

Enviar un mensaje entre un cliente y un servidor con corrutinas y Executors en Kotlin de la forma más simple

Servidor

[crayon-6a9d67b206c7f470536013/]

1. **import kotlinx.coroutines.***: Importa las clases y funciones relacionadas con corrutinas en Kotlin. Esto incluye las funciones `runBlocking`, `launch`, y el contexto `Dispatchers`.
2. **import java.net.ServerSocket**: Importa la clase `ServerSocket` para manejar las conexiones del servidor.
3. **fun main() = runBlocking {**: Define la función principal del programa. `runBlocking` es una función de las corrutinas que crea un bloque de corrutinas y espera a que todas las corrutinas lanzadas dentro de ese bloque se completen antes de continuar.
4. **val serverSocket = ServerSocket(12345)**: Crea un objeto `ServerSocket` que escucha en el puerto 12345.
5. **println("Servidor esperando conexiones en el puerto 12345...")**: Imprime un mensaje en la consola indicando que el servidor está esperando conexiones en el puerto 12345.
6. **while (true) {**: Inicia un bucle infinito que aceptará conexiones de clientes continuamente.
7. **val clienteSocket = serverSocket.accept()**: Espera y acepta una conexión de un cliente, devolviendo un objeto `Socket` para la comunicación con ese cliente.
8. **println("Cliente conectado desde:**

`${clienteSocket.inetAddress.hostAddress}`): Imprime la dirección IP del cliente conectado.

9. **`launch(Dispatchers.IO)`** {: Inicia una nueva corutina en el contexto de entrada/salida (IO) para manejar la comunicación con el cliente. Esto permite operaciones de entrada/salida sin bloquear el hilo principal.

10. **`val palabraRecibida = clienteSocket.getInputStream().bufferedReader().readLine()`**: Lee una línea desde el flujo de entrada del cliente (el mensaje enviado por el cliente).

11. **`println("Palabra recibida: $palabraRecibida")`**: Imprime la palabra recibida en la consola.

12. **`clienteSocket.close()`**: Cierra el socket del cliente después de manejar la comunicación.

Cliente

[crayon-6a9d67b206c86768581093/]