# Kune - Defect #572
# Add a participant to a wavelet doesn't work

01/14/2014 05:23 PM - Pablo Ojanguren

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 01/14/2014 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Pablo Ojanguren | **% Done:** | 100% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **Resolution:** | fixed | **Tags:** | |

**Description**

Add a participant to a wave doesn't work, the added participant doesn't see the wave.
The following exception is thrown when the participant is added:

```
org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role: cc.kune.domain.WaveEntity.participants,
no session or session was closed
    at
org.hibernate.collection.internal.AbstractPersistentCollection.throwLazyInitializationException(AbstractPersistentCollection.java:393)  at
:393)
    at
org.hibernate.collection.internal.AbstractPersistentCollection.throwLazyInitializationExceptionIfNotConnected(AbstractPersistentCollection.
tCollection.java:385)
    at org.hibernate.collection.internal.AbstractPersistentCollection.readSize(AbstractPersistentCollection.java:125)
    at org.hibernate.collection.internal.PersistentSet.size(PersistentSet.java:156)
    at cc.kune.domain.WaveEntity.add(WaveEntity.java:100)
    at cc.kune.core.server.manager.impl.WaveEntityManagerDefault.add(WaveEntityManagerDefault.java:64)
    at com.google.inject.persist.jpa.KuneJpaLocalTxnInterceptor.invoke(KuneJpaLocalTxnInterceptor.java:98)
    at
cc.kune.wave.server.search.CustomPerUserWaveViewHandlerImpl.addWaveToUser(CustomPerUserWaveViewHandlerImpl.java:78)  a
va:78)
    at
cc.kune.wave.server.search.CustomPerUserWaveViewHandlerImpl.access$2(CustomPerUserWaveViewHandlerImpl.java:73)
    at
cc.kune.wave.server.search.CustomPerUserWaveViewHandlerImpl$1.call(CustomPerUserWaveViewHandlerImpl.java:129)
    at cc.kune.wave.server.search.CustomPerUserWaveViewHandlerImpl$1.call(CustomPerUserWaveViewHandlerImpl.java:1)
    at java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:303)
    at java.util.concurrent.FutureTask.run(FutureTask.java:138)
    at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:895)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:918)
    at java.lang.Thread.run(Thread.java:662)
```

## Associated revisions

**Revision 204e97df - 01/16/2014 12:14 PM - Pablo Ojanguren**

#571, #572 Fixes

## History

**#1 - 01/14/2014 07:22 PM - Pablo Ojanguren**

At WaveEntity.java:68, adding option eager fetch for participants prevents that exception.

```
@ManyToMany(mappedBy = "waves", cascade = { CascadeType.PERSIST, CascadeType.MERGE }, fetch = FetchType.EAGER)
 private Set<ParticipantEntity> participants;
```

To consider performace of this.

**#2 - 01/19/2014 04:04 PM - Pablo Ojanguren**

*- Status changed from New to Closed*

*- Assignee set to Pablo Ojanguren*

*- % Done changed from 0 to 100*

*- Resolution set to fixed*

Fixed in commit 204e97d