

Session的save()方法用来将一个临时对象转变为持久化对象，也就是将一个新的实体保存到数据库中。通过save()将持久化对象保存到数据库需要经过以下步骤：

- 1系统根据指定的ID生成策略，为临时对象生成一个唯一的OID；
- 2将临时对象加载到缓存中，使之变成持久化对象；
- 3提交事务时，清理缓存，利用持久化对象包含的信息生成insert语句，将持久化对象保存到数据库。

OK，下面来看一个实例：

```
//创建SessionFactory
Configuration config = new Configuration();
ServiceRegistry sr = new
ServiceRegistryBuilder().applySettings(config.getProperties()).buildServiceRegistry();

SessionFactory sessionFactory = config.buildSessionFactory(sr);

//打开Session
Session session = sessionFactory.openSession();

//开启事务
Transaction tx = session.beginTransaction();

//创建临时对象并赋值
User user = new User();
user.setId("1");
user.setUserName("lsj");
session.save(user);

//提交事务
tx.commit();

//关闭Session
session.close();
```

PS：

- 1从Hibernate 4之后，Configuration类中，原先常用的，获取SessionFactory的方法buildSessionFactory()被标记为过时，官方建议使用buildSessionFactory(ServiceRegistry sr)这个方法来获取SessionFactory。
- 2如果映射文件中为对象的ID指定了生成策略，那么在程序中为其ID赋值是无效的。会被系统自动生成的值覆盖，例如：

映射文件指定ID生成策略由Hibernate控制自增：

```
<id name="id" type="string">
    <generator class="increment">
</id>
```

然后程序中进行如下操作：

```
user.setId("1");  
System.out.println("手动赋值为："+user.getId());  
session.save(user);  
System.out.println("存入数据库的值为："+user.getId());
```

这样，最终存入数据库的ID是Hibernate自动生成的自增ID，而不是我们手动给的值。另外，执行save()方法时并不会将对象存入数据库，在提交事务时，对象才被真正的保存到数据库中。save()方法是将对象保存到Session的缓存中，提交事务时，Hibernate会生成相应的insert语句，将对象保存到数据库。

再跟大家说一点需要注意的地方，Hibernate在提交事务的时候，会将之前对对象做的操作一并提交。就算是在save()方法之后做的操作也一样。什么意思呢？我们看下面的例子：

```
user.setUserName("old");  
session.save(user);  
user.setUserName("new");  
tx.commit();
```

执行以上操作，最终user.getUserName();得到的值将会是“new”，而不会是“old”，因为Hibernate在提交事务的时候把对user对象的所有操作都提交了。