**Basic App with Using Akka Framework:**

AKKA framework ü kullanılarak basit bir uygulama geliştirilecek. 2 aktörün kullanıldığı küçük bir sistem yapılacak. Bir aktör diğer aktöre "Hi from Actor1" yazacak ve diğer aktör de "Hi from Actor2" cevabını verecek.

(2 portta 2 Java app çalışacak, işlem portlar arası olacak)

***FirstActor.java:***

package org.example;  
  
import akka.actor.AbstractActor;  
import akka.actor.ActorSystem;  
import akka.actor.Props;  
import com.typesafe.config.Config;  
import com.typesafe.config.ConfigFactory;  
  
public class FirstActor extends AbstractActor {  
  
  
 @Override  
 public Receive createReceive() {  
 return receiveBuilder()  
 .matchEquals("start", s -> {  
 getContext().actorSelection("akka://Actor2System@localhost:25521/user/actor2")  
 .tell("hello from actor 1", getSelf());  
 })  
 .matchEquals("hello from actor 2", s -> {  
 System.out.println("actor 1 received: " + s);  
 })  
 .build();  
 }  
}  
  
class FirstActorMain{  
 public static void main(String[] args) {  
  
 Config config = ConfigFactory.load("firstActor.conf");  
 ActorSystem actorSystem = ActorSystem.create("Actor1System", config);  
 actorSystem.actorOf(Props.create(FirstActor.class), "actor1");  
 actorSystem.actorSelection("/user/actor1").tell("start", ActorSystem.create().deadLetters());  
  
 }  
}

***firstActor.conf:***

akka {  
 actor{  
 provider = remote  
 }  
 remote {  
 artery{  
 enabled = on  
 transport = tcp  
 canonical.hostname = "localhost"  
 canonical.port = 25520  
 }  
 }  
}

***SecondActor.java:***

package org.example;  
  
import akka.actor.AbstractActor;  
import akka.actor.ActorSystem;  
import akka.actor.Props;  
import com.typesafe.config.Config;  
import com.typesafe.config.ConfigFactory;  
  
public class SecondActor extends AbstractActor {  
  
 @Override  
 public Receive createReceive() {  
 return receiveBuilder()  
 .matchEquals("hello from actor 1", s -> {  
 System.out.println("Actor 2 received: " + s);  
 getSender().tell("hello from actor 2", getSelf());  
 })  
 .build();  
 }  
}  
  
class SecondActorMain{  
 public static void main(String[] args) {  
 Config config = ConfigFactory.load("secondActor.conf");  
 ActorSystem actorSystem = ActorSystem.create("Actor2System", config);  
 actorSystem.actorOf(Props.create(SecondActor.class), "actor2");  
 }  
}

***secondActor.conf:***

akka {  
 actor{  
 provider = remote  
 }  
 remote {  
 artery{  
 enabled = on  
 transport = tcp  
 canonical.hostname = "localhost"  
 canonical.port = 25521  
 }  
 }  
}

***OUTPUT:***

metin, multimedya yazılımı, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu