fansofjava

fansofjava

居然日访问量突破50了，看来mongodb真的快流行起来了。

2010-02-23 通过网页

>>更多闲聊

浏览: 14180 次

性别:

来自: Runner

我现在离线

详细资料 留言簿

搜索本博客

搜索

最近访客 >>更多访客

foruforo vdgame

2009-03-17

log4j | EJB3 TimerService

EJB3 事务管理

事务分本地事务和分布式事务，本地事务相对简单，这里讨论一下分布事事务。
分布式环境如何管理涉及一个以上资源的事务呢？主要是通过被称为两阶段提交（two-phase commit）来实现的。在执行一个以上的数据库操作时，如果第一个数据库操作成功，第二个数据库操作失败，要回滚第一个操作是非常困难的。所以在提交事务之前，两阶段提交协议会询问每个资源管理器的当前事务能否提交成功，如果任何一个资源管理器表示不能提交事务，那么就回滚整个事务。

EJB有两种使用事务的方式。
第一种方式通过容器管理的事务，叫CMT，另一种通过bean管理的事务叫BMT。

在CMT中，容器自动提供事务的开始、提交和回滚。总是在业务方法的开始和结束处标记事务的边界。下面以ejb3 in action上的例子说明：

Java代码

```
1.  @Stateless
2.  @TransactionManagement(TransactionManagementType.CONTAINER)
3.  public class OrderManagerBean {
4.      @Resource
5.      private SessionContext context;
6.  …
7.  @TransactionAttribute(TransactionAttributeType.REQUIRED)
8.
9.      public void placeSnagItOrder(Item item, Customer customer){
10.         try {
11.             if (!bidsExisting(item)){
12.                 validateCredit(customer);
13.                 chargeCustomer(customer, item);
14.                 removeItemFromBidding(item);
15.             }
16.         } catch (CreditValidationException cve) {
17.             context.setRollbackOnly();
18.         } catch (CreditProcessingException cpe){
```

xderam sopestar

博客分类

我的留言簿 >>更多留言

- 怒了 偶的神哪

  -- by tinguo002
- 两年以前的东西了，早忘了。

```
19.             context.setRollbackOnly();
20.         } catch (DatabaseException de) {
21.             context.setRollbackOnly();
22.         }
23.     }
24. }
```

其中，@TransactionAttribute(TransactionAttributeType.REQUIRED)表示指定事务的类型。
如果省略，默认为CMT方式。
@TransactionAttribute(TransactionAttributeType.REQUIRED)通知容器如何管理事务，
事务的属性控制了事务的使用范围，因为事务之间的关系非常的复杂，这个属性主要是用来处理事务与事务之间怎样来处理的的问题。
具体作用见附件图片。

如果产生一个系统异常，容器将自动回滚该事务。 EJBException是RuntimeException的子类，即也是一个系统运行时异常。
如果Bean抛出一个普通的非继承自RutimeException应用异常，事务将不会自动回滚，但可以
通过调用EJBContext的SetRollbackOnly回滚。
当然EJB上下文还有一个getRollbackOnly方法，通过返回一个boolean值来确定CMT事务是否已被标记为回滚。如果开始非常耗资源的操作前判断此bean的事务已被标记为回滚，则可以节约很多系统资源。
对于上面的异常回滚操作，还有一更加优雅的方式：

Java代码

```
1.  public void placeSnagItOrder(Item item, Customer customer)
2.      throws CreditValidationException, CreditProcessingException,
3.          DatabaseException {
4.      if (!bidsExisting(item)){
5.          validateCredit(customer);
6.          chargeCustomer(customer, item);
7.          removeItemFromBidding(item);
8.      }
9.  }
10. ...
11. @ApplicationException(rollback=true)
12. public class CreditValidationException extends Exception {
13. ...
14. @ApplicationException(rollback=true)
15. public class CreditProcessingException extends Exception {
16. ...
17. //系统异常
18. @ApplicationException(rollback=false)
19. public class DatabaseException extends RuntimeException {
```

-- by fansofjava

- 哥们，想当年你的毕业论文还有在不，借偶copy一下 我是大专的，学校不会很严， 贪 ...

  -- by tinguo002

其他分类

- 我的收藏 (82)
- 我的论坛主题贴 (6)
- 我的所有论坛贴 (263)
- 我的精华良好贴 (0)

最近加入圈子

- lucene爱好者
- JBoss SEAM
- liferay
- JBPM @net

存档

- 2010-03 (3)
- 2010-02 (3)
- 2010-01 (2)
- 更多存档...

最新评论

- struts2分页

  这个很容易理解，底层写两个方法，一个方法用于获取要显示的记录，一个用于获取总共多少 ...

  -- by fansofjava

- struts2分页

  打错字了，上面的一句： 还是

@ApplicationException把JAVA核对与不核对的异常标识为应用程序异常。其rollback默认为false,表示程序不会导致CMT自动回滚。但应用程序异常应当慎用，见下文：

If the container detects a system exception, such as an ArrayIndexOutOfBounds or NullPointerException that you did not guard for, it will still roll back the CMT transaction. However, in such cases the container will also assume that the Bean is in inconsistent state and will
destroy the instance. Because unnecessarily destroying Bean instances is costly, you should never
delibertely use system exceptions.

2.bean管理事务

由于CMT依靠容器开始、提交和回滚事务，所以会限制事务的边界位置。而BMT则允许通过编程的方式来指定事务的开始、提交和回滚的位置。主要使用的是javax.transaction.UserTransaction接口。

如下面代码：

Java代码

```java
@Stateless)
@TransactionManagement(TransactionManagementType.BEAN)
public class OrderManagerBean {
    @Resource
    private UserTransaction userTransaction;

    public void placeSnagItOrder(Item item, Customer customer){
        try {
            userTransaction.begin();
            if (!bidsExisting(item)){
                validateCredit(customer);
                chargeCustomer(customer, item);
                removeItemFromBidding(item);
            }
            userTransaction.commit();
        } catch (CreditValidationException cve) {
            userTransaction.rollback();
        } catch (CreditProcessingException cpe){
```

不明白数据集是什么时候被分割的555

-- by tinguo002

- struts2分页

55555555555555 我一直都是认为，在查询数据库的时候就已经把返回的Li ...

-- by tinguo002

- struts2分页

哇是呀 我最近也在为struts2的分页发愁，所以去网上查了一下，发现关于分页 ...

-- by tinguo002

- mongodb基本操作

不过zxc005 写道fansofjava 写道原来没注意看，居然还有Object ...

-- by fansofjava

评论排行榜

- struts2中文问题
- mongodb基本操作
- struts2分页
- DbUtils学习(一)
- 关于spring security的URL路径验证问题

RSS
Add to Google
订阅到 鲜果
[什么是RSS?]

```
19.                 userTransaction.rollback();
20.             } catch (DatabaseException de) {
21.                 userTransaction.rollback();
22.             } catch (Exception e) {
23.                 e.printStackTrace();
24.             }
25.         }
26.     }
```

@TransactionManagement(TransactionManagementType.BEAN) 指定了事务的类型为BMT，上面没有用到@TransactionAttribute，因为它只适用于CMT，在上面代码中，可以显示的指定事务的开始与提交，因此更加的灵活。上面最关键的地方是注入了UserTransaction资源。其中获得UserTransaction资源的方式有三种，除了用EJB资源的方式注入以外，还有以下两种方式：

（1） JNDI查找

Java代码

```
1.  Context context = new InitialContext();
2.  UserTransaction userTransaction =
3.   (UserTransaction) context.lookup("java:comp/UserTransaction");
4.  userTransaction.begin();
5.  // Perform transacted tasks.
6.  userTransaction.commit();
```

如果在EJB之外，则可使用此方法，如在不支持依赖注入的JBoss4.2.2的Web工程或helper class即帮助器类中都可以。

（2）EJBContext

Java代码

```
1.  @Resource
2.  private SessionContext context;
3.  ...
4.  UserTransaction userTransaction = context.getUserTransaction(); userTransaction.begin();
5.  // Perform transacted tasks.
6.  userTransaction.commit();
```

注意：getUserTransaction方法只能在BMT中使用。如果在CMT中使用，则会抛出IllegalStateException异常。且在BMT中不能使用EJBContext的getRollbackOnly和setRollbackOnly方法，如果这样使用，也会抛出IllegalStateException异常。

如果使用有状态的Session Bean且需要跨越方法调用维护事务，那么BMT是你唯一的选择，当然BMT这种技术复杂，容易出错，且不能连接已有的事务，当调用BMT方法时，总会暂停已有事务，极大的限制了组件的重用。故优先考虑CMT事务管理。

查看图片附件

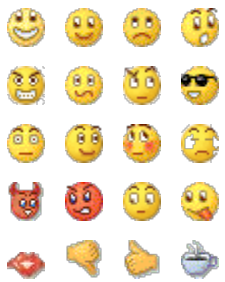
log4j | EJB3 TimerService

00:41 | 浏览 (1812) | 评论 (0) | 分类: EJB3 | 相关推荐

评论

发表评论

表情图标



B I U Quote Code List Img URL Flash Table

字体颜色: 标准 字体大小: 标准 对齐: 标准

提示：选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

您还没有登录，请登录后发表评论(快捷键 Alt+S / Ctrl+Enter) 提交