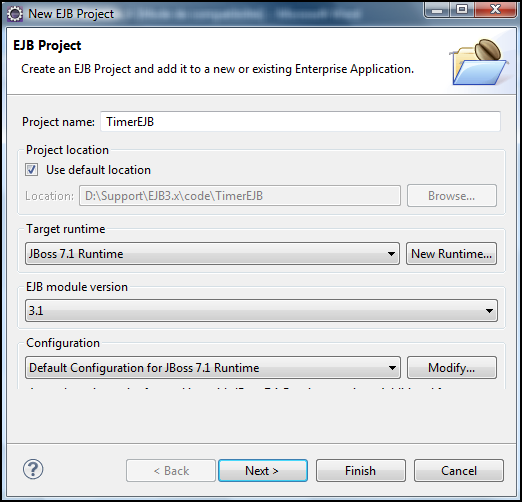
# **Projet :** TimerService

## **Etape**

Créez un projet EJB dans eclipse

Project Name: **TimerEJB**

****

**Finish**

## Etape : Créez l’interface remote

**package** com.formation.service;

**import** javax.ejb.Remote;

@Remote

**public** **interface** TimerSample {

**public** **void** scheduleTimer(**long** milliseconds);

**public** String checkTimerStatus();

}

## Etape : Créez le EJB

**package** com.formation.service.implementation;

**import** java.util.Collection;

**import** java.util.Date;

**import** java.util.Iterator;

**import** javax.annotation.Resource;

**import** javax.ejb.SessionContext;

**import** javax.ejb.Stateless;

**import** javax.ejb.Timeout;

**import** javax.ejb.Timer;

**import** com.formation.service.TimerSample;

@Stateless

**public** **class** TimerSampleBean **implements** TimerSample{

@Resource

**private** SessionContext ctx;

**public** **void** scheduleTimer(**long** milliseconds) {

ctx.getTimerService().createTimer(

**new** Date(**new** Date().getTime() + milliseconds), "Hello World");

}

@Timeout

**public** **void** timeoutHandler(Timer timer) {

System.*out*.println("---------------------");

System.*out*.println("\* Received Timer event: " + timer.getInfo());

System.*out*.println("---------------------");

timer.cancel();

}

@Override

**public** String checkTimerStatus() {

Timer timer = **null**;

Collection<Timer> timers = ctx.getTimerService().getTimers();

Iterator<Timer> iterator = timers.iterator();

**while** (iterator.hasNext()) {

timer = iterator.next();

**return** ("Timer will expire after " +

timer.getTimeRemaining() + " milliseconds.");

}

**return** ("No timer found");

}

}

## Etape : Créez un client java

Créez un projet java

Nom Projet : **TimerEJBClient**

## Etape : Configuration

Inclure le projet **TimerEJB** dans le path du projet **TimerEJBClient**

Inclure le jar **jboss\_client.jar**

Créez le fichier de configuration JBoss **jboss-ejb-client.properties**

Copier ClientUtil.java du projet HelloEJBClient vers le projet TimerEJBClient Attention : respectez l’arborescence des répertoires(**com.formation.util**).

## Etape : Client

Ecrire le client java

**package** com.formation.client;

**import** javax.naming.Context;

**import** javax.naming.NamingException;

**import** com.formation.service.TimerSample;

**import** com.formation.service.implementation.TimerSampleBean;

**import** com.formation.util.ClientUtil;

**public** **class** TimerJavaClient {

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

TimerSample bean = *doLookup*();

// 4. Appel du service

//starts the timer

bean.scheduleTimer(5000);

System.*out*.println(bean.checkTimerStatus());

}

**private** **static** TimerSample doLookup() {

Context context = **null**;

TimerSample bean = **null**;

**try** {

// 1. Obtenir le context

context = ClientUtil.*getInitialContext*();

// 2. Generate JNDI Lookup name

String lookupName = *getLookupName*();

// 3. Lookup and cast

bean = (TimerSample) context.lookup(lookupName);

} **catch** (NamingException e) {

e.printStackTrace();

}

**return** bean;

}

**private** **static** String getLookupName() {

// nom ear

String appName = "";

//nom du jar ejb (nom du projet dans eclipse par défaut)

String moduleName = "TimerEJB";

//nom optionnel JBoss7

String distinctName = "";

// Nom de la classe implementant le EJB

String beanName = TimerSampleBean.**class**.getSimpleName();

// Nom complet de l'interface

**final** String interfaceName = TimerSample.**class**.getName();

// Create a look up string name

String name = "ejb:" + appName + "/" + moduleName + "/"

+ distinctName + "/" + beanName + "!" + interfaceName;

**return** name;

}

}

## Etape : Test

Deployer le EJB

Lancer JBoss

Tester avec la classe **TimerJavaClient.java** du projet **TimerEJBClient**

Dans la console du client



Dans la console de JBoss 7

