1)Data Exist Validation : While creating new record in database

with old data it should avoid that old data using validatior

**UomDaoImpl code:**

@Override

public boolean isUomAlreadyExist(String uomModel) {

//select count(\*) from uomtab where uom\_model=?

@SuppressWarnings("unchecked")

List<Long> countList=(List<Long>)

ht.findByCriteria(DetachedCriteria.forClass(Uom.class)

.setProjection(Projections.rowCount())

.add(Restrictions.eq("uomModel", uomModel))

);

return countList.get(0)!=0?true:false;

}

2) Data Delete connected with other record: While deleting UOM which is used in item creation avoid with error message

check using Parent table - child FK column.

**UomDaoImpl code:**

@Override

public boolean isUomConnectedToItem(Integer id) {

//select count(\*) itemtab where uomIdFk=?

@SuppressWarnings("unchecked")

List<Long> countList=(List<Long>)

ht.findByCriteria(DetachedCriteria.forClass(Item.class)

.setProjection(Projections.rowCount())

.add(Restrictions.eq("uom.id", id))

);

return countList.get(0)!=0?true:false;

}

3)If row deleted with Id and trying to delete same record again then throw exception

"Data Detele Repeated" or equal message.

**UomController code :** Delete method re-written

@RequestMapping("/delete")

public String deleteUom(@RequestParam Integer id,ModelMap map) {

String message=null;

try {

if(service.isUomConnectedToItem(id)) {

message="Uom '"+id+"' Used in other Operations can't be deleted";

}else {

service.deleteUom(id);

message="Uom '"+id+"' deleted successfully";

}

}catch (HibernateOptimisticLockingFailureException e) {

message="Uom '"+id+"' Not Found";

}

map.addAttribute("message",message );

List<Uom> uoms=service.getAllUoms();

map.addAttribute("uoms",uoms);

return "UomData";

}

Enable Log4J:

<!-- Log4J Jars -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

log4j.properties

# Root logger option

log4j.rootLogger=DEBUG, stdout, file

# Redirect log messages to console

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c**{1}**:%L - %m%n

# Redirect log messages to a log file

log4j.appender.file=org.apache.log4j.RollingFileAppender

#outputs to Tomcat home

#log4j.appender.file.File=${catalina.home}/logs/myapp.log

log4j.appender.file.File=d:/logs/warehouse.log

log4j.appender.file.MaxFileSize=5MB

log4j.appender.file.MaxBackupIndex=10

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c**{1}**:%L - %m%n

##Frameworks levels

log4j.logger.org.springframework=WARN

log4j.logger.org.hibernate=WARN