Roboter Beispiel

```java
public void RobotEndpointExample() throws InstantiationException, IllegalAccessException, IOException {
        RobotEndpoint robotEnd = new RobotEndpoint();
        robotEnd.open(9999);

        robotEnd.addOnConnectionIncommingListener(new IConnectionIncommingEventHandler() {

            @Override
            public void OnNewConnection(Socket sock) {
                System.out.println("Controller has connected: " + sock.getRemoteSocketAddress().toString());

            }
        });

        robotEnd.addOnDisconnectListener(new IDisconnectedEventListener() {

            @Override
            public void OnDisconnect() {
                System.out.println("Controller has disconnected");
            }
        });

        robotEnd.addCommandListener(new ICommandReceiver() {

            @Override
            public void commandReceived(CommandBase cmd) {
                System.out.println("Command from controller received: " + cmd.getType().toString());
            }
        });

        IRobot robot = robotEnd.getRobotInterface();

        //Tell the controler that the reset was completed.
        robot.sendReturnReset();
    }
```

Controller Beispiel

```java
public void ControlerEndpointExample() throws InstantiationException, IllegalAccessException, IOException {
        ControlerEndpoint controlerEnd = new ControlerEndpoint();
        controlerEnd.connect(InetAddress.getByName("192.168.0.2"), 9999);;

        controlerEnd.addOnDisconnectListener(new IDisconnectedEventListener() {

            @Override
            public void OnDisconnect() {
                System.out.println("Controller has disconnected");
            }
        });

        controlerEnd.addCommandListener(new ICommandReceiver() {

            @Override
            public void commandReceived(CommandBase cmd) {
                System.out.println("Command from controller received: " + cmd.getType().toString());
            }
        });


        IControler controler = controlerEnd.getControlerInterface();

        //Reset the robot
        controler.sendReset();

        //Request the robot status
        controler.sendGetStatus();

        //Move the robot for 2 seconds(360° on both motors total)
        controler.sendMoveMotor(180, 180, 360, 360);
    }
```