

30 Novembre 2021



Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo

Esercitazione 7

Java RMI e Riferimenti Remoti

Obiettivi esercitazione

Hard skills

1

Invocazione di procedure remote
mediante JAVA RMI

2

Utilizzo di TAG di ricerca e
registrazione

Soft skills

1

Collaborazione per raggiungere un fine comune

2

Organizzazione del proprio lavoro in team

RegistryRemotoTagClient

```
public interface RegistryRemotoTagClient
    extends RegistryRemotoClient {

    public String[] cercaTag(Tag tag) throws RemoteException;
}
```

RegistryRemotoTagServer

```
public interface RegistryRemotoTagServer
    extends RegistryRemotoServer, RegistryRemotoTagClient {

    public boolean associaTag(String nome_logico_server, Tag tag)
        throws RemoteException;
}
```

RegistryRemotoTagImpl

```
public static void main(String[] args) {

    int registryRemotoPort = 1099;
    String registryRemotoHost = "localhost";
    String registryRemotoName = "RegistryRemotoTag";

    // Controllo dei parametri della riga di comando
    if (args.length != 0 && args.length != 1) {...}
    if (args.length == 1) {...}

    // Impostazione del SecurityManager
    if (System.getSecurityManager() == null)
        System.setSecurityManager(new RMISecurityManager());

    // Registrazione del servizio RMI
    String completeName = "/" + registryRemotoHost + ":" + registryRemotoPort
        + "/" + registryRemotoName;
    try {
        RegistryRemotoTagImpl serverRMI = new RegistryRemotoTagImpl();
        Naming.rebind(completeName, serverRMI);
        System.out.println("Server RMI: Servizio \" + registryRemotoName
            + "\" registrato");
    } catch (Exception e) {
        System.err.println("Server RMI \" + registryRemotoName + "\": "
            + e.getMessage());
        e.printStackTrace();
        System.exit(status: 1);
    }
}
```

Tag Enum Class

```
public enum Tag {  
    ECHO( textTag: "ECHO", val: 1), CONGRESSO( textTag: "CONGRESSO", val: 2);  
  
    private final String textTag;  
    private final int val;  
  
    Tag(String textTag, int val) {  
        this.textTag = textTag;  
        this.val = val;  
    }  
  
    public static boolean check(String textTag){  
        for (Tag itemTag: Tag.values()) {  
            if(itemTag.textTag.equals(textTag)) return true;  
        }  
        return false;  
    }  
}
```

cercaTag (Tag tag)

```
public synchronized String[] cercaTag(Tag tag) throws RemoteException {
    if(tag==null ) throw new RemoteException("il tag ha valore null");

    int count = 0;
    for(int i=0; i<this.tableSize; i++) {
        if(this.table[i][2] != null && this.table[i][2].equals(tag)) {
            count++;
        }
    }
    String[] nomiServizi = new String[count];
    count = 0;
    for(int i=0; i<this.tableSize; i++) {
        if(this.table[i][2] != null && this.table[i][2].equals(tag)) {
            nomiServizi[count++] = (String) this.table[i][0];
        }
    }

    return nomiServizi;
}
```

associaTag (String nome, Tag tag)

```
public synchronized boolean associaTag(String nome_logico_server, Tag tag)
    throws RemoteException {
    if(tag == null || nome_logico_server == null)
        throw new RemoteException("il tag o il nome logico sono null");

    for (int i=0; i<this.tableSize; i++) {
        if (this.table[i][0] != null && this.table[i][0].equals(nome_logico_server)) {
            this.table[i][2] = tag;
            System.out.println("Cambiato il tag del servizio : "+nome_logico_server);
            return true;
        }
    }
    return false;
}
```

ServerCongresso

```
public interface ServerCongresso extends Remote {  
    int registrazione(int giorno, String sessione, String speaker)  
        throws RemoteException;  
    Programma programma(int giorno) throws RemoteException;  
}
```

ServerCongressoImpl

```
try {  
    RegistryRemotoTagServer registryRemoto = (RegistryRemotoTagServer) Naming  
        .lookup(completeRemoteRegistryName);  
    ServerCongressoImpl serverRMI = new ServerCongressoImpl();  
    registryRemoto.aggiungi(serviceName, serverRMI);  
    System.out.println("Server RMI: Servizio \" + serviceName  
        + "\" registrato");  
    registryRemoto.associaTag(serviceName, Tag.CONGRESSO);  
}
```

ServerEcho

```
public interface ServerEcho extends Remote {  
    String message(String msg)  
        throws RemoteException;  
}
```

ServerEchoImpl

```
RegistryRemotoTagServer registryRemoto = (RegistryRemotoTagServer) Naming  
    .lookup(completeRemoteRegistryName);  
ServerEchoImpl serverRMI = new ServerEchoImpl();  
registryRemoto.aggiungi(serviceName, serverRMI);  
System.out.println("Server RMI: Servizio \" + serviceName  
    + "\" registrato");  
registryRemoto.associaTag(serviceName, Tag.ECHO);
```


Client

A

```
String completeRemoteRegistryName = "//" + registryRemotoHost + ":"  
    + registryRemotoPort + "/" + registryRemotoName;  
RegistryRemotoTagClient registryRemoto =  
    (RegistryRemotoTagClient) Naming.lookup(completeRemoteRegistryName);
```

B

```
String nomeTag;  
do {  
    System.out.print("Inserisci nome del tag: ");  
    nomeTag = stdin.readLine().trim();  
    if (nomeTag == null) {  
        System.err.println("Nessun tag del servizio inserito.");  
    }  
} while (!Tag.check(nomeTag));
```

C

```
String[] serviceList = null;  
Tag currentTag = null;  
if (nomeTag.equals("ECHO")) {  
    serviceList = registryRemoto.cercaTag(Tag.ECHO);  
    currentTag = Tag.ECHO;  
} else if (nomeTag.equals("CONGRESSO")) {  
    serviceList = registryRemoto.cercaTag(Tag.CONGRESSO);  
    currentTag = Tag.CONGRESSO;  
}  
  
if (serviceList.length == 0 || serviceList == null) {...}  
else {  
    while (serviceName == null) {  
        for (int count = 0; count < serviceList.length; count++) {  
            System.out.println(count + ". " + serviceList[count]);  
        }  
        System.out.print("Inserire indice servizio a cui collegarsi: ");  
        int sel = -1;  
        try {  
            sel = Integer.parseInt(stdin.readLine().trim());  
        }  
        catch (NumberFormatException e) {...}  
        if (sel < 0 || sel > (serviceList.length - 1)) {...}  
  
        serviceName = serviceList[sel];  
    }  
    try {  
        if (currentTag == Tag.CONGRESSO) {  
            serverRMICongresso = (ServerCongresso) registryRemoto.cerca(serviceName);  
        } else if (currentTag == Tag.ECHO) {  
            serverRMIEcho = (ServerEcho) registryRemoto.cerca(serviceName);  
        }  
    }  
}
```

7

Gestione progetto

Coding



git



Team



30 Novembre 2021



Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo

Grazie !