In Hibernate, it was often to build a SessionFactoryand pull out a Session as follows:

SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();

Session session = sessionFactory.openSession();

However, since Hibernate 4.x, this approach is deprecated. According to [Hibernate 4.0 API docs](https://www.codejava.net/frameworks/hibernate/building-hibernate-sessionfactory-from-service-registry#buildSessionFactory()), the Configuration class’ buildSessionFactory() method is deprecated and it recommends developers to use the buildSessionFactory(ServiceRegistry) instead. Here is the new recommended code snippet that builds the SessionFactorybased on a ServiceRegistry and obtains the Session:

Configuration configuration = new Configuration().configure();

ServiceRegistryBuilder registry = new ServiceRegistryBuilder();

registry.applySettings(configuration.getProperties());

ServiceRegistry serviceRegistry = registry.buildServiceRegistry();

SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry);

Session session = sessionFactory.openSession();

The Configuration class’ configure() method loads mappings and properties from the convention resource file hibernate.cfg.xml which should be present in the classpath. It’s also very common to create a separate utility class for building the SessionFactoryas follows:

package net.codejava.hibernate;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

import org.hibernate.service.ServiceRegistryBuilder;

public class HibernateUtil {

private static SessionFactory sessionFactory;

public static SessionFactory getSessionFactory() {

if (sessionFactory == null) {

Configuration configuration = new Configuration().configure();

ServiceRegistryBuilder registry = new ServiceRegistryBuilder();

registry.applySettings(configuration.getProperties());

ServiceRegistry serviceRegistry = registry.buildServiceRegistry();

sessionFactory = configuration.buildSessionFactory(serviceRegistry);

}

return sessionFactory;

}

}

**NOTE:**As of 2014, the above code is deprecated. Here’s new version of the HibernateUtil class that is up-to-date:

package net.codejava.hibernate;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

public class HibernateUtil {

private static SessionFactory sessionFactory;

public static SessionFactory getSessionFactory() {

if (sessionFactory == null) {

// loads configuration and mappings

Configuration configuration = new Configuration().configure();

ServiceRegistry serviceRegistry

= new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

// builds a session factory from the service registry

sessionFactory = configuration.buildSessionFactory(serviceRegistry);

}

return sessionFactory;

}

}

Here’s an example of using this utility class in a Hibernate application:

SessionFactory sessionFactory = HibernateUtil.getSessionFactory();

Session session = sessionFactory.openSession();

session.beginTransaction();

session.save(myObject);