Forums

Service Implementation Best Practices: Delegation to other localServices	Olaf Kock	March 14, 2012 7:18 PM
RE: Service Implementation Best Practices: Delegation to other localService	Julio Camarero	March 14, 2012 7:25 PM
RE: Service Implementation Best Practices: Delegation to other localService	Olaf Kock	March 14, 2012 9:00 PM
RE: Service Implementation Best Practices: Delegation to other localService	Olaf Kock	March 14, 2012 9:19 PM
RE: Service Implementation Best Practices: Delegation to other localService	Edward Shin	March 14, 2012 9:54 PM
RE: Service Implementation Best Practices: Delegation to other localService	Jonathon Omahen	March 14, 2012 11:37 PM
RE: Service Implementation Best Practices: Delegation to other localService	Armin Cyrus Dahncke	March 14, 2012 11:45 PM
RE: Service Implementation Best Practices: Delegation to other localService	Edward Shin	March 14, 2012 11:57 PM
RE: Service Implementation Best Practices: Delegation to other localService	Armin Cyrus Dahncke	March 15, 2012 8:30 AM
RE: Service Implementation Best Practices: Delegation to other localService	Richard Sezov	March 15, 2012 11:48 PM
RE: Service Implementation Best Practices: Delegation to other localService	Miguel Pastor Olivar	March 16, 2012 7:00 PM
RE: Service Implementation Best Practices: Delegation to other localService	Olaf Kock	March 16, 2012 8:50 PM

Service Implementation Best Practices: Delegation to other localServices

March 14, 2012 7:18 PM

Olaf Kock

Rank: Jedi Knight
Posts: 119
Join Date: September 24, 2008
Recent Posts

Interesting aspect in the current advanced developer training:

Our implementations (e.g. addManufacturer, see below) imply that everything always works - there's no transaction handling in the LocalServiceImpls that we teach:

```
    public Manufacturer addManufacturer(Manufacturer newManufacturer,
            ServiceContext serviceContext) throws SystemException,
            PortalException {

        long manufacturerId = CounterLocalServiceUtil
                .increment(Manufacturer.class.getName());
        long userId = serviceContext.getUserId();

        Manufacturer manufacturer = manufacturerPersistence
                .create(manufacturerId);
```

When one insert fails, all the others should be rolled back, right? Or am I just a victim of MySql autocommit? (admitting that I'm not extremely familiar with servicebuilder aspects beyond the training exercises)

We had the interesting aspect where one student inserted data with invalid references during the webservice exercise (e.g. invalid userId) that caused the later LAR export to fail. There seems to be not too much sanity check that we teach (or at least mention - would be enough for training purposes). He also inserted invalid companyIds, but it might have been in times when permission checking was not yet implemented in the WS exercise (I missed this due to other duties, so my codebase doesn't reflect it and I can't remember last year's training)

Anything interesting I can answer them?

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 7:25 PM

Julio Camarero

Rank: Padawan
Posts: 51
Join Date: July 16, 2008
Recent Posts

Hi Olaf,

any method inside a *LocalServiceImpl is transactional. This is explained is some slides about ServiceBuilder....
so, you shouldn't have the issues you are mentioning....

cheers!

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 9:00 PM

Olaf Kock

Rank: Jedi Knight
Posts: 119
Join Date: September 24, 2008
Recent Posts

thanks Julio,

as we saw the issues in an actual implementation (which puzzled me), it's probably due to mysql autocommit - thus nontransactional on the database side. I'll forward this to the student

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 9:19 PM

Olaf Kock

Rank: Jedi Knight
Posts: 119
Join Date: September 24, 2008
Recent Posts

ok, following up - he's using postgresql. Does calling a localservice from a localservice run in the same initial transaction context as well? He also had the manufacturerPersistence.update(manufacturer, false); call below the other localservice calls if that makes a difference

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 9:54 PM

Edward Shin

Rank: Jedi Knight
Posts: 153
Join Date: March 24, 2005
Recent Posts

I believe there's an annotation in one of the generated classes that tells Spring whether to make the method transactional, and any call within the same method should be run in the same transaction context afaik.

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 11:37 PM

Jonathon Omahen

Rank: Youngling
Posts: 24
Join Date: September 29, 2009
Recent Posts

AFAIK, like Julio said above, everything is transactional. That being said, I'm pretty sure that each service operates in its own transaction context. So if a separate service's transaction fails, that doesn't necessarily rollback your service's transaction. But I could be totally off on that.

Anyone have insight?

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 11:45 PM

Armin Cyrus Dahncke

Rank: Youngling
Posts: 12
Join Date: August 13, 2008
Recent Posts

AFAIK Jon is right in his assumption it is problematic to have a atomic transaction ootb, however if I recall correctly calling the persistence of another entity would circumvent that limitation. But then the BL in the referred impl classes will not b executed.
Setting up a different TM with JTA tomcat is supposed to work, but I have no insight how that is exactly done.

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 14, 2012 11:57 PM

Edward Shin

Rank: Jedi Knight
Posts: 153
Join Date: March 24, 2005
Recent Posts

Check UserLSI. We have a number of service calls to different services within the same addUser method. If one of those services fails like ContactLSI.addContact, then the entire transaction should roll back, and it should work the same in ManufacturerLSI too.

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 15, 2012 8:30 AM

Armin Cyrus Dahncke

Rank: Youngling
Posts: 12
Join Date: August 13, 2008
Recent Posts

I just had a look at ULSI in 6.0SP1 and 6.1 EE as those are the ones we use in the trainings:

```
        // Contact

        Date birthday = PortalUtil.getDate(
            birthdayMonth, birthdayDay, birthdayYear,
            new ContactBirthdayException());
```

Settings | Online Friends (0)

```
        contact.setCompanyId(user.getCompanyId());
        contact.setUserId(creatorUserId);
        contact.setUserName(creatorUserName);
        contact.setCreateDate(now);
        contact.setModifiedDate(now);
        contact.setAccountId(company.getAccountId());
        contact.setParentContactId(ContactConstants.DEFAULT_PARENT_CONTACT_ID);
        contact.setFirstName(firstName);
        contact.setMiddleName(middleName);
```

Actually its using a mix later in that code there are LocalServiceCalls when dealing with groups

```
// Group

        groupLocalService.addGroup(
            user.getUserId(), User.class.getName(), user.getUserId(), null,
            null, 0, StringPool.SLASH + screenName, true, null);

        // Groups

        if (groupIds != null) {
            groupLocalService.addUserGroups(userId, groupIds);
```

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 15, 2012 11:48 PM

Richard Sezov

Moderator

Rank: Padawan

Posts: 52

Join Date: February 7, 2007

Recent Posts

This is something that I think there's ambiguity on at the highest levels of the company. When preparing Liferay in Action, I was told that transactions only work if you use the injected persistence class (i.e., you set up a reference in Service Builder). I wrote that all up in the chapter where we start making use of assets.

This was later recanted, and I was told that transactions would also work by calling a -LocalServiceImpl directly. I left the text about references in because I was told it was still a best practice to reference the services you're planning to use in your services, so that a persistence implementation gets injected.

So for pedagogical purposes, I try to provide references and use the injected persistence class all the time. This both teaches the best practice and maintains consistency within the code. You'll see this throughout the code in Advanced Dev, and it was done on purpose that way.

--Rich

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 16, 2012 7:00 PM

Miguel Pastor Olivar

Rank: Youngling

Posts: 1

Join Date: March 2, 2011

Recent Posts

Hey guys,

Just for clarification on how transactions works on our services

The methods in services are transactional. So, when you call a method on a services this will run inside a transaction

The misconception usually arrives when calling a method service from another method of the same service class:

Imagine we have the following service

```
class SimpleService {
  public void doStuff(){
      doMoreStuff();
  }
 public void doMoreStuff(){
   // do my stufff
}
}
```

The transactional methods run inside a **TransactionInterceptor** aspect so, the method call doMoreStuff() is not being intercepted (you are inside the service!!!).

How must I write the code??

All the services have a reference to their own, so the previous class should be something like this:

Settings | Online Friends (0)

```
class SimpleService {
  public void doStuff(){
      simpleService.doMoreStuff();
  }
 public void doMoreStuff(){
   // do my stufff
}
BeanReference(SimpleService.class)
protected SimpleService simpleService
}
```

Note the implementation of the **doStuff** method; we are using the **simpleService** reference instead of using **this**. Using this approach this call will be intercepted by the transaction interceptor.

I'm not sure I expressed myself quite right so ping me again if you have more questions.

Cheers,

Migue

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

RE: Service Implementation Best Practices: Delegation to other localService

March 16, 2012 8:50 PM

Olaf Kock

Rank: Jedi Knight
Posts: 119
Join Date: September 24, 2008
Recent Posts

Not sure if I understand: What will be rolled back on this:

```
class SimpleService {
  public void doStuff(){
      simpleService.doMoreStuff();
      simpleService.doEvenMoreStuff();
  }
public void doMoreStuff(){
   // do something fancy that should be persisted to the database
}
public void doEvenMoreMoreStuff(){
   // do something fancy that should be persisted to the database
```

0 (0 Votes) Flag

Reply | Reply with Quote | Quick Reply

NEWS BLOGS TWITTER FACEBOOK

Powered by Liferay Portal EE

PRIVACY POLICY SITEMAP CONTACT US ABOUT US CAREERS

Settings | Online Friends (0)