

# spring中基于注解 的声明式事务控制配置步骤(不需xml)

---

## 在基于仍需要xml配置的基础上

### 替换bean.xml:

创建3个Config类

JdbcConfig

```
/**
 * 和连接数据库相关的配置类
 */
public class JdbcConfig {

    @Value("${jdbc.driver}")
    private String driver;

    @Value("${jdbc.url}")
    private String url;

    @Value("${jdbc.username}")
    private String username;

    @Value("${jdbc.password}")
    private String password;

    /**
     * 创建JdbcTemplate
     * @param dataSource
     * @return
     */
    @Bean(name="jdbcTemplate")
    public JdbcTemplate createJdbcTemplate(DataSource dataSource){
        return new JdbcTemplate(dataSource);
    }
}
```

```

/**
 * 创建数据源对象
 * @return
 */
@Bean(name="dataSource")
public DataSource createDataSource(){
    DriverManagerDataSource ds = new DriverManagerDataSource();
    ds.setDriverClassName(driver);
    ds.setUrl(url);
    ds.setUsername(username);
    ds.setPassword(password);
    return ds;
}
}
SpringConfiguration

```

```

/**
 * spring的配置类，相当于bean.xml
 */
@Configuration
@ComponentScan("com.itheima")
@Import({JdbcConfig.class,TransactionConfig.class})
@PropertySource("jdbcConfig.properties")
@EnableTransactionManagement
public class SpringConfiguration {
}

```

TransactionConfig

```

/**
 * 和事务相关的配置类
 */
public class TransactionConfig {

    /**
     * 用于创建事务管理器对象
     * @param dataSource
     * @return
     */
    @Bean(name="transactionManager")
    public PlatformTransactionManager createTransactionManager(DataSource
dataSource){
        return new DataSourceTransactionManager(dataSource);
    }
}

```