

Rozproszone systemy internetowe

JAVA Remote Method Invocation

cz. 2

[Help:](#)

[RMI w praktyce](#)

[RMI- step by step \(Eclipse\)](#)

[Getting Started Using Java RMI](#)

Ćwiczenie 1. Lista rekordów

Zaimplementować aplikację, w której na żądanie klienta serwer RMI

- pobiera i przesyła klientowi listę rekordów z bazy danych (może być statyczna imitacja bazy danych);
- wyszukuje rekord po nazwie i zwraca go klientowi.

Serwer zwraca klientowi serializowane obiekty (np. Produkt, Osoba, itp.).

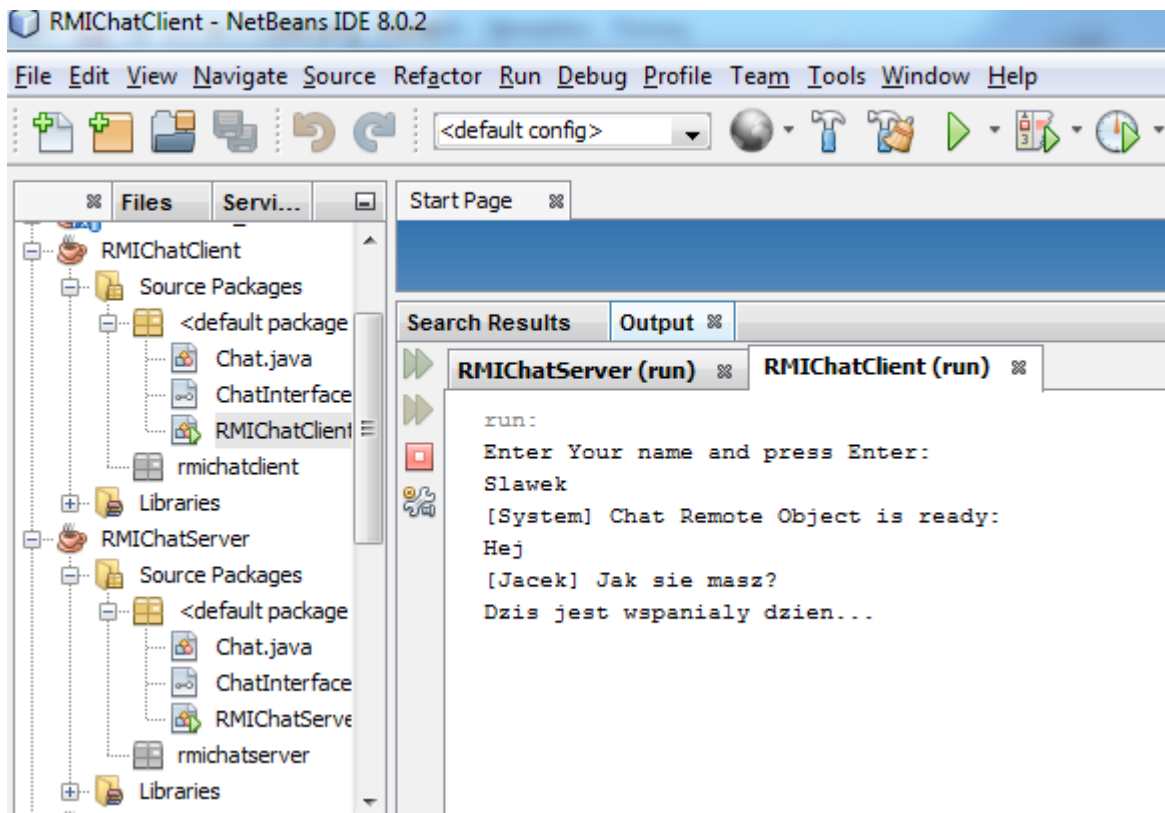
```
np. public List<Produkt> listaProduktow(){...}
```

System properties:

- The `java.rmi.server.codebase` property specifies the location, a codebase URL, from which the definitions for classes originating *from* this server can be downloaded. If the codebase specifies a directory hierarchy (as opposed to a JAR file), you must include a trailing slash at the end of the codebase URL.
- The `java.rmi.server.hostname` property specifies the host name or address to put in the stubs for remote objects exported in this Java virtual machine. This value is the host name or address used by clients when they attempt to communicate remote method invocations. By default, the RMI implementation uses the server's IP address as indicated by the `java.net.InetAddress.getLocalHost` API. However, sometimes, this address is not appropriate for all clients and a fully qualified host name would be more effective. To ensure that RMI uses a host name (or IP address) for the server that is routable from all potential clients, set the `java.rmi.server.hostname` property.
- The `java.security.policy` property is used to specify the policy file that contains the permissions you intend to grant.

Ćwiczenie 2. Chat

Zaimplementować aplikację Chat Client-Server - wysyłanie komunikatów między użytkownikami (uruchomić na 2 komputerach)



Ćwiczenie 3. Gra kółko i krzyżyk

Gra technologii Java RMI. Aplikacja można wykonywać w grupach **2-osobowych lub pojedynczo**. Aplikacja nie musi mieć interfejsu graficznego, może być konsolowa.

Zaimplementuj grę "kółko i krzyżyk". Implementacja powinna składać się z obiektów dwóch rodzajów – KKSerwer (obiekt zdalny zarządzający grą) oraz KKlient (obiekt gracza wywołujący metody z KKSerwer).

- Dodatkowym plusem będzie korzystanie z mechanizmu SSL.

=====

PS2.

1. RMI SSL - <https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/socketfactory/SSLInfo.html>

2. ORACLE Java Documentation - Java™ Remote Method Invocation API (Java RMI) - <https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/index.html>

3. Przykład w aplikacji Chat jak Server komunikuje się Klientem:
<http://www.ejbtutorial.com/java-rmi/java-rmi-example-simple-chat-program-between-server-and-client>