## 12. Transaction Management

### 12.1 Introduction to Spring Framework transaction management

### 12.2 Advantages of the Spring Framework’s transaction support model

### 12.3 Understanding the Spring Framework transaction abstraction

Spring事务原理，事务策略是通过org.springframework.transaction.PlatformTransactionManager接口。

**public** **interface** PlatformTransactionManager {

TransactionStatus getTransaction(

TransactionDefinition definition) **throws** TransactionException;

**void** commit(TransactionStatus status) **throws** TransactionException;

**void** rollback(TransactionStatus status) **throws** TransactionException;

}

TransactionException会被任何PlatformTransactionManager接口方法在运行状态下丢出异常，

getTranscation(…)方法会返回一个TransactionStatus依托于TransactionDefintion。

TransactionDefintion接口定了以下内容：

* Isolation（独立性）：
* Propageation（传播性）：所有的代码都可以在一个事务的局域内运行
* Timeout（逾期性）：在事务逾期时间范围前，事务能够常态地运行
* Read\_only status（只读）：一个只读事务状态，只读状态是一种优化措施，详见Hibernate。