
SimianArmy

multi-threading handling

Group 5

AWSClient

```
private synchronized void init() {
    if (nextId != null) {
        return;
    }
    File dbFile = null;
    dbFile = (dbFilename == null) ? tempDbFile() : new File(dbFilename);
    if (dbpassword != null) {
        db = DBMaker.newFileDB(dbFile)
            .closeOnJvmShutdown()
            .encryptionEnable(dbpassword)
            .make();
    } else {
        db = DBMaker.newFileDB(dbFile)
            .closeOnJvmShutdown()
            .make();
    }
    eventMap = db.getTreeMap("eventMap");
    nextId = db.createAtomicLong("next", 1);
}
```

initial implementation

```
private void init() {
    if (nextId == null) {
        synchronized(this) {
            if (nextId == null) {
                File dbFile = null;
                dbFile = (dbFilename == null) ? tempDbFile() : new File(dbFilename);
                if (dbpassword != null) {
                    db = DBMaker.newFileDB(dbFile)
                        .closeOnJvmShutdown()
                        .encryptionEnable(dbpassword)
                        .make();
                } else {
                    db = DBMaker.newFileDB(dbFile)
                        .closeOnJvmShutdown()
                        .make();
                }
                eventMap = db.getTreeMap("eventMap");
                nextId = db.createAtomicLong("next", 1);
            }
        }
    }
}
```

refactored implementation

LocalDbRecorder

```
/** {@inheritDoc} */  
@Override  
public synchronized ComputeService getJcloudsComputeService() {  
    if (jcloudsComputeService == null) {  
        String username = awsCredentialsProvider.getCredentials().getAWSAccessKeyId();  
        String password = awsCredentialsProvider.getCredentials().getAWSSecretKey();  
        ComputeServiceContext jcloudsContext = ContextBuilder.newBuilder("aws-ec2").credentials(username, password)  
            .modules(ImmutableSet.<Module>of(new SLF4JLoggingModule(), new JschSshClientModule()))  
            .buildView(ComputeServiceContext.class);  
  
        this.jcloudsComputeService = jcloudsContext.getComputeService();  
    }  
  
    return jcloudsComputeService;  
}
```

initial implementation

```
/** {@inheritDoc} */  
@Override  
public ComputeService getJcloudsComputeService() {  
    if (jcloudsComputeService == null) {  
        synchronized(this) {  
            if (jcloudsComputeService == null) {  
                String username = awsCredentialsProvider.getCredentials().getAWSAccessKeyId();  
                String password = awsCredentialsProvider.getCredentials().getAWSSecretKey();  
                ComputeServiceContext jcloudsContext = ContextBuilder.newBuilder("aws-ec2").credentials(username, password)  
                    .modules(ImmutableSet.<Module>of(new SLF4JLoggingModule(), new JschSshClientModule()))  
                    .buildView(ComputeServiceContext.class);  
  
                this.jcloudsComputeService = jcloudsContext.getComputeService();  
            }  
        }  
    }  
  
    return jcloudsComputeService;  
}
```

refactored implementation