

Spring的JDBC支持

一、JdbcTemplate 概述

1、是 spring 框架中提供的一个对象，是对原始 Jdbc API 对象的简单封装。

2、spring 框架为我们提供了很多的操作模板类。

操作关系型数据的：

JdbcTemplate

HibernateTemplate

操作 nosql 数据库的：

RedisTemplate

操作消息队列的：

JmsTemplate

二、spring配置JdbcTemplate

数据库连接 url= jdbc:mysql://localhost:3306/mybatis?
characterEncoding=utf8

1、引入相应jar包

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.itheima</groupId>
  <artifactId>day04_eesy_01jdbctemplate</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.0.2.RELEASE</version>
    </dependency>
```

```

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>5.0.2.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>5.0.2.RELEASE</version>
</dependency>

<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.6</version>
</dependency>
</dependencies>
</project>

```

2、编写spring 的配置文件

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">

```

配置连接池的三种方式

1) Spring默认的连接池:

```

<!-- 配置Spring默认的连接池 -->
<bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName" value="com.mysql.jdbc.Driver">
</property>
  <property name="url" value="jdbc:mysql://localhost:3306/easy"> </property>
  <property name="username" value="root"> </property>
  <property name="password" value="1234"> </property>
</bean>

```

执行操作

```

public static void main(String[] args) {
  //1.获取容器

```

```
ApplicationContext ac = new ClassPathXmlApplicationContext("bean.xml");  
//2.获取对象  
JdbcTemplate jt = ac.getBean("jdbcTemplate",JdbcTemplate.class);  
//3.执行操作  
jt.execute("insert into account(name,money)values('ddd',2222)");  
}
```

2) DBCP连接池

3) C3P0连接池