**Hibernate 4 : How to build SessionFactory,** If you have been watching previous hibernate releases then you must have noticed that they have deprecated a lot of classes in quick succession. **Deprecated classes are AnnotationConfiguration,  ServiceRegistryBuilder and so on**. These classes are used to build session factory through your java code and if you use them in your application; they might work well but you will always see some annoying warnings about their depreciation.

In this tutorial, I am giving an example of **building hibernate SessionFactory without using deprecated classes**mentioned above. I am using latest hibernate version i.e.**[Hibernate 4.3.6.Final](http://hibernate.org/orm/downloads/" \o "hibernate latest version)**, so you can make sure that you are using latest approach for building session factory.

**Classes used in building SessionFactory**

I have used following classes for building SessionFactory in hibernate 4.

1. **Configuration** : In place of deprecated AnnotationConfiguration
2. **StandardServiceRegistryBuilder** : In place of deprecated ServiceRegistryBuilder

Type 1.

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

sessionFactory = configuration.buildSessionFactory();

In Hibernate 3.6, “org.hibernate.cfg.AnnotationConfiguration” is deprecated, and all its functionality has been moved to “org.hibernate.cfg.Configuration“.

So , you can safely replace your “**AnnotationConfiguration**” with “**Configuration**” class.

Type 2.

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

*sessionFactory* = configuration.~~buildSessionFactory~~();

Type 3.

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

StandardServiceRegistryBuilder builder = **new** StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties());

sessionFactory = configuration.buildSessionFactory(builder.build());

Type 4.

Configuration configuration = **new** Configuration()

.configure(HibernateUtil.**class**.getResource("/hibernate.cfg.xml"));

StandardServiceRegistryBuilder serviceRegistryBuilder = **new** StandardServiceRegistryBuilder();

serviceRegistryBuilder.applySettings(configuration.getProperties());

ServiceRegistry serviceRegistry = serviceRegistryBuilder.build();

*sessionFactory* = configuration.buildSessionFactory(serviceRegistry);

Type 5.

// Create configuration instance

Configuration configuration = **new** Configuration();

// Create properties file

Properties properties = **new** Properties();

properties.load(Thread.*currentThread*().getContextClassLoader()

.getResourceAsStream("hibernate.properties"));

// Pass hibernate properties file

configuration.setProperties(properties);

// Since version 4.x, service registry is being used

ServiceRegistry serviceRegistry = **new** StandardServiceRegistryBuilder()

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

// Create session factory instance

*sessionFactory* = configuration.buildSessionFactory(serviceRegistry);