

GUÍA DE DESPLIEGUE DE LOS EJEMPLOS JAVA RMI

RMI-HolaMundo: Ejecución desde la línea de comandos

- Ver código de registro.bat, servidor.bat y cliente.bat
 - CLASSPATH apuntando a la clase (en registro, servidor y cliente)
 - NOTA: Aunque tanto al cliente como el servidor se les pasa el parámetro "java.security.policy", en el ejemplo de "HolaMundo", no aplica porque no se recupera la política de seguridad.
- Arrancar registro.bat y esperar unos segundos. Aceptar excepción Firewall
- Arrancar servidor.bat
- Editar cliente.bat y añadir un "pause" al final
- Arrancar cliente.bat (Se puede editar el bat y sustituir %1 por localhost. Si se omite, se toma por defecto localhost)

RMI-Dinamico: Ejecución desde el Eclipse

- Hacemos sobre el Proyecto RMI-Dinamico un Run as/Maven Install
 - Comprobar que la fase "package del POM" utilizando, utilizando el artifactId "maven-antrun-plugin" hace una copia de los ficheros Cliente.class, HolaMundo.class y Servidor.class a "../RMI-Dinamico-Web/src/main/webapp" desde "./target/classes", excluyendo los ficheros "**/*Dinamico.class", que son los que no se copian al servidor web y se ejecutan desde su ubicación
 - Comprobar en el Proyecto RMI-Dinamico_Web que están en "../RMI-Dinamico-Web/src/main/webapp"
- Sobre el Proyecto RMI-Dinamico_Web, Botón Derecho, Run As, Run on Server
 - Elegir el servidor Tomcat o Glassfish y Finish
 - IMPORTANTE: Probar que llegamos a las clases (TODAS) lanzando desde el navegador (tiene que saltar la ventana de "Guardar Como..."). Ej:
 - <http://localhost:8080/RMI-Dinamico-Web/es/ubu/Isi/Cliente.class>
- Ejecución de Servidor y Cliente desde el IDE
 - Arrancar el registro.bat (desde el IDE o desde la línea de comandos)
 - Arrancar el servidor (ServidorDinamico) como Java Application (sin argumentos)
 - Dará un error de Acces Denied
 - Incluir como parámetro de la VM en las Run Configurations
 - -Djava.security.policy=registerit.policy
 - -Djava.rmi.server.codebase=http://localhost:8080/RMI-Dinamico-Web/
 - Arrancar ahora el servidor
 - Arrancar el cliente (ClienteDinamico) haciendo lo mismo que en el servidor
- Con esta técnica, ServidorDinamico como ClienteDinamico tienen acceso remoto a las clases
 - Las clases se puede publicar bien por http, ftp, o file (carpeta compartida accesible por ambas partes)

RMI-DGC: Ejecución desde la línea de comandos

- Objetivo: Comprobar DGC (Distributed Garbage Collection)
 - El Cliente tiene un sleep de 30". Cambiarlo a 10". Thread.sleep(10000); y hacer un BUILD
- Arrancar el registro.bat
- Servidor.bat
 - Examinar las opciones
 - Intervalo para comprobar caducidades: -Dsun.rmi.dgc.checkInterval=500 (Recomendado a la mitad que leaseValue)
 - Duración estándar para las concesiones: -Djava.rmi.dgc.leaseValue=1000
 - Arrancar el Servidor
- Arrancar el cliente
 - Se crean los 100 objetos
 - Pasados los 10 segundos de pausa, el servidor destruye las referencias
 - Esto pasa porque concede caducidades de 1 segundo, es decir, una vez que se ejecuta, cuando termina el cliente, el servidor mantiene las referencias 1 segundo
 - Arrancar varias veces el cliente
- Modificación propuesta para comprobar DGC
 - Arrancar el registro.bat
 - Modificar Servidor.bat
 - Intervalo para comprobar caducidades: -Dsun.rmi.dgc.checkInterval=15000
 - Duración estándar para las concesiones: -Djava.rmi.dgc.leaseValue=30000
 - Arrancar servidor.bat
 - Arrancar el cliente
 - Se crean los 100 objetos. Esperar los 10" a que termine
 - Ejecutar otra vez el cliente y esperar los 10" a que termine
 - Ejecutar una tercera vez el cliente y esperar los 10" a que termine
 - El servidor NO elimina las referencias hasta pasados 30" de cada una de las ejecuciones
 - Pasados 30" de cada ejecución, las referencias del servidor de la ejecución que caduca se eliminan

RMI-Callbacks: Ejecución desde Eclipse

- Arrancar el registro.bat (comprobar parámetros)
- Servidor
 - Incluir como parámetro de la VM en las Run Configurations
 - -Djava.security.policy=registerit.policy
- CallbackApplet
 - OJO: Cambiar el puerto al puerto anónimo "0".
 - UnicastRemoteObject.exportObject(this, 0);
 - port - the port number on which the remote object receives calls (if port is zero, an anonymous port is chosen)
 - Arrancar CallbackApplet:
 - Botón Derecho sobre (ejecutar como Java Applet)

Licencia

Autor: Raúl Marticorena
Área de Lenguajes y Sistemas Informáticos
Departamento de Ingeniería Informática
Escuela Politécnica Superior
UNIVERSIDAD DE BURGOS



Esta obra está bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported. No se permite un uso comercial de esta obra ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula esta obra original

Licencia disponible en <http://creativecommons.org/licenses/by-nc-sa/3.0/>