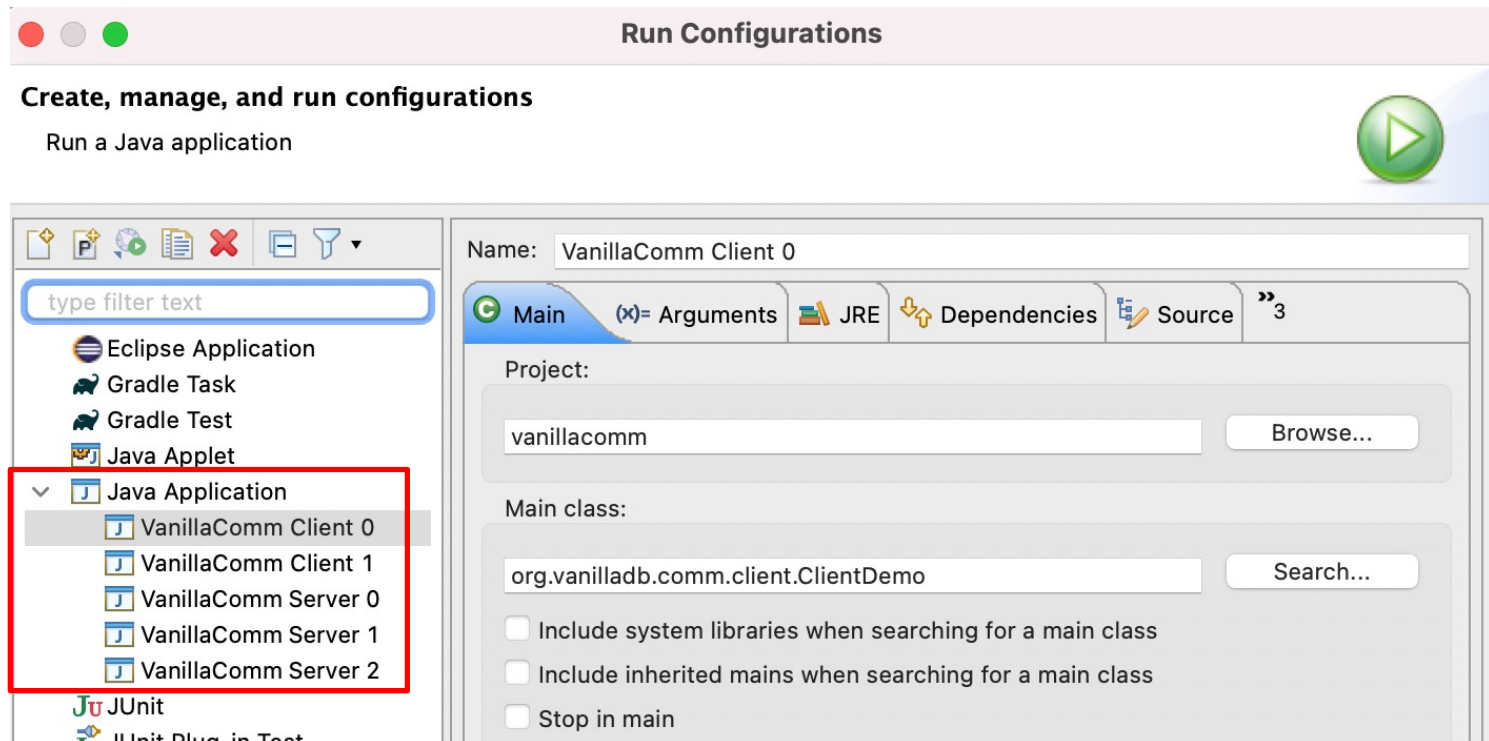


```
47
48 # Default handlers for all loggers
49 handlers=java.util.logging.ConsoleHandler
50
51 # Logging levels for handlers
52 java.util.logging.ConsoleHandler.level=ALL
53
54 # Default logging level for all loggers
55 .level=FINE
```

Change to “.level=INFO”

Getting Started

3. Creates a run configuration for each server and client.



Run Configuration - Server

- Main class: `org.vanilladb.comm.server.ServerDemo`
- Program arguments: [server id]
 - Example: `0` for server no.0
- VM arguments:

```
-Dorg.vanilladb.comm.config.file=target/classes/org/vanilladb/comm/vanillacomm.properties  
-Djava.util.logging.config.file=target/classes/java/util/logging/logging.properties
```


Run Configuration - Client

- Main class: `org.vanilladb.comm.client.ClientDemo`
- Program arguments: [client id]
 - Example: `0` for client no.0
- VM arguments:

```
-Dorg.vanilladb.comm.config.file=target/classes/org/vanilladb/comm/vanillacomm.properties  
-Djava.util.logging.config.file=target/classes/java/util/logging/logging.properties
```

Getting Started

4. Starts up the servers



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The console output is as follows:

```
VanillaComm Server 0 [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/java (2021年5月31日 下午4:09:28)  
5月 31, 2021 4:09:34 下午 org.vanilladb.comm.protocols.zabacceptance.ZabAcceptanceSession handleProcessConnected  
FINE: Received ProcessConnected  
5月 31, 2021 4:09:34 下午 org.vanilladb.comm.protocols.zabproposal.ZabProposalSession handleProcessConnected  
FINE: Received ProcessConnected  
5月 31, 2021 4:09:34 下午 org.vanilladb.comm.protocols.totalorderappl.TotalOrderApplicationSession handleProcessCo  
FINE: Received ProcessConnected  
5月 31, 2021 4:09:34 下午 org.vanilladb.comm.server.VanillaCommServer onAllProcessesReady  
INFO: All processes are ready.  
5月 31, 2021 4:09:34 下午 org.vanilladb.comm.server.ServerDemo onServerReady  
INFO: The server is ready!
```

The last two lines of the log are enclosed in a red rectangular box.

Getting Started

5. Once all the servers are ready, starts the clients
 - The clients will start sending requests to servers.
 - Then, the servers will run a protocol to order these requests.

Results

- The server will show the total-ordered broadcast messages
 - You should see all the servers show the messages in the same order

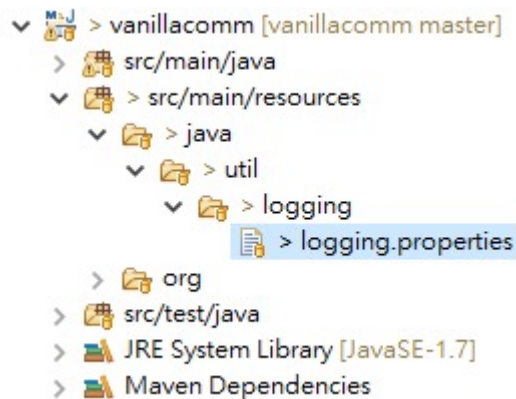
```
5月 31, 2021 4:13:57 下午 org.vanilladb.comm.server.VanillaCommServer onAllProcessesReady  
INFO: All processes are ready.
```

```
5月 31, 2021 4:13:57 下午 org.vanilladb.comm.server.ServerDemo onServerReady  
INFO: The server is ready!
```

```
Received a total order message: Request #0 from client 0, serial number: 1  
Received a total order message: Request #1 from client 0, serial number: 2  
Received a total order message: Request #2 from client 0, serial number: 3  
Received a total order message: Request #3 from client 0, serial number: 4  
Received a total order message: Request #4 from client 0, serial number: 5  
Received a total order message: Request #5 from client 0, serial number: 6  
Received a total order message: Request #6 from client 0, serial number: 7  
Received a total order message: Request #7 from client 0, serial number: 8  
Received a total order message: Request #100000 from client 1, serial number: 9  
Received a total order message: Request #8 from client 0, serial number: 10  
Received a total order message: Request #100001 from client 1, serial number: 11  
Received a total order message: Request #9 from client 0, serial number: 12  
Received a total order message: Request #100002 from client 1, serial number: 13  
Received a total order message: Request #10 from client 0, serial number: 14  
Received a total order message: Request #100003 from client 1, serial number: 15  
Received a total order message: Request #11 from client 0, serial number: 16  
Received a total order message: Request #100004 from client 1, serial number: 17  
Received a total order message: Request #12 from client 0, serial number: 18  
Received a total order message: Request #100005 from client 1, serial number: 19
```

More Logs!!!

- The current logs only show necessary messages.
 - However, you can change logging setting to see more.



```
48# Default handlers for all loggers
49handlers=java.util.logging.ConsoleHandler
50
51# Logging levels for handlers
52java.util.logging.ConsoleHandler.level=ALL
53
54# Default logging level for all loggers
55.level=INFO
56
```

Change to “.level=FINE”

VanillaComm Server 0 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (2020年6月4日 下午6:32:51)

六月 04, 2020 6:32:51 下午 org.vanilladb.comm.server.ServerDemo main
資訊: Initializing the server...
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.server.VanillaCommServer run
資訊: Starts the network service
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.totalorderappl.TotalOrderApplicationSession handleChannelInit
詳細: Received ChannelInit
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.totalorderappl.TotalOrderApplicationSession handleChannelInit
資訊: Socket registration request sent.
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.p2pappl.P2pApplicationSession handleChannelInit
詳細: Received ChannelInit
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.tcpfd.TcpFailureDetectionSession handleProcessListInit
詳細: Received ProcessListInit from Channel P2P Channel
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.tcpfd.TcpFailureDetectionSession handleRegisterSocket
詳細: Received RegisterSocketEvent
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.zabproposal.ZabProposalSession handleProcessListInit
詳細: Received ProcessListInit
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.zabacceptance.ZabAcceptanceSession handleProcessListInit
詳細: Received ProcessListInit
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.zabelection.ZabElectionSession handleProcessListInit
詳細: Received ProcessListInit
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.beb.BestEffortBroadcastSession handleProcessListInit
詳細: Received ProcessListInit
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.tcpfd.TcpFailureDetectionSession handleProcessListInit
詳細: Received ProcessListInit from Channel Zab Channel
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.tcpfd.TcpFailureDetectionSession handleRegisterSocket
詳細: Received RegisterSocketEvent
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.tcpfd.TcpFailureDetectionSession handleRegisterSocket
詳細: Sending heartbeats to all other nodes
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.totalorderappl.TotalOrderApplicationSession handleRegisterSocketEvent
詳細: Received RegisterSocket
六月 04, 2020 6:32:52 下午 org.vanilladb.comm.protocols.totalorderappl.TotalOrderApplicationSession handleRegisterSocketEvent
資訊: Socket registration completed. (/127.0.0.1:42961)

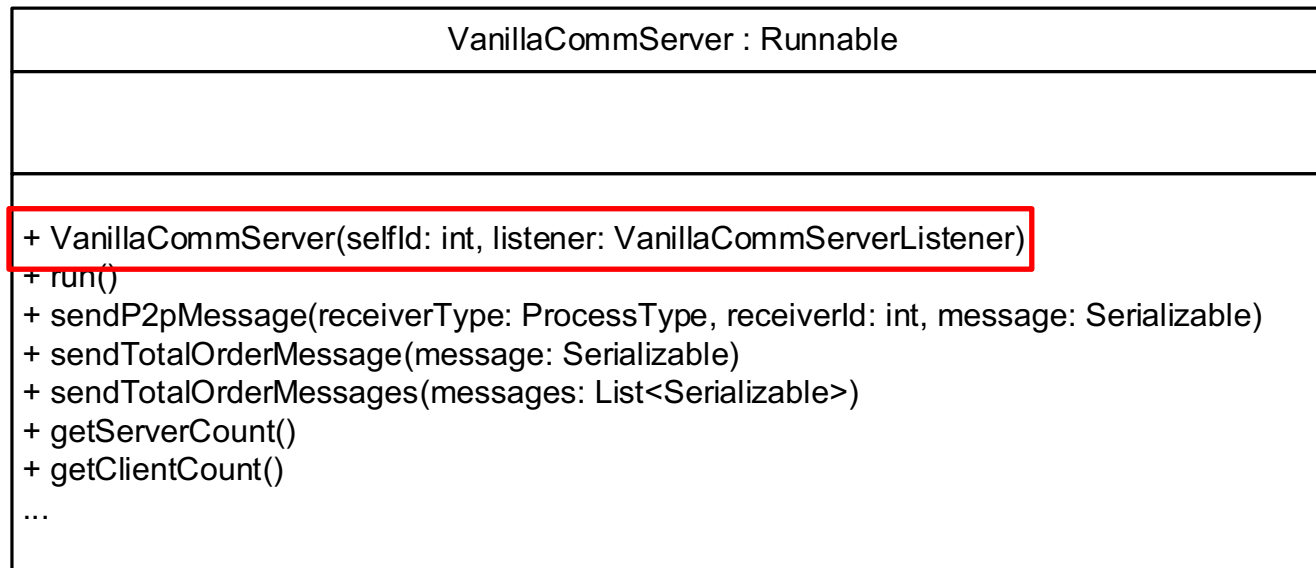
Let's See How To Use APIs

Main Components

- Four main components (you need to know)
 - VanillaCommServer
 - VanillaCommServerListener
 - VanillaCommClient
 - VanillaCommClientListener

VanillaCommServer

- Need to provide the server id and a listener for messages during construction



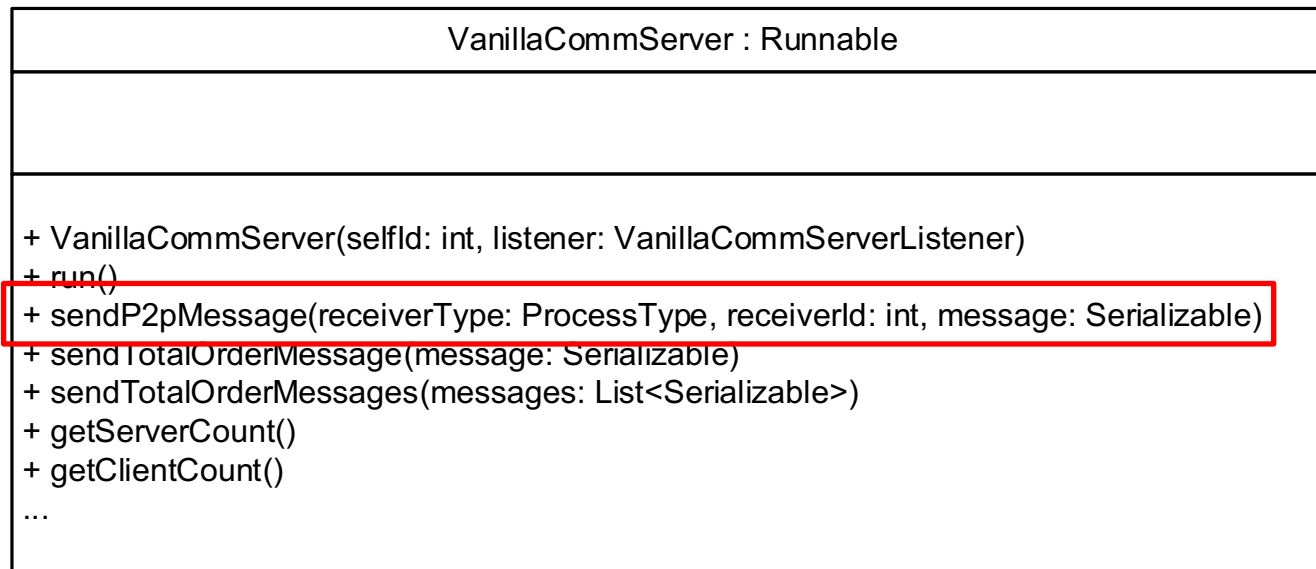
VanillaCommServer

- Need to be run in a dedicated thread
 - `new Thread(new VanillaCommServer(...))`

VanillaCommServer Runnable
<pre>+ VanillaCommServer(selfId: int, listener: VanillaCommServerListener) + run() + sendP2pMessage(receiverType: ProcessType, receiverId: int, message: Serializable) + sendTotalOrderMessage(message: Serializable) + sendTotalOrderMessages(messages: List<Serializable>) + getServerCount() + getClientCount() ...</pre>

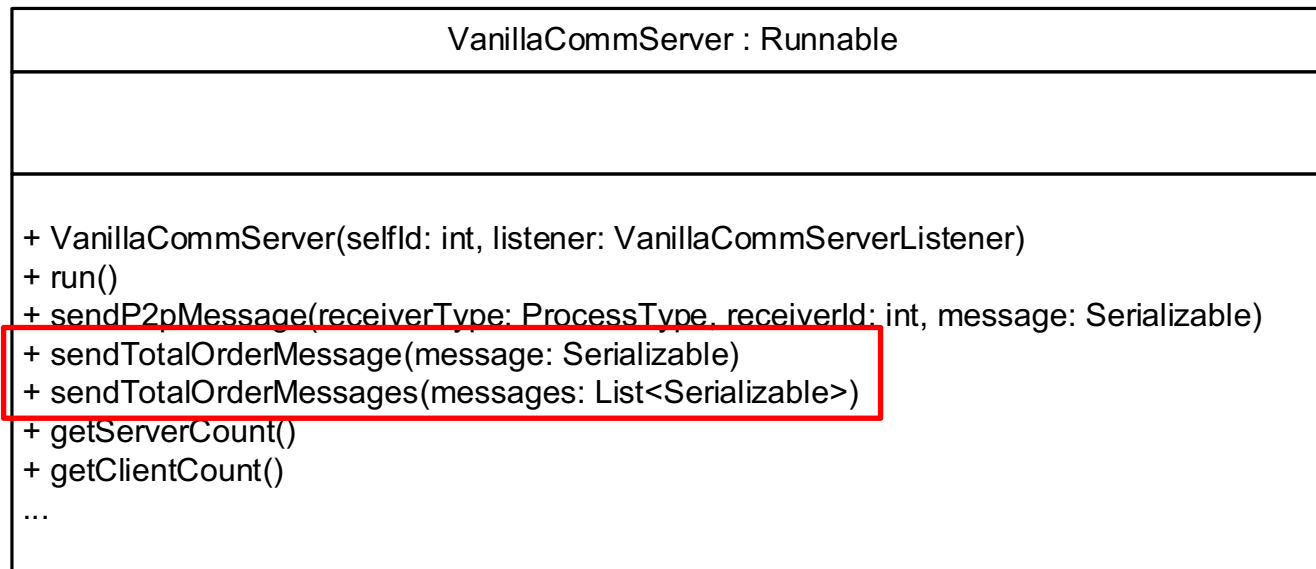
VanillaCommServer

- A P2p Message: a message from a process to another process
 - Need to specify if it is a server or a client



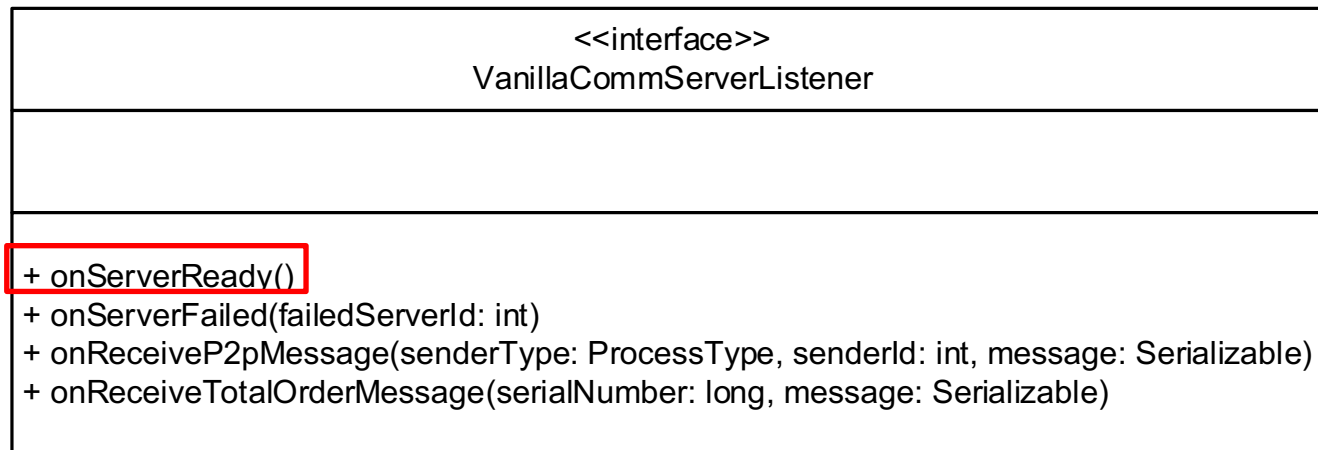
VanillaCommServer

- A Total-order Message: a broadcast message that will be sent to **all servers** in the same order



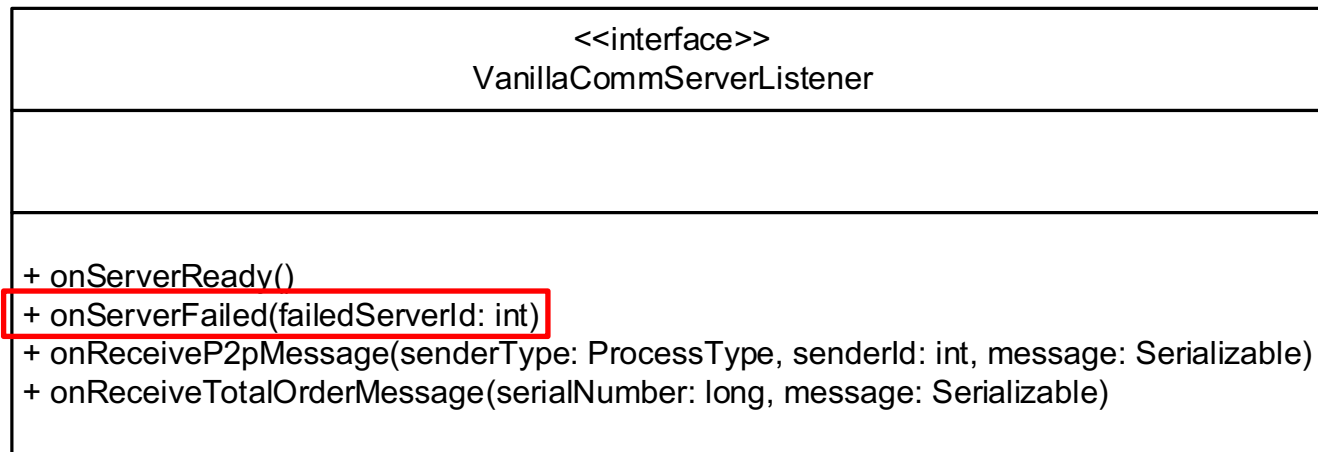
VanillaCommServerListener

- Be notified when the server is good to go



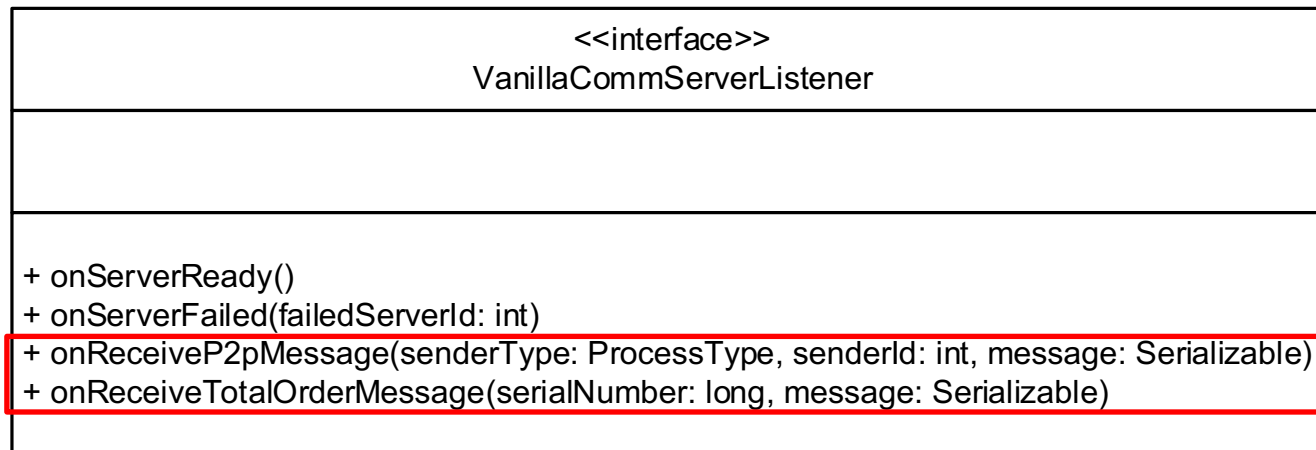
VanillaCommServerListener

- Be notified when one of servers is failed



VanillaCommServerListener

- Be notified when the server received a message



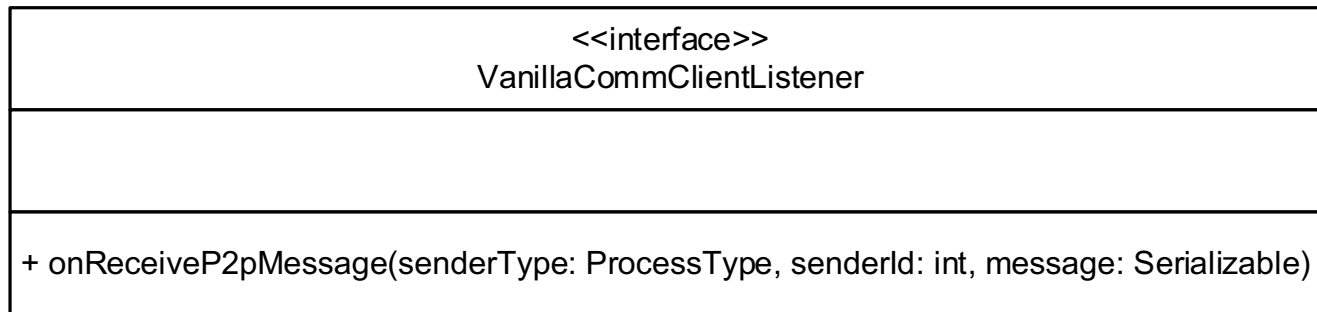
VanillaCommClient

- Similar to VanillaCommServer without the ability to send total-ordered messages
 - Check demo's code to see how to total-order a client's request.

VanillaCommClient : Runnable
<pre>+ VanillaCommClient(selfId: int, listener: VanillaCommClientListener) + run() + sendP2pMessage(receiverType: ProcessType, receiverId: int, message: Serializable) + getServerCount() + getClientCount() ...</pre>

VanillaCommClientListener

- Similar to VanillaCommServerListener without the ability to receive total-ordered messages



The Example Code

- Don't forget to check the example code!
 - `org.vanilladb.comm.client.ClientDemo`
 - `org.vanilladb.comm.server.ServerDemo`