

Compilation Instructions

Step 1: Navigate to Project Directory

```
cd /path/to/your/project/directory
```

Step 2: Compile the Java Files

```
javac PubSubServer.java
```

```
javac PubSubClient.java
```

Note: If compilation is successful, you should see .class files created in the same directory.

Running the Application

Step 1: Start the Server

Open a terminal/command prompt and run:

```
java PubSubServer <PORT>
```

Example:

```
java PubSubServer 5000
```

Expected Output:

Server starting on 127.0.0.1:5000

Server listening on port 5000...

Waiting for client connections...

Press Ctrl+C to shutdown server

Step 2: Start Clients

Open **separate** terminal/command prompt windows for each client and run:

```
java PubSubClient <SERVER_IP> <SERVER_PORT> <CLIENT_TYPE> <TOPIC>
```

Parameters:

- SERVER_IP: IP address of the server (use 127.0.0.1 for local testing)
- SERVER_PORT: Port number the server is listening on
- CLIENT_TYPE: Either PUBLISHER or SUBSCRIBER
- TOPIC: The topic name for publishing/subscribing

Example Usage Scenarios

Scenario 1: Single Topic (SPORTS)

Terminal 1 - Server

```
java PubSubServer 5000
```

Terminal 2 - Publisher for SPORTS

```
java PubSubClient 127.0.0.1 5000 PUBLISHER SPORTS
```

Terminal 3 - Subscriber for SPORTS

```
java PubSubClient 127.0.0.1 5000 SUBSCRIBER SPORTS
```

Terminal 4 - Another Subscriber for SPORTS

```
java PubSubClient 127.0.0.1 5000 SUBSCRIBER SPORTS
```

Scenario 2: Multiple Topics

Terminal 1 - Server

```
java PubSubServer 5000
```

Terminal 2 - Publisher for TECH topic

```
java PubSubClient 127.0.0.1 5000 PUBLISHER TECH
```

Terminal 3 - Subscriber for TECH topic

```
java PubSubClient 127.0.0.1 5000 SUBSCRIBER TECH
```

Terminal 4 - Publisher for SPORTS topic

```
java PubSubClient 127.0.0.1 5000 PUBLISHER SPORTS
```

Terminal 5 - Subscriber for SPORTS topic

```
java PubSubClient 127.0.0.1 5000 SUBSCRIBER SPORTS
```

How It Works

1. **Server:** Acts as a message broker, managing publishers and subscribers by topic
2. **Publishers:** Send messages to specific topics
3. **Subscribers:** Receive messages from publishers on topics they're subscribed to
4. **Topic Filtering:** Messages are only delivered to subscribers of the same topic

Testing Instructions

Test 1: Basic Client-Server Communication

1. Start the server
2. Start one publisher and one subscriber on the same topic
3. Type messages in the publisher terminal
4. Verify messages appear in the subscriber terminal

Test 2: Multiple Subscribers

1. Start the server
2. Start one publisher and multiple subscribers on the same topic
3. Type messages in the publisher terminal
4. Verify messages appear in all subscriber terminals

Test 3: Topic Filtering

1. Start the server
2. Start publishers and subscribers on different topics (e.g., SPORTS, TECH)
3. Type messages in each publisher terminal
4. Verify messages only appear in subscribers of the same topic

Commands During Execution

For Publishers:

- Type any message and press Enter to publish
- Type terminate to disconnect and exit

For Subscribers:

- Messages from publishers will appear automatically
- Type terminate to disconnect and exit

For Server:

- Press Ctrl+C to shutdown the server
- Server displays connection status and message routing information

Expected Behavior

Server Output:

Server starting on 127.0.0.1:5000

Server listening on port 5000...

Waiting for client connections...

Publisher connected from /127.0.0.1:12345 for topic 'SPORTS'

Subscriber connected from /127.0.0.1:12346 for topic 'SPORTS'

=== Current Topic Status ===

Publishers:

Topic 'SPORTS': 1 publisher(s)

Subscribers:

Topic 'SPORTS': 1 subscriber(s)

=====

Received from PUBLISHER /127.0.0.1:12345 on topic 'SPORTS': Hello World!

Message delivered to 1 subscriber(s) on topic 'SPORTS'

Publisher Output:

Connecting to server 127.0.0.1:5000 as PUBLISHER for topic 'SPORTS'...

Connected to server!

Ready to publish messages on topic 'SPORTS'

Type your messages (type 'terminate' to quit):

[SPORTS] > Hello World!

[SPORTS] > terminate

Terminating connection...

Client terminated.

Subscriber Output:

Connecting to server 127.0.0.1:5000 as SUBSCRIBER for topic 'SPORTS'...

Connected to server!

Listening for messages on topic 'SPORTS'...

Type your messages (type 'terminate' to quit):

>

[SPORTS] Publisher /127.0.0.1:12345: Hello World!

> terminate

Terminating connection...

Client terminated.

Troubleshooting

Common Issues:

1. **"Could not connect to server"**
 - Make sure the server is running
 - Check if the IP address and port are correct
 - Verify firewall settings
2. **"PORT must be a valid integer"**
 - Ensure you're using a valid port number (1024-65535)
3. **"CLIENT_TYPE must be either 'PUBLISHER' or 'SUBSCRIBER'"**
 - Check spelling and case sensitivity

Tips:

- Use different terminal windows for each component
- Keep the server running while testing clients
- Use 127.0.0.1 for local testing
- Make sure port numbers don't conflict with other applications