import ashimalinksharing.\*

class BootStrap {

def init = { servletContext ->

createUser()

createTopics()

/\*createResources()

createRatings()

createSubcriptions()

createLinkResource()

createReadingItems()

createDocumentResource()\*/

}

void createUser(){

User user=new User(fname:"ashima",email:"ashima19@gmail.com",userName:"ashima123",password:"pass123",confirmPassword:"pass123",lname:"yadav",admin:true,active:false);

user.save(failOnError:true)

User user1=new User(fname:"reena",email:"reena@gmail.com",userName:"reena1",password:"123ree",confirmPassword:"123ree",lname:"kumari",admin:false,active:false);

user1.save(failOnError:true)

User user2=new User(fname:"ajay",email:"ajay@gmail.com",userName:"ajaysingh",password:"heyyou",confirmPassword:"heyyou",lname:"singh",admin:false,active:false);

user2.save(failOnError:true)

}

void createTopics(){

List users1=User.list();

users1.each {user->

5.times{

Topic topic=new Topic(name:"groovy ${it}" ,visibility:PUBLIC )

user.addToTopics(topic)

user.addToSubscriptions(new Subscription(topic:topic,Seriousness:SERIOUS))

user.save(failOnError:true);

}

}

}

/\*void createResources(){

List topics=Topic.list();

topics.eachWithIndex { topic,index->

User user=User.findById(index%2+1)

topic.addToResources(

new LinkResource(creater:user, title:"link title ${index+1}",description:"link title${index+1} descripion", url:"https://grails.github.io/grails-doc/link${index+1}" )

)

topic.addToResources(

new DocumentResource(creater:user, title:"document title ${index+1}",description:" document title${index+1}descripion", filePath:"//filepath/grails.github.io/grails-doc/link${index+1}" )

)

topic.save(failOnError:true);

}

}

void createReadingItems(){

int size=Resource.list().size();

User.list().each{user->

Random random=new Random()

//println Resource.get(Math.abs(random.nextInt() % size))

user.addToReadingItems(new ReadingItem(resource:Resource.get(Math.abs(random.nextInt() % size)),isRead:true ))

user.addToReadingItems(new ReadingItem(resource:Resource.get(Math.abs(random.nextInt() % size)),isRead:true ))

user.addToReadingItems(new ReadingItem(resource:Resource.get(Math.abs(random.nextInt() % size)),isRead:true ))

user.save(failOnError:true);

}

}

void createDocumentResource(){}

void createLinkResource(){}

void createRatings(){}

void createSubcriptions(){}

\*/

def destroy = {

}

}