

实验一.txt

实验一， 数据源注入

DataSourceDemo

```
|  
|-----tarena  
|  
|-----spring-config.xml
```

```
<!-- 加入Spring2Core -->  
<!-- 加入Spring2AOP -->  
<!-- 加入Spring2PersistenceCore -->  
<!-- 加入Spring2PersistenceJDBC -->  
<!-- 加入MySQL -->
```

步骤一，

```
public class User {
```

```
    private int id;  
    private String name;
```

```
    public User() {  
    }
```

```
    public int getId() {  
        return id;  
    }
```

```
    public void setId(int id) {  
        this.id = id;  
    }
```

```
    public String getName() {  
        return name;  
    }
```

```
    public void setName(String name) {  
        this.name = name;  
    }
```

实验一.txt

```
}
```

```
}
```

步骤二，

```
public interface IUserDao {  
    public void find(int id);  
    public void insert(User user);  
    public void update(User user);  
    public void delete(User user);  
}
```

步骤三，

```
public class UserDaoImpl implements IUserDao {  
  
    // 关联的属性  
    private DataSource dataSource;  
  
    // 注入的方法  
    public void setDataSource(DataSource dataSource) {  
        this.dataSource = dataSource;  
        if (this.dataSource != null) {  
            try {  
                if  
(this.dataSource.getConnection() != null) {  
System.out.println("与数据库成功连接");  
                }  
            } catch (SQLException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
  
    public void delete(User user) {  
        System.out.println("正在删除用户。。。");  
    }  
}
```

实验一.txt

```
public void find(int id) {  
    System.out.println("正在查询用户。。。");  
}  
  
public void insert(User user) {  
    System.out.println("正在增加用户。。。");  
}  
  
public void update(User user) {  
    System.out.println("正在修改用户。。。");  
}  
}
```

步骤四，

```
<?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-2.0.xsd">
```

```
    <!-- 带连接池功能的  
org.apache.commons.dbcp.BasicDataSource -->  
    <bean id="DriverManagerDataSource"  
class="org.springframework.jdbc.datasource.DriverManagerDataSour  
ce">  
        <property name="driverClassName">  
  
<value>com.mysql.jdbc.Driver</value>  
        </property>  
        <property name="url">
```

实验一.txt

```
<value>jdbc:mysql://localhost:3306/test</value>
    </property>
    <property name="username">
        <value>root</value>
    </property>
    <property name="password">
        <value>admin</value>
    </property>
</bean>

<bean id="UserDaoImpl" class="tarena.UserDaoImpl">
    <property name="dataSource">
        <ref bean="DriverManagerDataSource"
/>
    </property>
</bean>
</beans>
```

步骤五，

```
public class Test {
```

```
    public static void main(String[] args) {
```

```
        ApplicationContext context = new
ClassPathXmlApplicationContext(
                                "tarena/spring-config.xml");
    }
}
```

步骤六，

与数据库成功连接

实验一.txt