

التطبيقات الموزعة – جلسة 7 – RMI مقدمة 1

أهداف الجلسة

في هذه الجلسة سيتعرف الطلاب على بناء Client Library لخدمات RMI وكيفية إضافة التسامح مع الأخطاء إلى برامج RMI بحيث إذا فشل مخدم ما يتم توجيه الطلبات لمخدم آخر.

في نهاية الجلسة سيكون الطالب قادر على:

1. بناء Client Library لكي تستخدم في الوصول إلى خدمات RMI بدل التخاطب مع RMI بشكل مباشر.
2. إضافة إمكانية التسامح مع الأخطاء بحيث إذا فشل مخدم RMI يتم توجيه الطلبات لمخدم آخر.
3. حل مثال عن خدمة تسجيل مستخدمين جدد وتسجيل الدخول لمستخدمين موجودين في النظام، وتعلم استخدام صف Storage مجرد لتخزين المعلومات عن طريق تعريف صفوف ترث منه.
4. توسيع النظام الموزع عن طريق غضافة مخدمات جديدة و خدمة تحدد المخدمات الشغالة حالياً. (مطالعة)

بناء Client Library

مكتبة الزبون Client Library عبارة عن مجموعة من الصفوف و الطرائق التي تسهل الوصول إلى خدمة معينة من خلال تغليف توابع استدعاء الخدمة ضمن صفوف و طرائق مختلفة، في مجال ال RMI تساهم مكتبة الزبون في جعل الاستدعاءات فعلاً تبدو على أنها تتم على كائنات محلية من دون أي استخدام ل lookup أو RMI Registry.

عادةً ما يقوم مطور الخدمة ببناء مكتبة تسهل استثمار خدمته من قبل المطورين و المبرمجين الآخرين و يقوم بتغليف هذه المكتبة بملف JAR و توزيعه عبر شبكة الإنترنت، بحيث يقوم المستخدم للخدمة بتنزيل الملف و تضمينه ضمن مشروعه ثم البدء باستخدامه كما يستخدم أي مكتبة أخرى.

سأقوم برفع مقاطع فيديو توضح كيف يتم توليد ملف JAR و استخدامه ضمن eclipse و netbeans.

عدا عن كون هذه المكتبات تسهل استخدام الخدمات فإنها أيضاً تمكن من تغيير واجهة الخدمات المقدمة و تحديث المكتبة بشكل متوافق مع النسخ السابقة منها الأمر الذي يمكن مستخدمي المكتبة من متابعة استخدام أي نسخة جديدة منها من دون أي تغيير في الكود الخاص بهم.

التسامح مع الأخطاء

إن أي نظام موزع لا بد أن يكون قادر على التسامح مع أي خطأ يصيب أي مخدم يعمل ضمنه، بمعنى أن يكون قادر على الإستمرار في تقديم خدماته حتى بعد تعرض أي مخدم ضمنه للعطل أو توقفه عن العمل لأي سبب كان، لتحقيق هذا الأمر لا بد من آلية لكشف تعطل أي مخدم و توقف إرسال الطلبات إليه كما لا بد من أن يكون أي مخدم قادر على تقديم أي طلب يأتي إليه من دون الرجوع لأي مخدم آخر ضمن النظام.

يمكن تحقيق هذا الأمر من طرف الزبون أو من طرف المخدم.

من طرف الزبون يتم بناء مكتبة الزبون بحيث تقوم بإرسال طلب لمخدم آخر في حال فشل الطلب من مخدم ما من دون أن يعرف برنامج الزبون بأمر الفشل.

من طرف المخدم يمكن استخدام موازن حمل بحيث يتولى توجيه الطلبات القادمة إليه من الزبائن إلى المخدمات التي تعمل فقط و يتجنب المخدمات التي تسبب أي نوع من المشاكل.

التوسيع

أحياناً قد تحتاج النظم الموزعة إلى إضافة مخدمات جديدة إلى النظام بسبب عدم مقدرة المخدمات الحالية على تقديم جميع طلبات الزبائن, في هذه الحالة لا بد أن يوفر النظام الموزع المقدرة على غضافة مخدم جديد بشكل مؤتمت أو يدوي مع إمكانية استخدام هذا المخدم لتقديم طلبات الزبائن فوراً.

في حال كنا نستخدم مكتبة زبون للوصول لخدمات النظام يمكن برمجة خدمة خاصة تخزن أسماء المخدمات المتوفرة و يتم استخدامها من قبل مكتبة الزبون لأغرسال الطلبات إلى أي مخدم منها في حال فشل أحد المخدمات أو في حال كانت إستجابة بعض المخدمات بطيئة.

أما في حال كنا نستخدم موزان حمل يمكن أن نضيف المخدمات الجديدة إلى إعداداته بحيث يبدأ بتوجيه الطلبات إليها فوراً.

تمرين

تتضمن المحاضرة 3 تمارين كل واحد منها يبني على التمرين الذي يسبقه, في البداية يتم بناء خدمة RMI تقدم خدمة تسجيل مستخدم جديد و تسجيل دخول للمستخدم بحيث الأولى تحفظ اسم المستخدم مع password hash في ملف أو ضمن قوائم مترابطة في الذاكرة و الثانية تقارن كلمة السر المدخلة مع الكلمة المخزنة و ترد true في حالة التطابق و إلا ترد false.

هنا يتم بناء واجهة UserInt تتضمن الطريقتين login و register و صف User يقوم بتحقيق الواجهة السابقة و يحقق الطريقتين, ضمن هذا الصف يتم تعريف كائن من النوع المجرد Storage لتخزين اسم المستخدم و كلمة المرور (في الطريقة register) أو استعادة كلمة المرور المخزنة لمستخدم مخزن مسبقاً (في الطريقة login).

يتم تعريف صف مجرد باسم Storage يوفر طريقتين مجردتين هما إضافة مستخدم جديد مع كلمة سر و الحصول على كلمة السر لمستخدم مخزن مسبقاً.

يتم تعريف صف FileStorage يرث الصف السابق و يخزن المعلومات ضمن ملفات, و يتم تعريف صف MemoryStorage يرث الصف Storage و يخزن المعلومات ضمن قائمتين مترابطتين في الذاكرة.

بعد ذلك يجب بناء مكتبة زبون Client Library تتضمن صف UserService يوفر طريقتين login و register لاستدعاء الطريقتين في الصف User عن بعد على المخدم.

الآن نقوم بتوليد ملف JAR لهذه المكتبة و استيراده ضمن مشروع آخر و نستخدم الصف UserService.

حل التمارين موجود على github ضمن الروابط التالية:

<https://github.com/mohsenSy/RMIIdentityServiceProject>

<https://github.com/mohsenSy/RMIIdentityServiceClient>

Contents

أهداف الجلسة.....	1
بناء Client Library.....	1
التسامح مع الأخطاء.....	1
التوسيع.....	2
تمرين.....	2