

Sistema Banatuak **RMI laborategia**

Mikel Larrea

KAT Saila, UPV/EHU



Urratsak

- Interfazea
- Inplementazioa
- Zerbitzaria
- Bezeroa
- Aplikazioa martxan jartzeko:
 - Java-ko konpiladorea
 - RMI-ko konpiladorea
 - RMI-ko izen-zerbitzaria abiatu
 - Zerbitzaria abiatu
 - Bezeroa abiatu

Interfazea

- Urruneko atzipena izango duten objektuen *definizioa*:
 - metodoak, beraien parametroekin
- Ez da metodo hauen inplementaziorik ematen!
- java.rmi.Remote interfazea hedatzea beharrezkoa da
- Urruneko metodo guztiek java.rmi.RemoteException jaurti dezakete

Interfazea (Hello.java)

```
package hello;

import java.rmi.Remote;
import java.rmi.RemoteException;
import java.util.Date;

public interface Hello extends Remote {
    String sayHello() throws RemoteException;
    Date getDate() throws RemoteException;
}
```

Inplementazioa

- Java-ko klase arrunt bat da
- java.rmi.server.UnicastRemoteObject klasea hedatzen du, eta lehenengo urratsean definitutako interfazea implementatzen du
 - Interfazearen urruneko metodo guztiak implementatzen ditu
- Klase honen metodo eraikitzailean UnicastRemoteObject klasearen metodo eraikitzailea deitzen da: *super()*

Inplementazioa (HelloImpl.java)

```
package hello;

import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.RMISecurityManager;
import java.rmi.server.UnicastRemoteObject;
import java.util.Date;

public class HelloImpl extends UnicastRemoteObject
    implements Hello {

    public HelloImpl() throws RemoteException {
        super();
    }

    public String sayHello() {
        return "Kaixo Mikel!";
    }

    public Date getDate() {
        return new Date();
    }

    // ...
}
```

Zerbitzaria

- Segurtasun kudeatzaile bat abiatzen du
- Urrunetik atzigarri den objektua sortzen du
 - Inplementazio klasearen instantzia bat da
- Naming.rebind bidez urruneko atzipena izango duen objektua RMI-ko izen-zerbitzarian alta ematen du (izen ezagun bat emanaz)
- Zerbitzaria klase independente baten barruan inplementatu daiteke, edota inplementazio klasearen barruan (main metodoan)

Zerbitzaria (HelloImpl.java)

```
// ...

public static void main(String args[]) {

    // Create and install a security manager
    System.setSecurityManager(new RMISecurityManager());

    try {
        Hello obj = new HelloImpl();
        // Bind this object instance to the name "HelloServer"
        Naming.rebind("HelloServer", obj);
        System.out.println("HelloServer bound in registry");
    } catch (Exception e) {
        System.out.println("HelloImpl exception: " +
            e.getMessage());
        e.printStackTrace();
    }
}
```


Bezeroa

- Segurtasun kudeatzaile bat abiatzen du
- Bezeroak urruneko objektuarekin kontaktatzeko, lehendabizi erreferentzia bat lortu behar du, izen-zerbitzariari eskaera eginez
 - Naming.lookup bidez urruneko objektuaren erreferentzia lortzen du izen-zerbitzariarengandik (eskaeran aurreko izen ezaguna bera pasa behar da)
- Erreferentzia lortu ondoren, (urruneko) objektuaren metodoak deitzen ditu, emaitzak jasoz eta pantailaratuz

Bezerao (HelloClient.java)

```
package hello;

import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.RMISecurityManager;
import java.util.Date;

public class HelloClient {

    public static void main(String args[]) {

        if (args.length != 1) {
            System.out.println(("Usage: java HelloClient host"));
            System.exit(1);
        }

        System.setSecurityManager(new RMISecurityManager());
        // ...
    }
}
```

Bezeroa (HelloClient.java)

```
// ...
try {
    Hello obj = (Hello)Naming.lookup("rmi://" + args[0] +
                                     "/HelloServer");

    String str = obj.sayHello();
    System.out.println(str);
    // invoke method on server object and locally
    // (to compare dates)
    Date d_remote = obj.getDate();
    Date d_local = new Date();
    System.out.println("Date on server is " + d_remote);
    System.out.println("Date on client is " + d_local);
} catch (Exception e) {
    System.out.println("HelloClient exception: " +
                       e.getMessage());
    e.printStackTrace();
}
}
```

Aplikazioa martxan jartzeko

- Java-ko konpiladorea:
 - `javac hello/*.java`
- RMI-ko konpiladorea (implementazio klasearen aurka!):
 - `rmic hello.HelloImpl`
- RMI-ko izen-zerbitzaria abiatu (Java-rekin dator):
 - `rmiregistry`
- Zerbitzari programa abiatu:
 - `java -Djava.security.policy=hello/policy hello.HelloImpl`
- Bezero programa abiatu:
 - `java -Djava.security.policy=hello/policy hello.HelloClient host`