

# Cause of Death: Hibernate

**Disclaimer:**

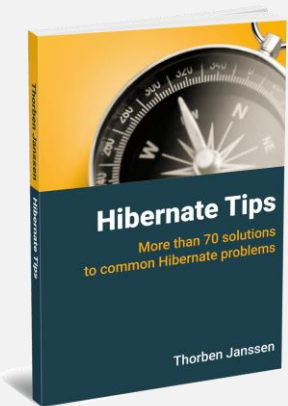
**You can use Hibernate for  
high-performance applications**

Independent consultant,  
trainer and author



CDI 2.0 Expert  
Group Member

Co-Organizer  
JUG Paderborn



Hibernate Tips  
More than 70 solutions to common  
Hibernate problems  
[www.hibernate-tips.com](http://www.hibernate-tips.com)



Thorben Janssen



@thjanssen123



/c/ThoughtsOnJava



/ThoughtsOnJava

[www.thoughts-on-java.org](http://www.thoughts-on-java.org)



[www.thoughts-on-java.org](http://www.thoughts-on-java.org)

**Breaking Things = MORE FUN!**

# How to kill a project

- Veeeery bad performance
- Inconsistent results
- Data loss

# Bad Performance

# Avoid Bad Performance

- Use FetchType.LAZY & use case specific projections
  - Exclude associations in equals / hashCode / toString
- Use Set instead of List
- Be careful about CascadeType.REMOVE
- Check 2nd Level Cache configuration with Query Cache

# Inconsistent Results



# Avoid Inconsistent Results

- Use separate JOIN and JOIN FETCH clauses
- Don't forget your caches
- Update both ends of bi-directional associations
- Don't use Open Session in View anti-pattern

# Data Loss

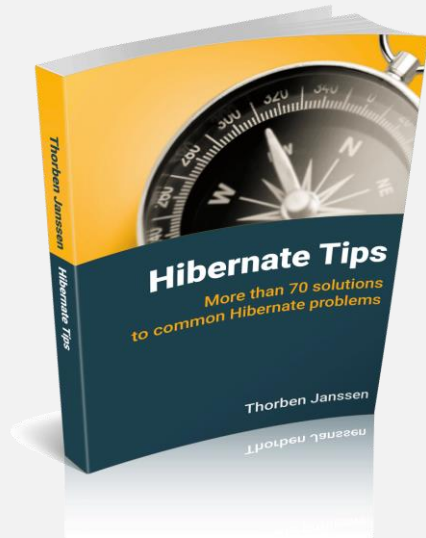
# Avoid Data Loss

- Never clear without flush
- Don't use CascadeType.REMOVE for Many-to-Many

# Tutorials, Online Courses & Consulting

[www.thoughts-on-java.org](http://www.thoughts-on-java.org)

## My Book



### **Hibernate Tips**

*More than 70 solutions to common  
Hibernate problems*

[www.hibernate-tips.com](http://www.hibernate-tips.com)